Symmetry-based monocular 3D vehicle ground-truthing for traffic analytics

by Trong Thao Tran

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE

GRADUATE PROGRAM IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

YORK UNIVERSITY

TORONTO, ONTARIO

SEPTEMBER, 2024

© Trong Thao Tran, 2024

Abstract

3D object detection is a critical component of autonomous driving and traffic analytics. Current research relies on LiDAR-derived ground truth for training and evaluation. However, LiDAR ground truth can be expensive and usually inaccurate in the far field due to the sparsity of LiDAR returns. Here, assuming a fully calibrated camera and a 3D terrain model, we explore whether relatively inexpensive RGB imagery might be used to obtain 3D ground truth, based on the bilateral symmetry of motor vehicles. From manually annotated symmetry points and tire-ground contact points, a vertical symmetry plane and 3D point cloud are inferred and used to estimate vehicle location, pose, and dimensions. These dimensional estimates are then input to a probabilistic model derived from a standard public motor vehicle dataset to form maximum a posteriori estimates of any remaining dimensions. In evaluations on a public traffic dataset, we find that this novel symmetry-based approach is much more accurate than LiDAR-based ground-truthing on single frames and comparable to LiDAR-based methods that propagate information across frames of vehicles.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Dr. James Elder, for his invaluable guidance, feedback, and support throughout the Master's program. His expertise and insistence on high-quality work have consistently inspired me to enhance my thesis. Reflecting on this period, I see how my work has benefited from Dr. Elder's sharp judgment, meticulous reviews, and insightful comments. As I approach the conclusion of my study, I am particularly thankful for Dr. Elder's encouragement to produce significant evaluation results and his support in submitting a paper to substantiate my research findings.

I am also grateful to Dr. Robert Allison for his role on the supervisory committee, providing support and advice for my thesis proposal and the final version. In Dr. Elder's lab, I appreciate Ms. Anna Kajor for her flexibility and assistance in managing my lab requests. Without Dr. Elder's financial support and Anna's incredible organizational skills, completing my study would have been a far greater challenge. I also extend my thanks to my fellow lab members and the community at York University for the enriching discussions both in and outside of class, which have led to many lasting friendships.

The entire Master's study at York University has been challenging but very rewarding in terms of knowledge acquisition and the improvement of research skills. I have learned how to tackle difficult problems in research with patience, grit, and flexibility. This has also been a period of personal growth in terms of mental strength and the ability to cope with pressure, which can be incredibly useful for my future career. It has been a wonderful experience for me to explore how the research process functions and how academia collaborates to ensure that high-quality and novel research is produced.

Finally, I want to acknowledge my family and friends for their patience during my study. There were times when I was unavailable to them, and I am deeply grateful for their unwavering presence and support.

Table of Contents

Abstract ii

Acknowledgements iii

Table of Contents v

List of Tables vi

List of Figures ix

1 Introduction 1
- 1.1 Background . . . 1
  - 1.1.1 Limitations of LiDAR-based ground-truthing . . . 1
  - 1.1.2 Thesis proposal . . . 2
- 1.2 Overview of method . . . 3
- 1.3 Results . . . 8

2 Related work 9
- 2.1 Ground-truthing for 3D object detection . . . 9
  - 2.1.1 Inter-frame propagation in LiDAR-based ground-truthing . . . 9
- 2.2 Monocular 3D inference in the built environment . . . 11
  - 2.2.1 Fitting projected cuboids . . . 11
  - 2.2.2 Parallelism . . . 11
  - 2.2.3 Bilateral symmetry . . . 14
- 2.3 Limitations of prior work . . . 18
  - 2.3.1 LiDAR sparsity . . . 18
  - 2.3.2 Dependence on straight line segments and the Manhattan assumption . . . 18
  - 2.3.3 Restriction to cuboid estimation . . . 19
  - 2.3.4 Absolute scale problem . . . 19
  - 2.3.5 Methods to automatically detect symmetry point pairs from 2D images . . . 20

3 Dataset 21

4 Annotation GUI 27
- 4.1 Tire-ground contact points . . . 27
- 4.2 Symmetry point pairs . . . 27
  - 4.2.1 Extremal symmetry point pairs . . . 30
- 4.3 Time Spent on Annotations . . . 32
- 4.4 YORK-SYM Dataset . . . 32

5 Inferring 3D points from 2D annotated points **33**
5.1 Camera projection . . . 33
5.1.1 Coordinate frames . . . 34
5.2 Terrain model . . . 35
5.3 Estimating the orientation of the symmetry plane . . . 38
5.4 Estimating the location of the symmetry plane . . . 40
5.4.1 Scenario 1: At least one tire-ground contact point is annotated on both sides of the vehicle . . . 42
5.4.2 Scenario 2: Tire-ground contact points on only one side of vehicle . . . 44
5.5 Estimating the 3D locations of symmetry point pairs . . . 46

6 Refining the algebraic estimates **48**
6.1 Incorporating statistics from the Canadian Vehicle Specifications (CVS) database to improve prediction quality . . . 50
6.1.1 Producing a complete bounding box from a set of 3D predicted symmetry point pairs and tire-ground contact points . . . 50
6.1.2 Using statistics from the Canadian Database as constraints during the non-linear optimization process . . . 54
6.2 Resolving systemic bias in annotations . . . 54

7 Evaluation **60**
7.1 Ground-truthing for 3D object detection . . . 60
7.1.1 3D object detection metrics . . . 62
7.1.2 Results and analysis . . . 63
7.1.3 Results and analysis based on difficulty levels . . . 64
7.2 Evaluating against real ground truth . . . 64
7.3 Evaluating consistency within a track . . . 71
7.4 Ablation studies . . . 72
7.4.1 Method to merge annotations from two annotators . . . 72
7.5 Qualitative results . . . 72

8 Discussion **77**
8.1 Limitations . . . 77
8.2 Conclusion . . . 78
8.3 Future Work . . . 78

Bibliography **80**

List of Tables

3.1 Specifications of cameras and LiDAR units used in the DAIR-V2X dataset (Yu et al., 2022). . 22

6.1 Descriptive statistics for key dimensions in the Canadian Vehicle Specifications database (Canada, 2023) (in cm). **OH:** Overall height. **OW:** Overall width. **WB:** Wheelbase. **RH:** Rear overhang. **FH:** Front overhang. 51
6.2 Pearson correlation coefficients for key dimensions. **OH:** Overall height. **OW:** Overall width. **WB:** Wheelbase. **RH:** Rear overhang. **FH:** Front overhang. 51
6.3 Maximum constraints adopted using estimated statistics from the Canadian Vehicle Specifications (CVS) database (Canada, 2023). 54
6.4 Comparison of predicted *OW* against ground truth make/model *OW* across four categories of annotation (see text). 57

7.1 Deviations between 3D bounding boxes estimated from symmetry and LiDAR data 63
7.2 Deviations between 3D bounding boxes estimated from symmetry and LiDAR data as a function of distance: Near-field ($\leq$ 30m), Mid-field (30 m - 50 m), Far-field (50m - 150m) 64
7.3 Comparison of estimated vehicle dimensions of OUR method, LiDAR (LD), and LiDAR with inter-frame propagation (LD_{prop}) against real ground truth. N = **697** vehicles. **569** out of **697** vehicles have wheelbase estimates. 69
7.4 Consistency of vehicle dimension estimates of OUR method, LiDAR (LD), and LiDAR with inter-frame propagation (LD_{prop}) over time. MSD: Mean standard deviation. MPSD: Mean percent standard deviation. **N = 375** tracks containing **2050** vehicle instances. 71
7.5 Comparison of performance when using different annotations 72

List of Figures

1.1 The 3D bounding box annotation process requires locating, rotating, and scaling operations. Adapted from Wang et al. (2019). . . . 2
1.2 Density of points returned from a Toyota Esquire by different LiDAR units tested. Adapted from Lambert et al. (2020). . . . 2
1.3 One of the intersections in the dataset DAIR-V2X (Yu et al., 2022), the primary dataset used for evaluating our method (see chapter 3) . . . 3
1.4 Recovery of 3D objects with bilateral symmetry in Li et al. (2012). . . . 4
1.5 The GUI we developed to annotate symmetry point pairs (linked by yellow line segments) and tire-ground contact points (magenta). An extremal symmetry point pair is highlighted with circle endpoints (white endpoint on the driver's side, yellow endpoint on the passenger's side) and a text label. They are aligned vertically with the outer edges of the vehicle's tires and are on the same 3D plane as the magenta tire-ground contact points. . . . 4
1.6 Our symmetry-based 3D vehicle ground-truthing pipeline. . . . 5
1.7 The overall process of symmetry-based monocular 3D vehicle ground-truthing. . . . 7

2.1 The frustum proposal, generated from the 2D bounding box in the RGB image, can help annotators locate distant objects with higher accuracy than by just examining the point cloud alone. This is because sparse LiDAR returns must be contained within this frustum (Wang et al., 2020). . . . 10
2.2 3D bounding boxes propagation across adjacent frames (Zimmer et al., 2019). . . . 11
2.3 The point of view of a camera used by Corral-Soto and Elder (2017) and a candidate cuboid model. . . . 12
2.4 Parallel line segments on the image are represented as interpretation planes on the Gauss Sphere. They converge at a common vanishing point. Adapted from Denis et al. (2008). . . . 13
2.5 The vanishing point vector is orthogonal to the great circle formed by a set of noisy interpretation plane normals. Adapted from Denis et al. (2008). . . . 13
2.6 The LS3D algorithm for generating 3D scale models of buildings based on the Manhattan assumption (Qian et al., 2019). . . . 14
2.7 Properties of bilateral symmetry. Adapted from (Pizlo et al., 2014). . . . 15
2.8 A perspective image of a bilaterally symmetric polygon with symmetry lines, which are also parallel, converging at a common vanishing point. Adapted from Pizlo et al. (2014). . . . 16
2.9 An example of degenerate views from Pizlo et al. (2014) that cannot be used to recover 3D shape using a 3D symmetry constraint. . . . 17
2.10 Li et al. use symmetry constraints to recover the 3D shapes of objects in an indoor environment. 18
2.11 Li et al. scaled the objects in 3D by assuming that they rest on an estimated ground plane. Adapted from Li et al. (2012). . . . 19

3.1 Setup of cameras and a LiDAR unit for the KITTI dataset (Geiger et al., 2013). . . . 21
3.2 Setup of cameras and LiDAR units for the DAIR-V2X dataset (Yu et al., 2022). RSU stands for Road-side Unit. . . . 22
3.3 Representative frames from the DAIR-V2X-C subset of the DAIR-V2X dataset (Yu et al., 2022). . . . 23

3.4 Example of a monocular image and its corresponding point cloud in the DAIR-V2X dataset (Yu et al., 2022). . . . 24
3.5 An example of a track verified by annotators in the DAIR-V2X dataset (Yu et al., 2022). . . 26

4.1 Annotation GUI was created using CVAT (CVAT.ai, 2023). Symmetry point pairs are indicated by yellow line segments. An extremal symmetry point pair is highlighted with circle endpoints (white endpoint on the driver's side, yellow endpoint on the passenger's side) and a text label. Tire-ground contact points are magenta. . . . 28
4.2 Locations of the four types of tire-ground contact points on a top view of a vehicle's body: Driver front (DF), Driver rear (DR), Passenger front (PF), and Passenger rear (PR). . . . 28
4.3 Tire-ground contact points represent the locations on the outer edges of a vehicle where the tires touch the ground surface (red points). 3D model provided by remtromol (2021). . . . 29
4.4 Annotations of three tire-ground contact points (magenta) for Driver front (DF), Driver rear (DR), and Passenger rear (PR). . . . 30
4.5 Symmetry point pairs are marked with yellow spheres. Extremal symmetry point pairs (green) are aligned vertically with the outer edges of a vehicle's tires. Tire-ground contact points (red) are also at the outer edges of a vehicle's tires. 3D model provided by remtromol (2021). . . . 31
4.6 Symmetry points are indicated by red points on the driver's side and green points on the passenger's side. An extremal symmetry point pair is marked with a bigger size. . . . 32

5.1 The camera is tilted down toward the road, showing the relationship between the **World coordinate frame** (Red-Green-Blue axes) and the **Camera coordinate frame** (Cyan-Yellow-Magenta axes). The figure also shows the transformation from the **LiDAR coordinate frame** to the **World coordinate frame** for the DAIR-V2X dataset (Yu et al., 2022). . . . 35
5.2 Raster of vertical depth Y_i for each pixel (x_i, y_i) in the region of interest. The region of interest, ROI, is highlighted by a red boundary. . . . 36
5.3 Hand-crafted image masks are employed to eliminate LiDAR returns from non-terrain infrastructure and vehicles in the far field. These are employed only to form the terrain model for the intersection. . . . 37
5.4 LiDAR returns in the far field for this intersection of the DAIR-V2X dataset (Yu et al., 2022) contain points from vehicles' bodies. Notice the projected LiDAR points on the three vehicles in the far field. . . . 37
5.5 Accumulation of LiDAR points across frames of one of the intersections of the DAIR-V2X dataset (Yu et al., 2022) within the ROI (red) defined by the masking procedure. . . . 38
5.6 Horizon line (in red) for a frame of the DAIR-V2X dataset (Yu et al., 2022). . . . 39
5.7 Each symmetry line $\tilde{\mathbf{l}}'_i$ (yellow) will, in general, intersect with the horizon line (red) at some point $\tilde{\mathbf{v}}'_{\mathrm{i}} = \tilde{\mathbf{l}}'_i \times \tilde{\mathbf{l}}_h$ (green) that is displaced from the true vanishing point $\tilde{\mathbf{v}}_{\mathrm{s}}$ due to annotation error. The point $\tilde{\mathbf{v}}^*_{\mathrm{s}}$ (blue), which minimizes the sum of squared distances to the symmetry lines $\left\{\tilde{\mathbf{l}}'_1, \ldots, \tilde{\mathbf{l}}'_{n_p}\right\}$, establishes the orientation of the symmetry plane of the vehicle. Symmetry points are in red on the driver's side and are in green on the passenger's side. . . . 40
5.8 Left extremal plane, symmetry plane, right extremal plane are in red, green, blue respectively. Extremal symmetry points are in green and tire-ground contact points are in red. 3D model provided by remtromol (2021). . . . 41
5.9 In this plan view, PR is the 3D location of the passenger rear tire-ground contact point and PF is the 3D location of the passenger front tire-ground contact point. We rectify the 3D locations of tire-ground contact points by finding their projections on the extremal plane. Original 3D point locations are in red and rectified 3D point locations are in green. . . . 43
5.10 This vehicle has tire-ground contact points (magenta) annotated only for the front and rear tires on the passenger's side. Symmetry points are in red on the driver's side and are in green on the passenger's side. An extremal symmetry point pair is shown with a bigger size. . . . 44

6.1 The **Vehicle-centered coordinate frame** with the origin $\mathbf{T}^*$ (black circle) located at one of the tire-ground contact points (Driver front in this example). The Z-axis of the vehicle's coordinate frame is constrained to point from the **driver's side** to the **passenger's side** and align with the normal to the symmetry plane $\mathbf{N}_s$. 49
6.2 Dimensions provided in the Canadian Vehicle Specifications (CVS) database (Canada, 2023) 52
6.3 Overall width, or OW, from the Canadian Vehicle Specifications (CVS) database (Canada, 2023) does not contain side-view mirrors. Adapted from Canada (2023). 53
6.4 Side view of a vehicle: Comparison of a predicted 3D bounding box before and after incorporating the Multivariate Gaussian model of dimensions. 53
6.5 An example of enhanced prediction following the application of constraints to the predicted dimensions: Prior to the enforcement of these constraints, the predicted 3D bounding box was excessively long due to the unconstrained Rear Overhang (RH). 55
6.6 An example of enhanced prediction following the application of constraints to the predicted dimensions: Prior to the enforcement of these constraints, the predicted 3D bounding box was excessively long due to the unconstrained Front Overhang (FH). 56
6.7 Cases in Category 3 where the annotation on the far side of the car causes the underestimation of overall width *OW*. 58

7.1 The 3D LiDAR-based bounding box does not cover the entire vehicle body as tire-ground contact points are excluded. 61
7.2 An example of a 3D LiDAR-based bounding box that does not cover the entire vehicle body. Our predicted 3D bounding boxes cover the entire vehicle's bodies while LiDAR-derived bounding boxes from the DAIR-V2X (Yu et al., 2022) dataset do not, causing **IoU** to be lower. 65
7.3 Examples of vehicle make and model using Google Image reverse search. 67
7.4 Two cases when Google Image reverse search returns ambiguous results. 68
7.5 LiDAR-derived bounding box (blue) and LiDAR-derived bounding box with inter-frame propagation (red). 70
7.6 An instance achieving the highest IoU with LiDAR-based ground truth (**IoU = 0.76**). . . . 74
7.7 An instance achieving the median IoU with LiDAR-based ground truth (**IoU = 0.34**). . . . 75
7.8 An instance achieving the lowest IoU with LiDAR-based ground truth (**IoU = 0.002**). . . . 76

Chapter 1

Introduction

1.1 Background

In recent years, substantial investments have been made in autonomous driving with the anticipation that autonomy can enhance safety and productivity. Autonomous vehicles like cars and robotic vehicles operating in unconstrained environments need to perceive their surroundings to prevent collisions and plan their trajectories. Concurrently, there has been a growing interest in traffic analytics at intersections and on highways, employing both Light Detection and Ranging (LiDAR) and camera images as inputs. The goals include object tracking, vehicle counting, and speed enforcement. For both autonomous driving and traffic analytics applications, 3-dimensional (3D) object detection is crucial. The purpose of 3D object detection is to detect an object's pose, location, and dimensions using sensor data. The training and evaluation of 3D object detection algorithms require 3D ground truth in the form of datasets. Hence, the creation of 3D vehicle ground-truth data receives significant attention and investment from companies and research organizations. One popular 3D object detection dataset is KITTI, which is an acronym for the Karlsruhe Institute of Technology and Toyota Technological Institute (Geiger et al., 2013). Since the introduction of this dataset, most state-of-the-art algorithms use LiDAR, RGB imagery, or a combination of both as inputs. In datasets like KITTI, nuScenes (Caesar et al., 2020), and Waymo (Sun et al., 2019), LiDAR sensors are mounted atop a vehicle. From time of flight, we obtain a 3D point cloud representing objects.

1.1.1 Limitations of LiDAR-based ground-truthing

Typically, a 3D bounding box supplies ground truth labels for the pose, location, and dimensions of a corresponding vehicle. The annotation process involves various operations such as locating, rotating, and

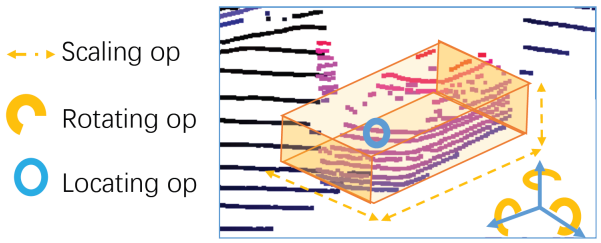


Figure 1.1: The 3D bounding box annotation process requires locating, rotating, and scaling operations. Adapted from Wang et al. (2019).

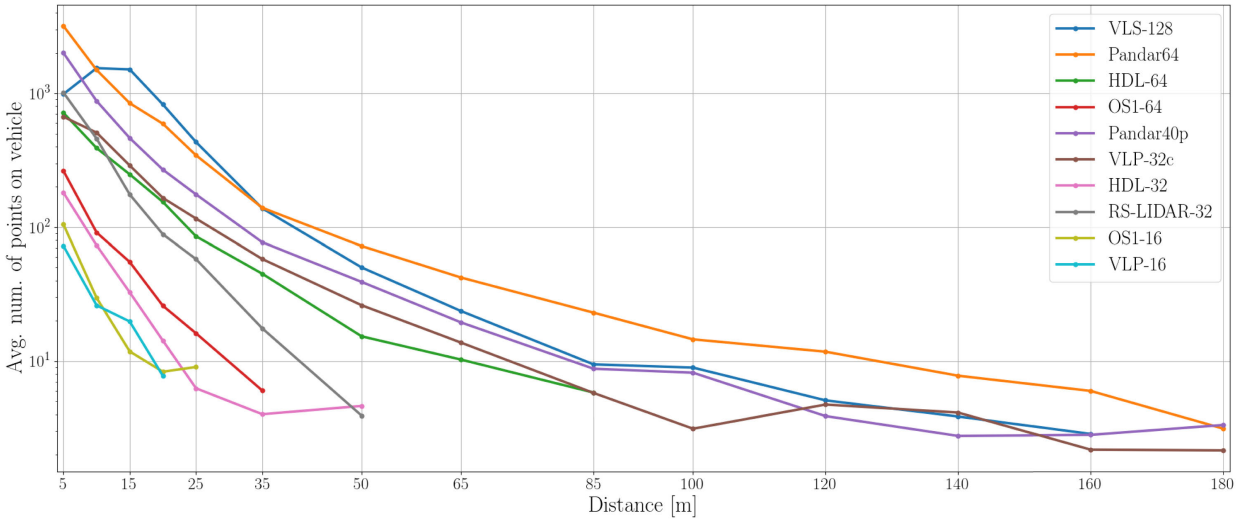


Figure 1.2: Density of points returned from a Toyota Esquire by different LiDAR units tested. Adapted from Lambert et al. (2020).

scaling (Figure 1.1). However, there are not always enough LiDAR returns for annotators to do these operations with accuracy and certainty. The number of LiDAR points returned from an object depends on its distance to the LiDAR unit. The further an object is from the LiDAR, the fewer returns received from that object. In Lambert et al. (2020), the authors analyzed ten popular LiDAR units in an indoor test bed with a target vehicle, a Toyota Esquire, positioned at varying distances up to 200 meters from a LiDAR unit. Depending on the LiDAR unit tested, the number of returns drops below ten at distances between 15 and 130 meters (Figure 1.2).

1.1.2 Thesis proposal

The sparsity and ambiguity of LiDAR data in the far field necessitate inter-frame propagation and verification using monocular imagery, which greatly complicates the ground-truthing process and may not produce accurate results. In my thesis, I propose an alternative ground-truthing approach that relies solely on **monocular imagery**. To develop this idea, I will focus on motor vehicles at traffic intersections. Figure 1.3

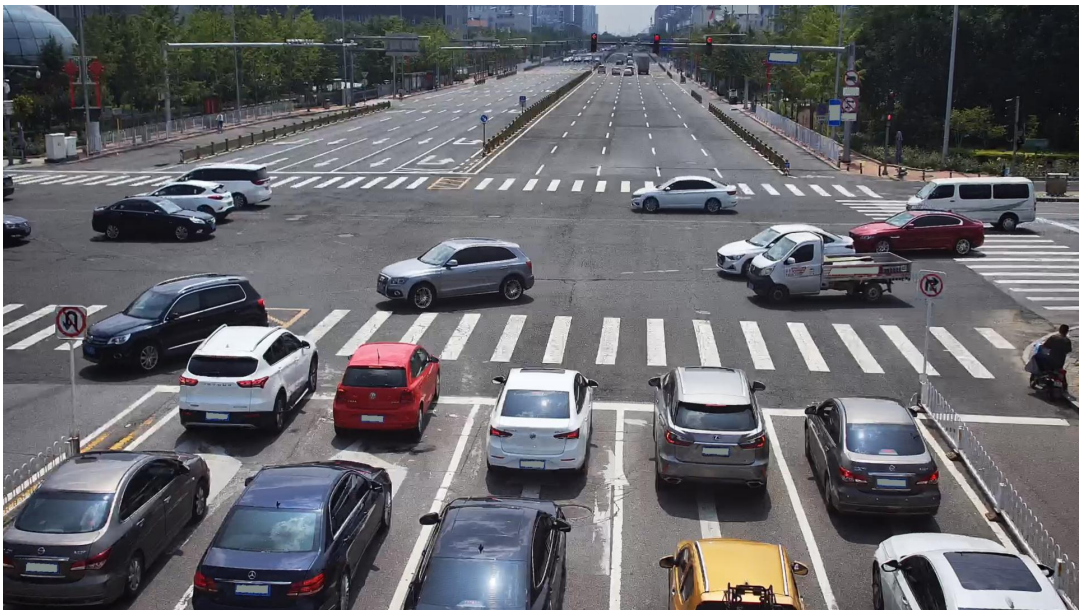


Figure 1.3: One of the intersections in the dataset DAIR-V2X (Yu et al., 2022), the primary dataset used for evaluating our method (see chapter 3)

shows an image of traffic navigating through an intersection from the (Yu et al., 2022) dataset, which is the primary dataset used for evaluating our method (see chapter 3). The diversity of vehicle poses at such intersections is crucial for demonstrating the robustness of our 3D ground-truthing algorithm.

In particular, we propose to exploit the **bilateral symmetry** of motor vehicles, drawing inspiration from the human visual system's sensitivity to symmetry. Li et al. (2012) explain how humans can perceive 3D shapes from 2D retinal images by merging 2D information with robust prior constraints such as symmetry. The methods in Li et al. (2012) computationally demonstrate this for an indoor scene if symmetry correspondences of bilaterally symmetric furniture are identified (Figure 1.4).

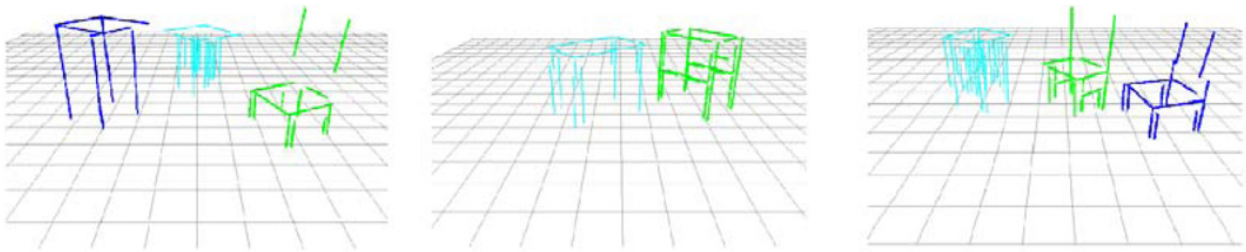
1.2 Overview of method

In our proposed method (Figure 1.6), with the images obtained from calibrated cameras, the first step involves human annotators identifying a set of symmetry correspondences and tire-ground contact points on vehicles, using our own annotation software (Figure 1.5).

Symmetry correspondences are pairs of symmetry points that mirror each other: for each pair, one point is on the left side of a vehicle and the other point is on the right side of a vehicle. There is a single plane of symmetry in 3D that divides the vehicle into a right half and a left half. We also define two extremal planes: the left extremal plane is the vertical plane that contains the left tire-ground contact points (if present) and the left extremal symmetry points (if present), and the right extremal plane is the vertical



(a) Estimation of 3D bounding boxes.



(b) 3D point cloud of furniture in the scene.

Figure 1.4: Recovery of 3D objects with bilateral symmetry in Li et al. (2012).

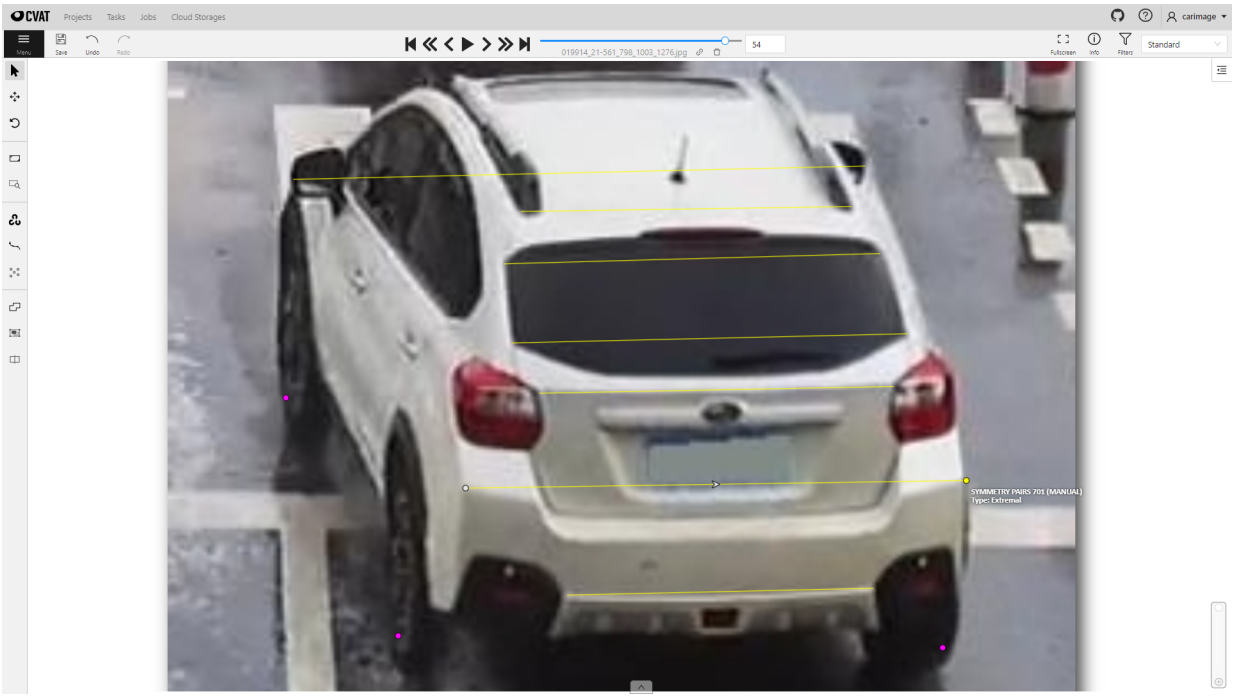


Figure 1.5: The GUI we developed to annotate symmetry point pairs (linked by yellow line segments) and tire-ground contact points (magenta). An extremal symmetry point pair is highlighted with circle endpoints (white endpoint on the driver's side, yellow endpoint on the passenger's side) and a text label. They are aligned vertically with the outer edges of the vehicle's tires and are on the same 3D plane as the magenta tire-ground contact points.

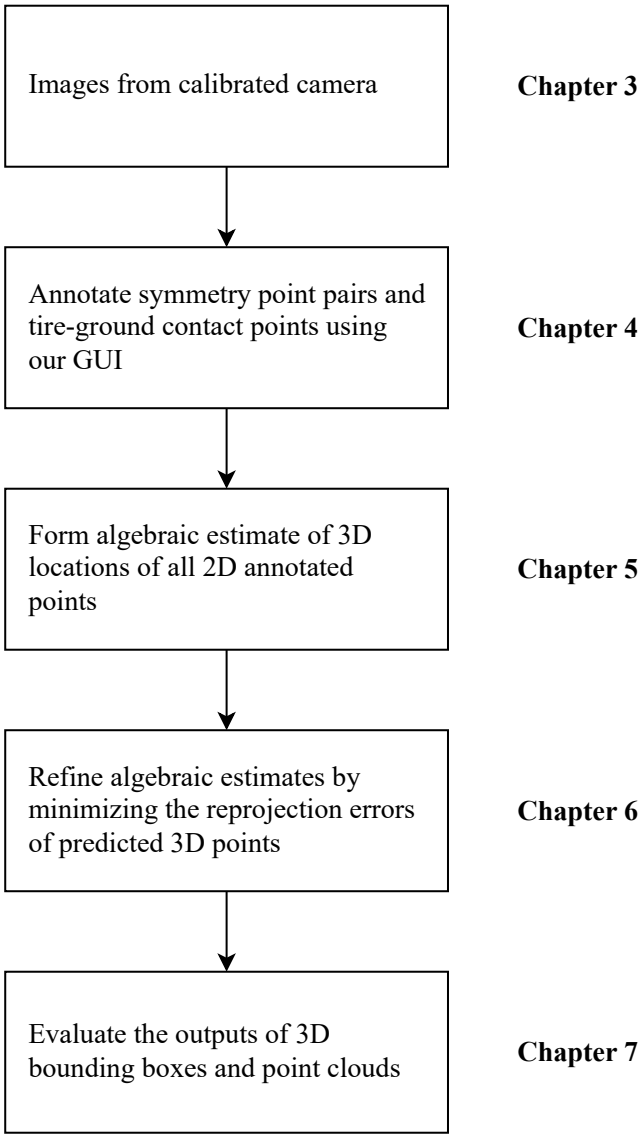


Figure 1.6: Our symmetry-based 3D vehicle ground-truthing pipeline.

plane that contains the right tire-ground contact points (if present) and the right extremal symmetry points (if present). Extremal symmetry point pairs are aligned vertically with the outer edges of a vehicle's tires. The symmetry plane, left extremal plane, and right extremal plane are parallel to each other. The symmetry plane is equidistant from the extremal planes.

The annotated symmetry point pairs are then used to estimate the orientation of the symmetry plane based on parallel line segments connecting symmetry point pairs. With a calibrated camera, we can obtain the horizon line. Assuming vehicles have zero roll, the symmetry lines should converge at a vanishing point on this horizon line (Figure 1.7a). With a calibrated camera, this vanishing point determines the direction of the corresponding 3D lines. This 3D direction determines the orientation of a vehicle's symmetry plane.

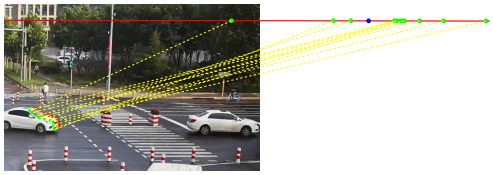
To locate this symmetry plane, a high-resolution terrain model is crucial. This study uses the LiDAR data available in the DAIR-V2X (Yu et al., 2022) dataset to construct a terrain model (Figure 1.7b). Other methods and data sources can also be used (Wilson, 2012). With a high-resolution terrain model, we then find the 3D locations of tire-ground contact points by finding the nearest 3D ground points to the rays from their 2D annotations. As mentioned above, these tire-ground contact points are on the extremal planes and therefore determine the planes' location. We can then locate the symmetry plane by maintaining an equal distance constraint (Figure 1.7c).

After fully identifying the algebraic representation of the symmetry plane, we can calculate the 3D locations of symmetry point pairs based on the constraint that they are at an equal distance from the symmetry plane, and the 3D lines connecting them are orthogonal to the symmetry plane (Figure 1.7d). Following the initial algebraic solution, a nonlinear optimization is employed to further refine these 3D estimates by minimizing the reprojection errors (Figure 1.7e).

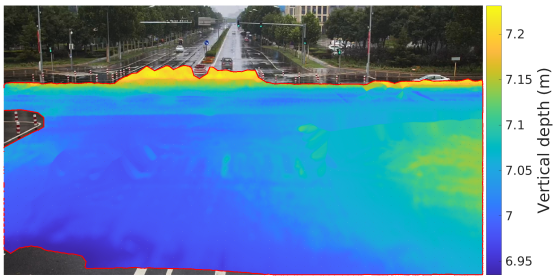
Our method does not require any training and is fully automatic, except for the 2D annotations, which can be cost-effective considering the greater expense of 3D LiDAR-based ground-truthing. It is also interpretable, using bilateral symmetry that is present in most vehicles. The information provided by our ground truth is also richer. The outputs of our ground-truthing method include the 3D locations of the symmetry point pairs and tire-ground contact points (Figure 1.7f).

Depending on the pose of the vehicle, not all dimensions of the vehicle may be determined. A joint Gaussian prior model of vehicle dimensions, based on a vehicle database (Canada, 2023), is employed to estimate maximum *a posteriori* estimates of the remaining dimensions, conditioned upon the observed dimensions. These dimensions, along with predicted 3D points, can then be used to estimate a 3D bounding box (Figure 1.7f).

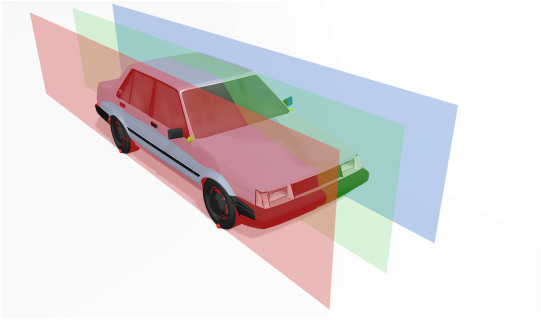
In summary, our main contributions are:



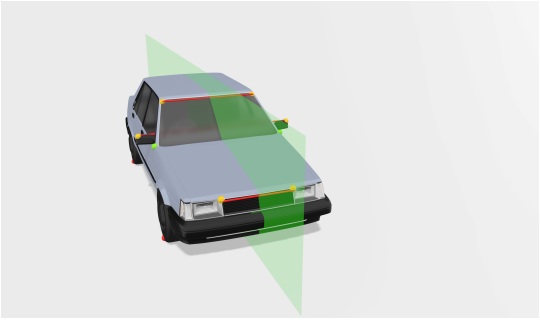
(a) A set of symmetry lines (yellow) in the image can be used to estimate a common vanishing point (blue) on the horizon (red). The intersections (green) may be displaced due to annotation errors. The vanishing point helps establish the orientation of the vehicle's symmetry plane.



(b) Terrain model from accumulated LiDAR point clouds localize tire-ground contact points in 3D.



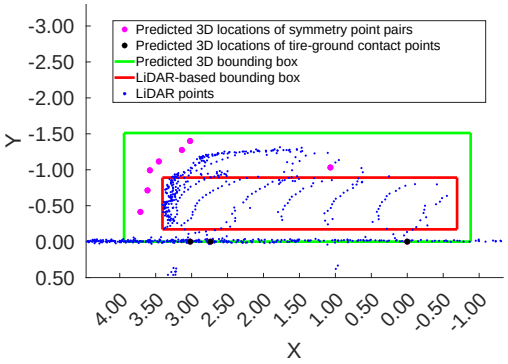
(c) Left extremal plane, symmetry plane, right extremal plane are in red, green, blue respectively. 3D locations of tire-ground contact points and extremal symmetry points localize these planes.



(d) 3D symmetry lines (red) are orthogonal to the symmetry plane (green). Symmetry point pairs (yellow) are at an equal distance from the symmetry plane. These constraints help find the 3D locations of symmetry point pairs.



(e) Annotated points (red) and reprojected points (green) after the nonlinear optimization process to minimize the reprojection errors.



(f) Visualization of the method's outputs of a bounding box and predicted 3D locations along with the 3D LiDAR-based bounding box and LiDAR points.

Figure 1.7: The overall process of symmetry-based monocular 3D vehicle ground-truthing.

- Annotation software to annotate symmetry correspondences and tire-ground contact points.
- YORK-SYM dataset containing annotations of symmetry correspondences and tire-ground contact points for over 2000 vehicles from the DAIR-V2X dataset (Yu et al., 2022). To the best of our knowledge, this is the first attempt at obtaining symmetry annotations for vehicles at traffic intersections, despite the appearance of many infrastructure sensors-captured traffic datasets.
- A highly novel, efficient, and effective LiDAR-free, symmetry-based methodology for acquiring 3D bounding box ground truth for motor vehicles that requires only a single (calibrated) RGB camera.

1.3 Results

To assess this approach, we recruited novice online workers to annotate over 2,000 vehicles using our custom annotation tool and evaluated the inferred 3D object positions, poses, and dimensions against LiDAR-based ground truth. We also compared these results to known vehicle dimensions inferred from confident make and model classification.

Based on the results in chapter 7, our method predicts critical dimensions, such as overall width and overall height, more accurately compared to LiDAR-derived bounding boxes on single frames and comparable to the LiDAR-based method even when their bounding boxes are propagated across frames. LiDAR-derived bounding boxes without propagation are much less accurate when compared to the ground truth make/model dimensions. Regarding vehicle orientation, our method produced very low heading angle errors, demonstrating that symmetry point pairs are a strong constraint that can help predict a vehicle's orientation reliably, even in the far field.

However, camera-LiDAR calibration errors in the dataset can affect the evaluation of our system's performance. Thus, securing high-quality, reliable ground truth is critical for further evaluating the accuracy of our system. Given the significant errors in LiDAR-derived ground truth, in addition to evaluating against known vehicle dimensions inferred from confident make and model classification, we also assessed the consistency of inferred dimensions over time. We found that our predictions of dimensions were very consistent over time. Overall, we found that annotators could rapidly and reliably annotate the symmetry points required for accurate identification of 3D vehicle positions, poses, and dimensions, yielding 3D ground truth that compares favorably to LiDAR-based ground truth, particularly in the far field.

Chapter 2

Related work

2.1 Ground-truthing for 3D object detection

One of the first 3D object detection datasets, KITTI (Geiger et al., 2013), continues to be widely used as a benchmark. Such large-scale datasets require a significant amount of time for ground-truthing, resulting in high costs. The process of 3D ground-truthing involves: selecting a 3D bounding box location, extending the box to cover the object's point cloud, and then adjusting the orientation of the bounding box. To deal with LiDAR's far-field sparsity, annotators often look at corresponding monocular images while locating, scaling, and rotating 3D boxes, ensuring the image of the 3D bounding box aligns well with the corresponding object in the 2D image (Wang et al., 2020). This approach is particularly useful for locating distant objects with sparse LiDAR returns. Figure 2.1 shows a distant vehicle with sparse LiDAR returns. The frustum proposal generated from the 2D bounding box of that vehicle guides the annotators to locate the correct 3D point cloud, as the 3D points must be located within this frustum (Wang et al., 2020). Without the information from an RGB image, localization would be very challenging.

2.1.1 Inter-frame propagation in LiDAR-based ground-truthing

Zimmer et al. (2019) introduced a 3D annotation software, 3D BAT, of which the primary contribution was a semi-automatic labeling method that used a linear model to propagate objects across adjacent frames corresponding to the entire track of a moving vehicle. During this process, annotators selected the beginning and end frames (Figure 2.2a), and then 3D bounding boxes were propagated with small changes to their location, rotation, and scale (Figure 2.2b). A similar approach was adopted by Wang et al. (2020). However, this introduces complexity into the ground-truthing process and still depends upon there being sufficient

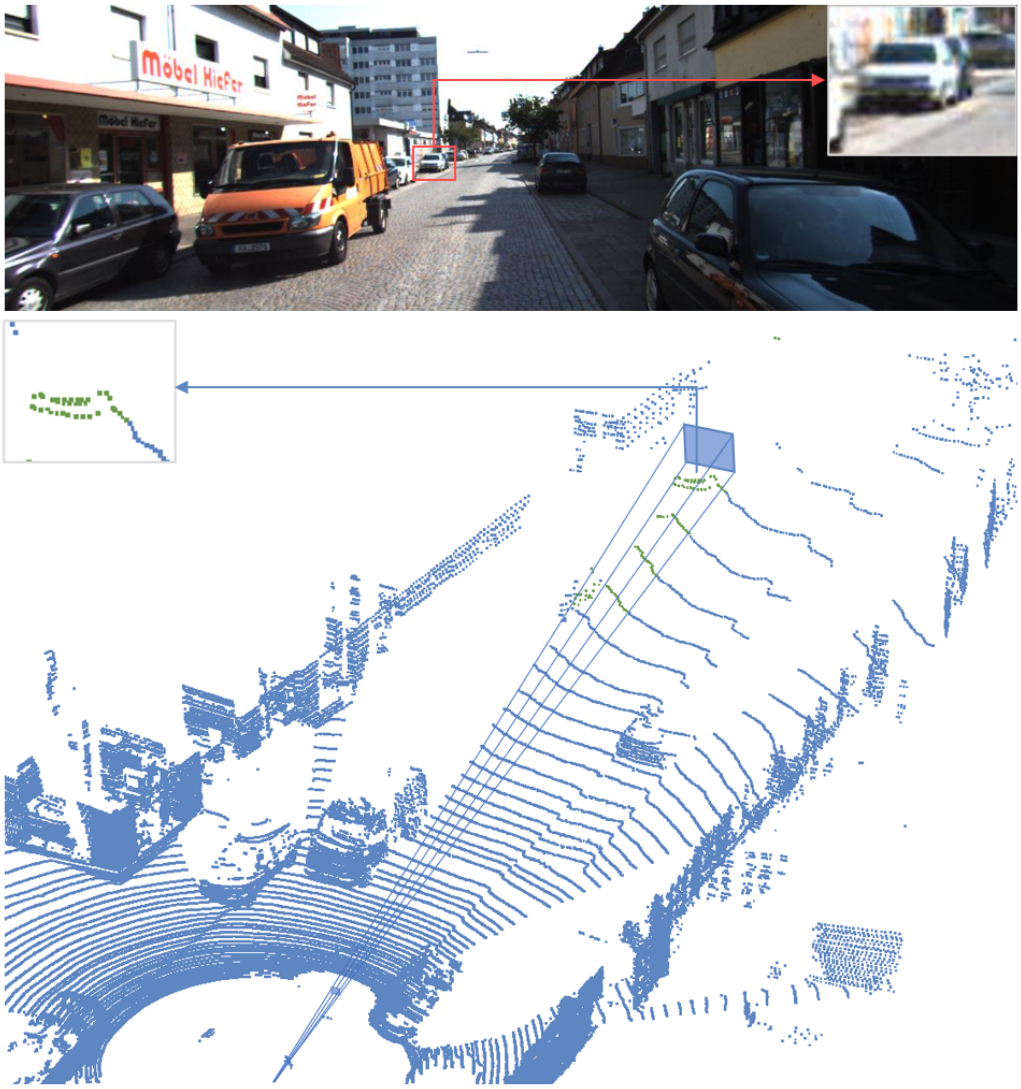
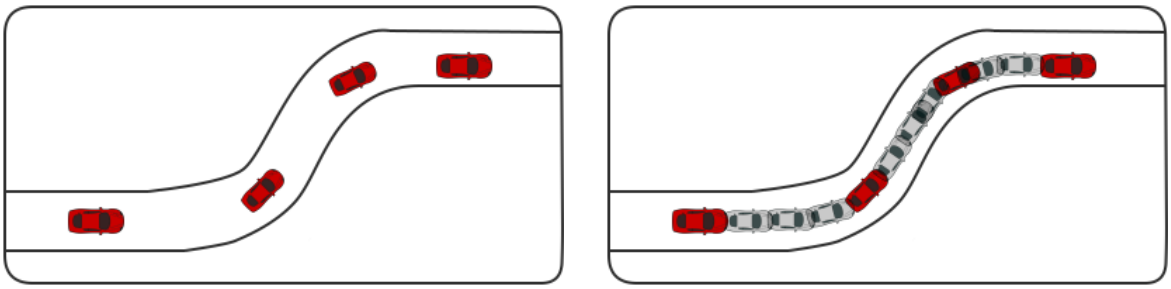


Figure 2.1: The frustum proposal, generated from the 2D bounding box in the RGB image, can help annotators locate distant objects with higher accuracy than by just examining the point cloud alone. This is because sparse LiDAR returns must be contained within this frustum (Wang et al., 2020).



(a) An annotator selects the beginning and end frames and uses control points (red) for propagation.

(b) The entire track of vehicles is shown in gray.

Figure 2.2: 3D bounding boxes propagation across adjacent frames (Zimmer et al., 2019).

LiDAR points in at least one frame to ground-truth the vehicle dimensions.

Companies such as ScaleAI (2023) offer automatic or semi-automatic 3D annotation techniques that also employ propagation methods. However, due to their commercial nature, these companies do not provide detailed explanations of how their propagation methods work.

2.2 Monocular 3D inference in the built environment

2.2.1 Fitting projected cuboids

Relating 2D monocular images to their corresponding 3D built environments can provide crucial information for localizing objects in 3D space. Corral-Soto and Elder (2017) estimated 3D bounding boxes from monocular imagery by fitting projected silhouettes of cuboids to the estimated segmentation of vehicles in the image plane. One limitation of this approach is that motor vehicles are not exactly cuboids (Figure 2.3). This motivates the exploration of more general 3D regularities that might provide more exact monocular constraints.

2.2.2 Parallelism

Parallelism is a characteristic commonly found in the built environment. Man-made objects such as chairs and tables, as well as structures like buildings, contain many parallel lines. Therefore, parallel lines have been a topic of interest in many classical computer vision methods. Under perspective projection, parallel lines in the 3D scene converge at a common vanishing point in the image plane. With a calibrated camera, the location of a vanishing point can be used to identify the orientation of those 3D parallel lines, which provides useful information about the structure of the 3D scene.

To identify such a vanishing point from a group of parallel line segments, Collins and Weiss (1990) used

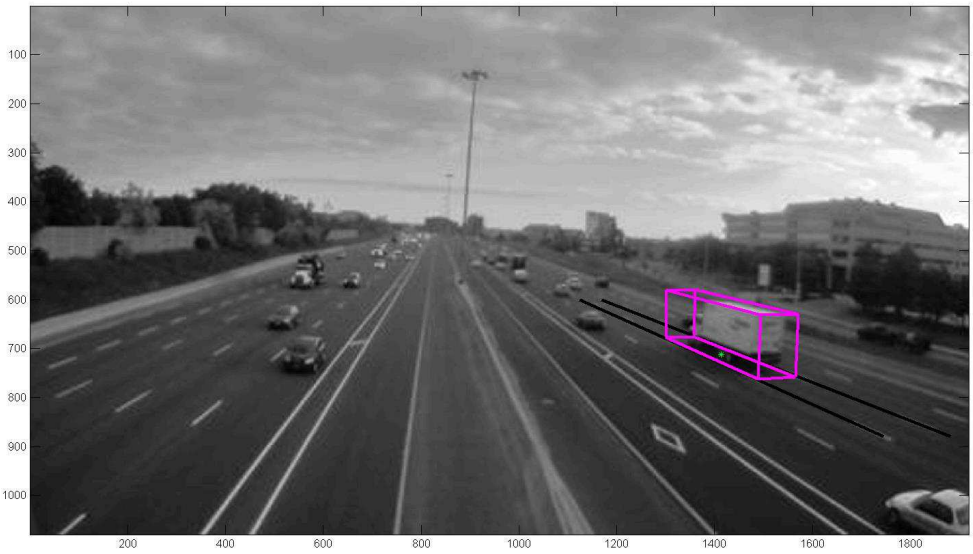


Figure 2.3: The point of view of a camera used by Corral-Soto and Elder (2017) and a candidate cuboid model.

a statistical approach on the Gauss Sphere. Specifically, on the Gauss Sphere, parallel line segments on the image can be thought of as interpretation planes (Figure 2.4), and their intersections with the Gauss Sphere form great circles. The group of normals to these interpretation planes forms an equatorial distribution around a great circle, which the vanishing point vector is orthogonal to (Figure 2.5). Estimation of this equatorial distribution leads to the location of the vanishing point where the parallel line segments converge.

Observing that indoor and outdoor urban structures are often created on a Manhattan 3D grid, Coughlan and Yuille (1999) introduced the Manhattan assumption. This states that parallel line segments in an image can be classified into one of three main orthogonal directions: two horizontal and one vertical. Pixel gradient features were used in a probabilistic model to estimate these directions and recover the camera's orientation with respect to the Manhattan frame of the three mutually orthogonal directions. With the same goal, follow-up work by Denis et al. (2008) replaced pixel gradient features with sparse edge features to estimate the camera's rotation with improved efficiency and accuracy.

For 3D surface reconstruction of buildings that possess Manhattan structures, LS3D (Qian et al., 2019) first classified image edges into one of the three Manhattan directions (Figure 2.6a). Then, using the Gestalt principle of proximity, orthogonal segments were grouped as neighbors. These were then lifted to 3D using Manhattan constraints but the relative depth was still unknown (Figure 2.6b). Through a two-stage constrained optimization process that minimizes the distance between parallel planes that overlap in the image, a 3D scale model was produced (Figure 2.6c). The range map generated by LS3D outperformed deep-learning approaches (Figure 2.6d), which did not take advantage of the Manhattan assumption, demonstrating the

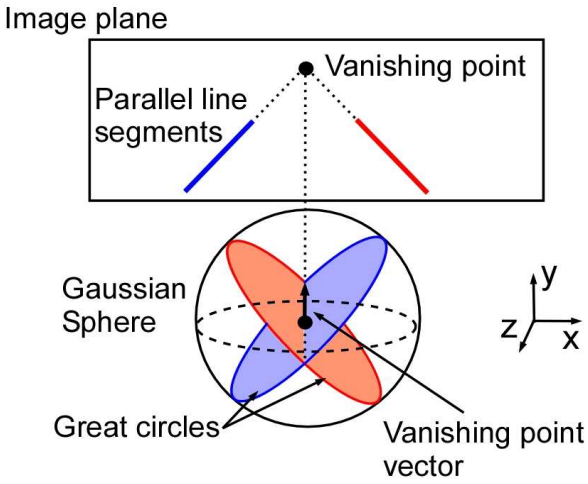


Figure 2.4: Parallel line segments on the image are represented as interpretation planes on the Gauss Sphere. They converge at a common vanishing point. Adapted from Denis et al. (2008).

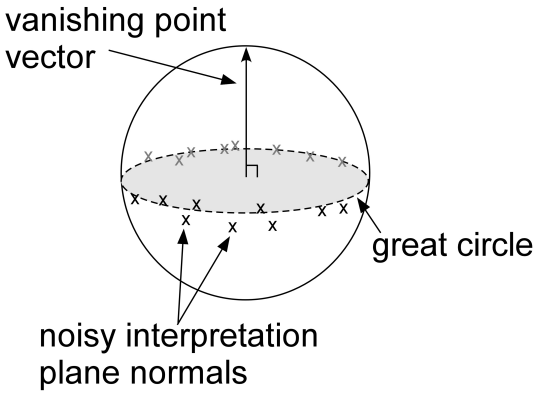
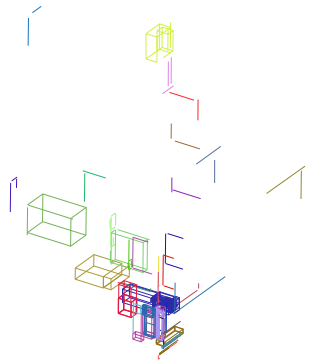


Figure 2.5: The vanishing point vector is orthogonal to the great circle formed by a set of noisy interpretation plane normals. Adapted from Denis et al. (2008).

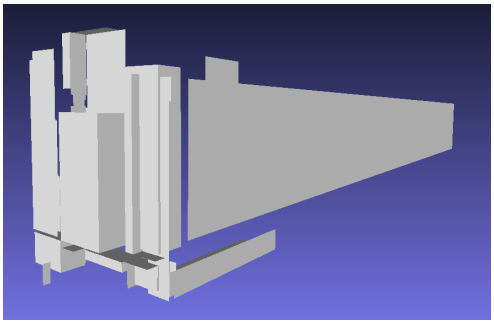
value of including strong geometrical constraints in single-view 3D reconstruction methods.



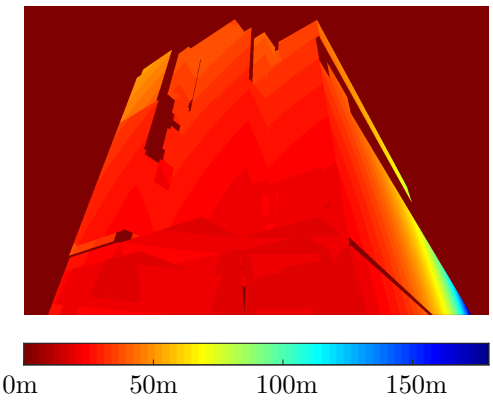
(a) Lines on images are classified into three Manhattan directions.



(b) Line segments are lifted to 3D using Manhattan constraints.



(c) 3D scale model is produced through a two-stage constrained optimization process.



(d) Range map

Figure 2.6: The LS3D algorithm for generating 3D scale models of buildings based on the Manhattan assumption (Qian et al., 2019).

2.2.3 Bilateral symmetry

Motor vehicles possess an even stronger geometric regularity than parallelism or rectilinearity: bilateral symmetry. To introduce the geometry of bilateral symmetry, we can use a chair as an example (Figure 2.7a).

3D symmetry plane

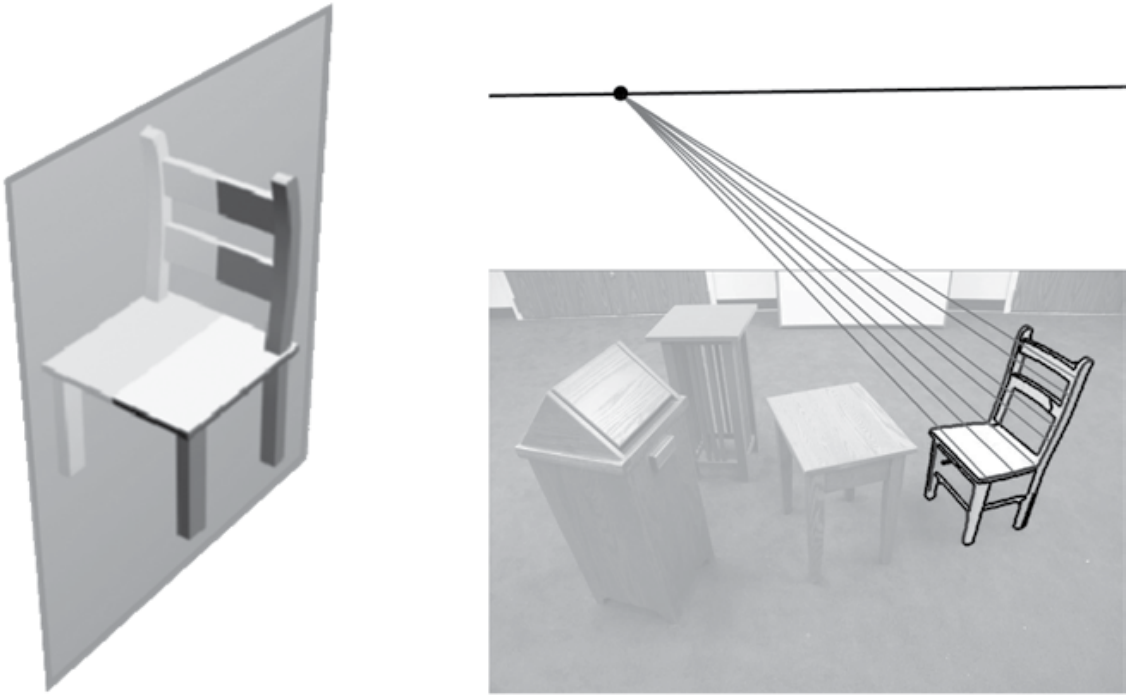
This chair is bilaterally symmetric. There is a single plane of symmetry in 3D that divides the chair into a right half and a left half that mirrors each other (Figure 2.7b).



(a) An example of an object with bilateral symmetry.



(b) The lines connecting symmetry point pairs are orthogonal to the symmetry plane. The symmetry point pairs are also at an equal distance from the symmetry plane.



(c) Symmetry lines converge at a common vanishing point on the horizon.

Figure 2.7: Properties of bilateral symmetry. Adapted from (Pizlo et al., 2014).

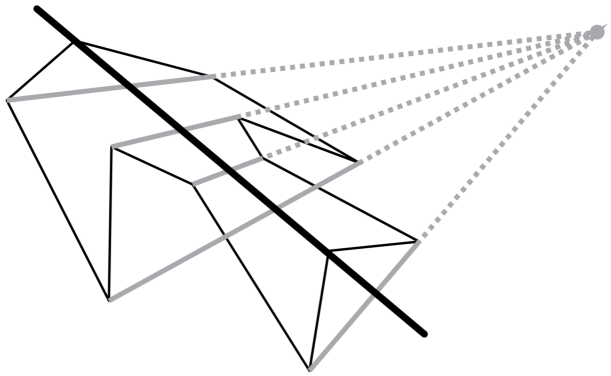


Figure 2.8: A perspective image of a bilaterally symmetric polygon with symmetry lines, which are also parallel, converging at a common vanishing point. Adapted from Pizlo et al. (2014).

Symmetry point pairs and their properties

There exist symmetry point pairs that include two symmetry points: one on the left half and the other one on the right half (Figure 2.7b). In 3D, the lines connecting these pairs are orthogonal to the symmetry plane, and the points in each pair are at an equal distance from the symmetry plane. These are two important constraints that can be used to recover the 3D locations of these symmetry point pairs.

Images of symmetry lines converge at a vanishing point

Since they are all orthogonal to the symmetry plane and thus parallel to each other, when projected on the image, these symmetry lines converge at a common vanishing point (Figure 2.7c). This is an invariant of perspective projection that exists in images from human eyes or a camera. Another example of a bilaterally symmetric figure under perspective projection is shown in Figure 2.8.

These converging symmetry lines provide useful information for estimating a symmetry plane. For a calibrated camera, this vanishing point determines the 3D orientation of the symmetry plane. If the symmetry lines are known to be horizontal, the symmetry plane is vertical and this vanishing point must lie on the horizon.

The challenge of finding symmetry point pairs

Finding symmetry correspondences on images is quite challenging, as explained by Pizlo et al. (2014). The reason is that a perspective image of a 3D symmetric object is typically not symmetric. An image of a 3D symmetric object is symmetric if and only if the viewing camera lies in the symmetry plane, which can be from an infinite number of views (Figure 2.9).



Figure 2.9: An example of degenerate views from Pizlo et al. (2014) that cannot be used to recover 3D shape using a 3D symmetry constraint.

Degenerate views

As shown in Figure 2.9, Pizlo et al. described **degenerate views** that cannot be used to recover the 3D shape using a 3D symmetry constraint. Images of bilaterally symmetric objects are symmetric in these cases. These are views where the camera is in the symmetry plane; hence, we cannot mathematically resolve the relative depth of the symmetry points.

Symmetry is a global feature

Pizlo et al. also described another characteristic of symmetry that symmetry is a **global feature**. As all symmetry point pairs contribute to the determination of the symmetry plane, we can take advantage of this characteristic to turn this problem into an over-determined one if we can identify many symmetry point pairs. Unfortunately, this is very challenging because the search space is four-dimensional: two degrees of freedom come from the vertical symmetry plane (its location and orientation) and the other two degrees of freedom come from identifying 2D symmetry points on an image. Identifying by local appearance whether two points are corresponding symmetry points is difficult because the two points will generally appear different when viewed from slightly different angles.

In our approach, symmetry point pairs are manually annotated. Despite possible annotation errors, we can use these annotations to identify the symmetry plane instead of dealing with the size of the search space. The use of human annotations is different from Pizlo et al. (2014), in which converging symmetry lines on a chair were identified first (Figure 2.7c). While the paper of Li et al. (2012) from the same group explained the approach in more detail, it was still unclear how these symmetry lines were identified for each object in the scene (Figure 2.10). Both our approach and Pizlo et al.'s approach make use of the constraint that images of horizontal symmetry lines converge at a vanishing point on the horizon to identify the orientation of the symmetry plane. To determine the 3D locations of symmetry point correspondences, both approaches



Figure 2.10: Li et al. use symmetry constraints to recover the 3D shapes of objects in an indoor environment.

use the constraints that their corresponding 3D points are equidistant from the symmetry plane and the lines connecting pairs of corresponding 3D symmetry points are orthogonal to the symmetry plane.

2.3 Limitations of prior work

2.3.1 LiDAR sparsity

Lambert et al. (2020) highlighted the limitation of sparse returns from LiDAR units, especially for objects in the far field, even for high-cost units. While inter-frame propagation procedures (Zimmer et al., 2019) can alleviate this issue, they rely on the accuracy of a linear model for the correct propagation of bounding boxes across frames.

RGB images, on the other hand, can provide additional information often overlooked in ground-truthing processes. Wang et al. (2020) showed that in cases with fewer LiDAR returns, annotators were often unsure whether points in the far field belong to a vehicle or which vehicle they belong to. Annotators relied on camera images to fill in missing information and complete the annotation process.

2.3.2 Dependence on straight line segments and the Manhattan assumption

LS3D (Qian et al., 2019) relied on identifying straight line segments from man-made structures such as buildings. For motor vehicles, there are fewer straight-line segments. Therefore, using automatic identification of straight-line segments to find the orientation of a vehicle may not be feasible. Even though some vehicles

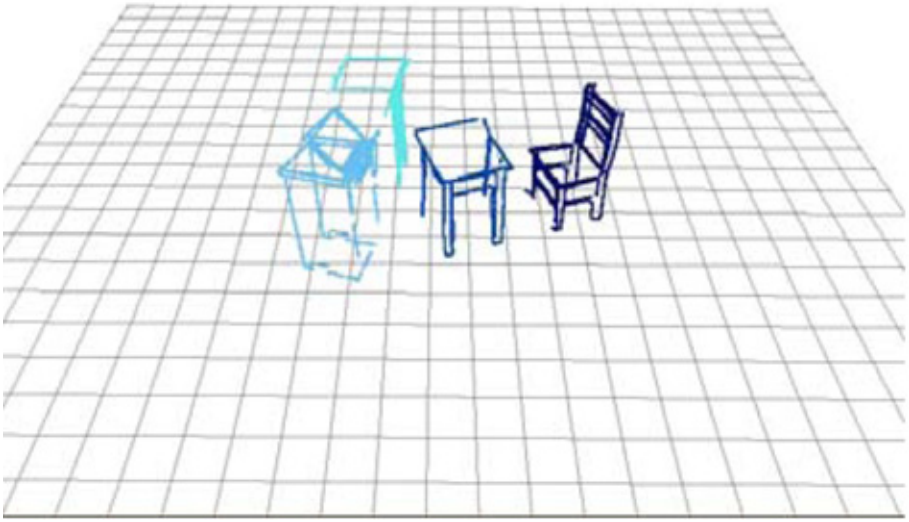


Figure 2.11: Li et al. scaled the objects in 3D by assuming that they rest on an estimated ground plane. Adapted from Li et al. (2012).

possess many straight lines, these lines may not easily be classified into the three main Manhattan directions: two horizontal and one vertical.

2.3.3 Restriction to cuboid estimation

The output of Corral-Soto and Elder (2017) is 3D cuboid (Figure 2.3). While these are useful for evaluating 3D object detection methods, a vehicle's characteristics extend beyond a simple bounding box. The 3D locations of symmetry points and tire-contact points can provide additional information useful for fine-grain classification of vehicle make and model.

2.3.4 Absolute scale problem

LS3D (Qian et al., 2019) did not address the problem of absolute scaling of structures. To solve the absolute scale problem, Li et al. (2012) identified 3D points belonging to the ground in stereo images and then fit a plane to these points (Figure 2.11). The absolute object scale was determined by assuming objects resting on the ground plane (Li et al., 2012). Objects with occluded bottom parts were the only ones that could not be properly scaled. Instead of stereo, we assume known camera intrinsics and extrinsics and then use annotated tire-ground contact points to scale vehicles in 3D. Instead of using a planar surface as in Li et al. (2012), employing a high-resolution terrain model can increase accuracy.

2.3.5 Methods to automatically detect symmetry point pairs from 2D images

We are aware of some methods that can detect symmetry from 2D images automatically. These methods apply to datasets such as ShapeNet (Chang et al., 2015) and Pix3D (Sun et al., 2018). Zhou et al. (2021) builds an end-to-end network to propose 3D symmetry planes from just 2D images. The main assumption the authors make is that correspondences in 3D, after being projected onto a 2D image, should maintain an intra-pixel relationship, which means symmetrical points with respect to the symmetry plane should have similar surface properties such as material and texture. Therefore, after extracting features from images, given a potential symmetry plane normal, Zhou et al. (2021) uses a cost volume network to measure how good one particular candidate is by evaluating pixel-wise correspondences. If the proposed symmetry plane is a good one, then there will be a correspondence signal at a particular depth for most unoccluded points, resulting in a lower cost generated from that cost volume. From this cost volume, Zhou et al. (2021) then converts it into a confidence score and optimizes it end-to-end so that the symmetry plane closest to the ground truth symmetry plane has the highest confidence score. Lin et al. (2021), a more recent work, outperforms Zhou et al. (2021). In this study, for real-world images of Pix3D (Sun et al., 2018), angle accuracy AA@1 degree is 27.5, AA@5 degrees is 53.0, and AA@10 degrees is 62.8. This means that these automated methods are not good enough for our symmetry-based method because our symmetry-based method produces very low heading angle errors of only 2-4 degrees. Also, the above methods are trained and tested on synthetic or pre-segmented object datasets and thus do not have to handle background or occlusions.

Chapter 3

Dataset

To explore the potential of using symmetry constraints for monocular ground-truthing of 3D vehicle detection, we require a dataset consisting of images of vehicles in realistic driving scenarios, along with paired LiDAR point clouds and annotated 3D LiDAR-based bounding boxes.

Until recently, available LiDAR datasets for 3D object detection primarily focused on LiDAR units mounted on vehicles, such as KITTI (Geiger et al., 2013), nuScenes (Caesar et al., 2020), and Waymo (Sun et al., 2019). Refer to Figure 3.1 for an example setup. These datasets are more relevant to autonomous driving rather than traffic analytics. However, more recently, datasets captured by sensors mounted in city and highway infrastructure have emerged.

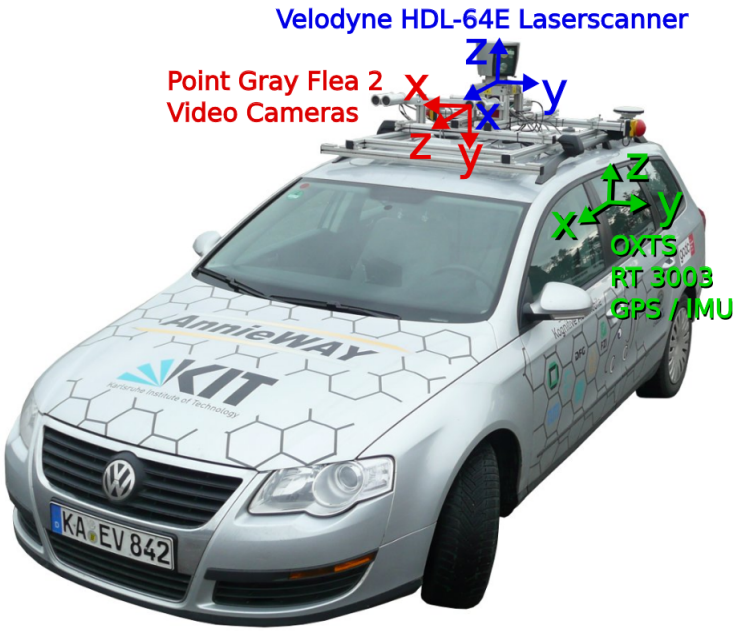


Figure 3.1: Setup of cameras and a LiDAR unit for the KITTI dataset (Geiger et al., 2013).

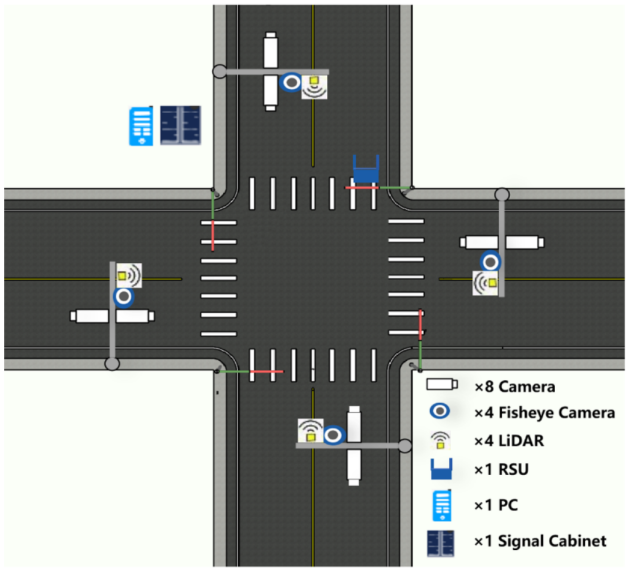


Figure 3.2: Setup of cameras and LiDAR units for the DAIR-V2X dataset (Yu et al., 2022). RSU stands for Road-side Unit.

Table 3.1: Specifications of cameras and LiDAR units used in the DAIR-V2X dataset (Yu et al., 2022).

Sensor	**Specifications**
LiDAR	300 beams, 10Hz capture frequency, 100°horizontal FOV, -30° to 10°vertical FOV, $\leq 280m$, $\pm 3cm$ accuracy
Camera	RGB, 25Hz capture frequency, 1920x1080 resolution, JPEG compressed. FOV is not provided.

One such dataset is DAIR-V2X (Yu et al., 2022), which covers traffic intersections in Beijing, China. For our research, we work with the DAIR-V2X-C subset of this dataset, as it provides the necessary timestamps for the camera and LiDAR sensor. DAIR-V2X-C combines data from only one camera and one LiDAR unit while there are additional units available in the test site (Figure 3.2). Using the timestamp difference between the sensors, we can minimize errors caused by imperfect sensor synchronization.

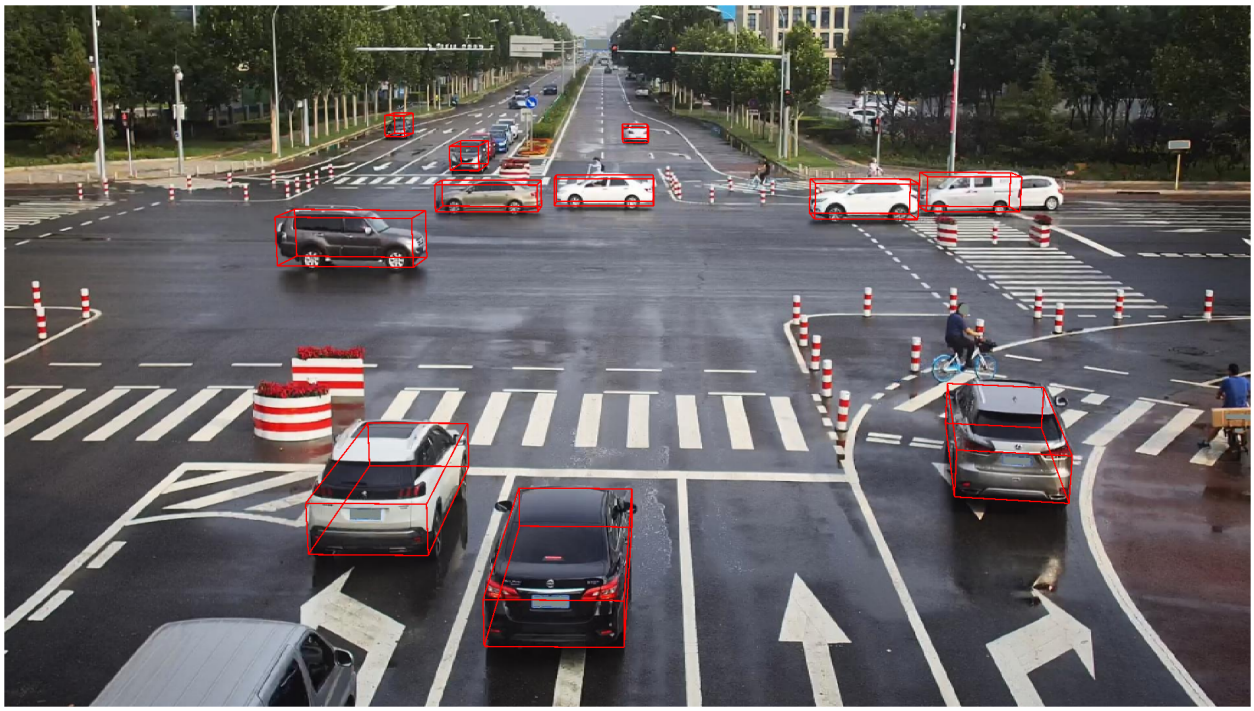
The specifications of the sensors used in the dataset are shown in Table 3.1. Although Figure 3.2 shows 12 cameras and 4 LiDAR units, DAIR-V2X only provides outputs from one camera and one infrastructure LiDAR unit (Yu et al., 2022). DAIR-V2X-C contains frames from 6 different intersections, with the number of frames varying across intersections. Representative frames from the dataset are shown in Figure 3.3.

The dataset provides both 2D and 3D LiDAR-based bounding boxes, manually annotated by humans. However, no information is provided regarding the method used to annotate bounding boxes for vehicles with sparse LiDAR returns. Additionally, the dataset includes LiDAR point clouds for each frame, which we use to build our terrain model (see section 5.2). Visualizations of 3D bounding boxes and a LiDAR point cloud for a frame are shown in Figure 3.4.

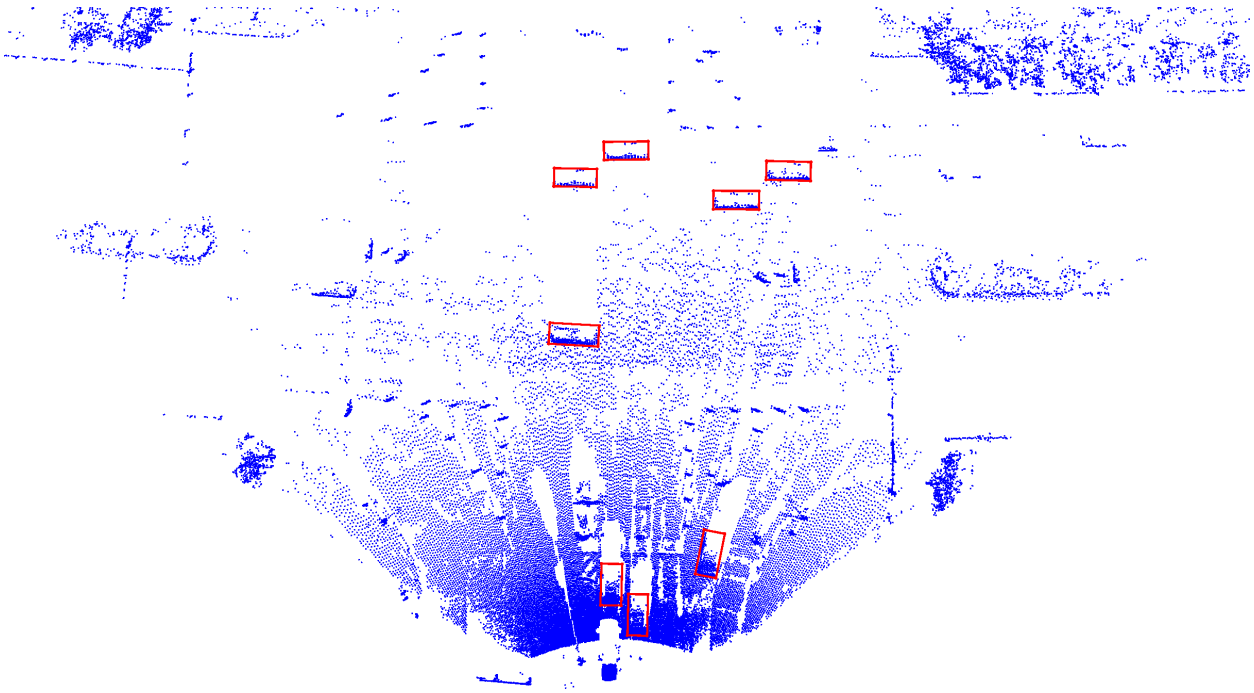
To create our dataset, we selected frames from all 6 intersections that met the following criteria: the



Figure 3.3: Representative frames from the DAIR-V2X-C subset of the DAIR-V2X dataset (Yu et al., 2022).



(a) 3D bounding boxes projected onto a monocular image.



(b) LiDAR point cloud with 3D LiDAR-based bounding boxes.

Figure 3.4: Example of a monocular image and its corresponding point cloud in the DAIR-V2X dataset (Yu et al., 2022).

difference between the LiDAR timestamp and the camera timestamp should be less than 9 milliseconds. Assuming a maximum vehicle speed of 50 km/h at these intersections, the distance error caused by sensor synchronization should be within 12.5 cm.

From the selected frames, we further filtered the vehicles based on the following requirements:

- Belonged to the Car, Van, or Bus classes.
- Were not truncated or occluded.

For truncation, the dataset has three possible labels: 0, 1, and 2, indicating three types of truncation: non-truncated, transversely truncated, and longitudinally truncated respectively. As for occlusion, the dataset also has three possible labels: 0, 1, and 2, indicating three types of occlusion: fully visible, partly occluded, and largely occluded respectively.

Although the dataset does not provide tracking information, having annotations of the same vehicle traversing through an intersection is highly valuable for evaluation purposes (see chapter 7). To address this, we employed the Du et al. (2022)'s multi-object tracker, StrongSort, with the Yolov8 (Jocher et al., 2023) object detection model to generate tracks. Human annotators were then asked to verify the validity of candidate tracks. Annotators were presented with a series of images stitched together based on a candidate track (Figure 3.5) and were required to determine if it represented a valid track.

In addition to these requirements, we introduced an additional criterion based on the tracking information: vehicles on the same track had to be at least 3 meters apart to be included. This criterion aimed to improve the diversity of annotated vehicles in terms of their location and pose. An example of a vehicle track passing through an intersection is shown in Figure 3.5.

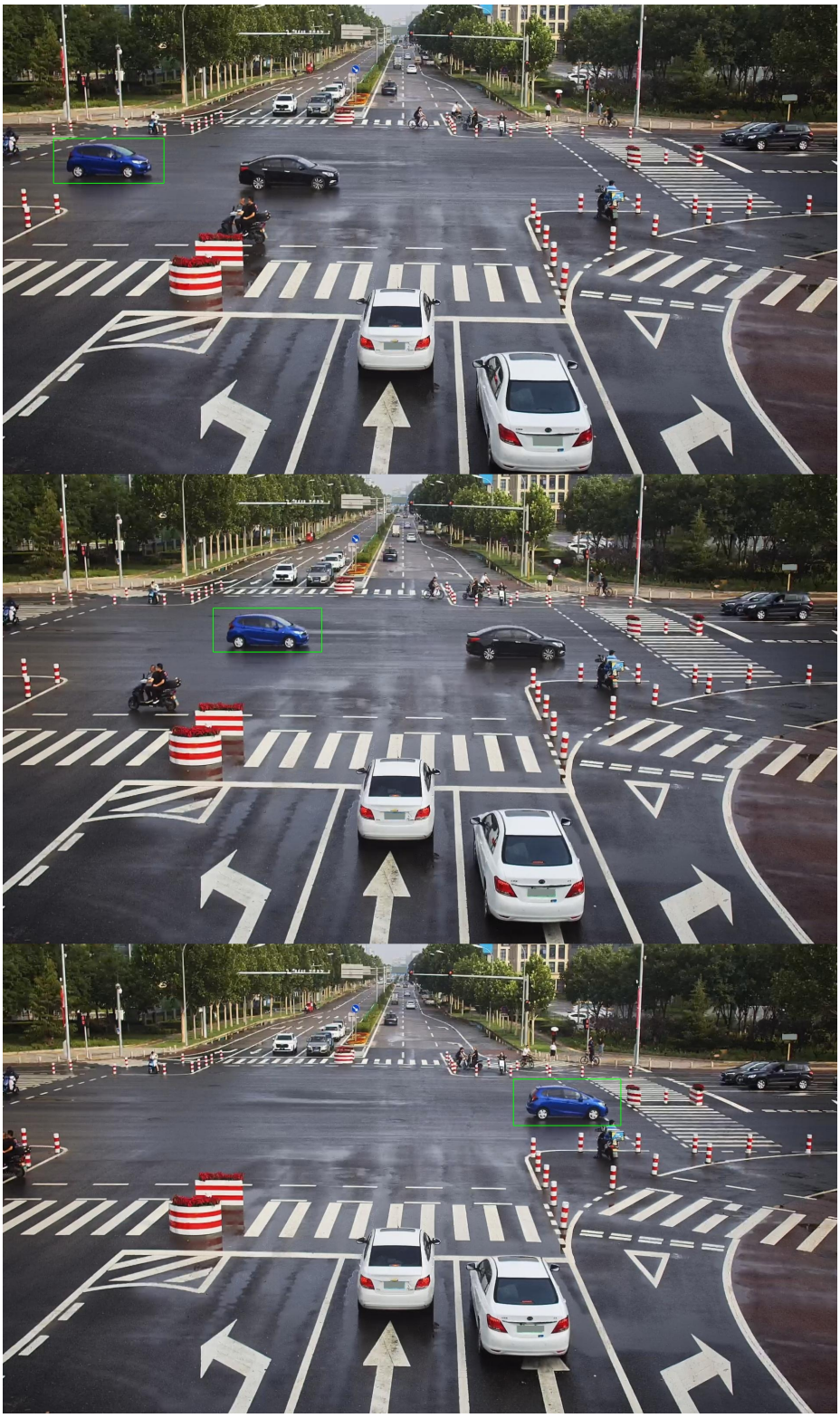


Figure 3.5: An example of a track verified by annotators in the DAIR-V2X dataset (Yu et al., 2022).

Chapter 4

Annotation GUI

To obtain 2D annotations, we developed our own annotation GUI using CVAT (CVAT.ai, 2023) to enable annotators to annotate on a web platform (Figure 4.1). The code for this GUI will be made available on our lab's repository. There are two types of points that need to be annotated, as defined below.

4.1 Tire-ground contact points

Four types of tire-ground contact points can be annotated: Driver front, Driver rear, Passenger front, and Passenger rear (Figure 4.2).

Tire-ground contact points are points on a 2D image that correspond to the points in 3D where the outer edge of a tire touches the ground surface, as illustrated in Figure 4.3. We chose the outer edge as it is typically more visible from most viewpoints.

Depending on the location and pose of a vehicle, some of these types may not be visible. We require at least one tire-ground contact point for our method: if no tire-ground contact points are directly visible, annotators are asked to estimate the location of at least one point. An example of the annotation of actual vehicle tire-ground contact points can be seen in Figure 4.4.

4.2 Symmetry point pairs

Symmetry point pairs are 2D image projections of a 3D point on the body of a vehicle and its mirror reflection across the symmetry plane. These symmetry points come in pairs: one on the left side and another on the right side of the vehicle. In 3D, the line connecting a pair of symmetry points is orthogonal to the vehicle's symmetry plane and provides crucial information about the vehicle's pose.

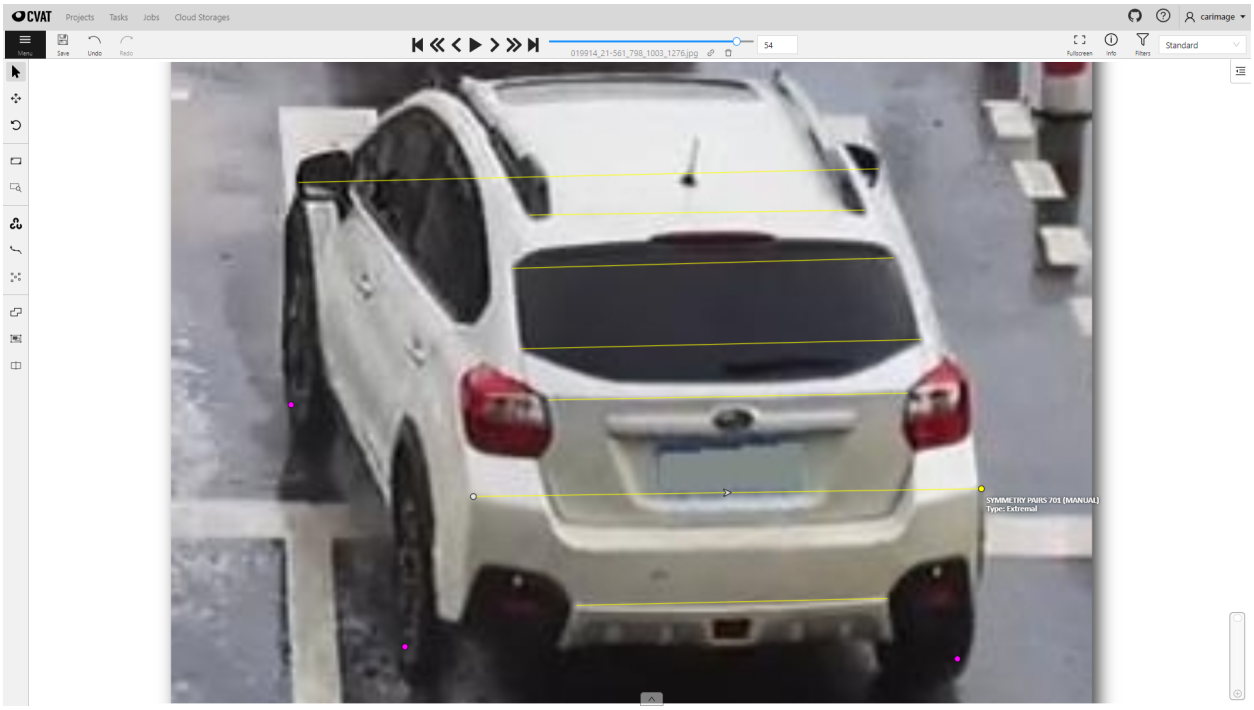


Figure 4.1: Annotation GUI was created using CVAT (CVAT.ai, 2023). Symmetry point pairs are indicated by yellow line segments. An extremal symmetry point pair is highlighted with circle endpoints (white endpoint on the driver's side, yellow endpoint on the passenger's side) and a text label. Tire-ground contact points are magenta.

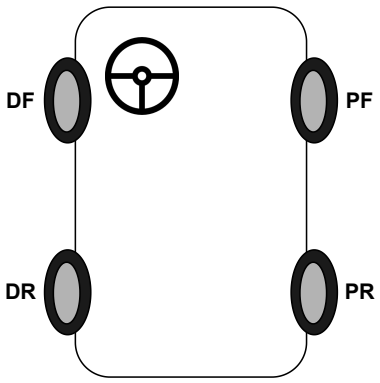


Figure 4.2: Locations of the four types of tire-ground contact points on a top view of a vehicle's body: Driver front (DF), Driver rear (DR), Passenger front (PF), and Passenger rear (PR).



Figure 4.3: Tire-ground contact points represent the locations on the outer edges of a vehicle where the tires touch the ground surface (red points). 3D model provided by remtromol (2021).



Figure 4.4: Annotations of three tire-ground contact points (magenta) for Driver front (DF), Driver rear (DR), and Passenger rear (PR).

For each symmetry point pair, as shown in Figure 4.5, annotators are required to draw a pair of points: one corresponding to the driver's side and the other corresponding to the passenger's side of the vehicle. They are also asked to annotate only points with distinguishing features or markings such as corners of the windshield, edges of the side-view mirrors, and headlights.

4.2.1 Extremal symmetry point pairs

For each vehicle, one or more symmetry point pairs can be designated as **extremal** symmetry point pairs. These symmetry point pairs align vertically with the outer edges of the tires, as depicted in Figure 4.5. At least one extremal symmetry point pair is required to locate the symmetry plane in cases where we do not have tire-ground contact points annotated on both sides of the vehicle (see subsection 5.4.2).

The remaining symmetry point pairs are referred to as **non-extremal** symmetry point pairs. If both side-view mirrors are visible, annotators can also designate a **mirror** symmetry pair. An example of symmetry point annotations is shown in Figure 4.6.



Figure 4.5: Symmetry point pairs are marked with yellow spheres. Extremal symmetry point pairs (green) are aligned vertically with the outer edges of a vehicle's tires. Tire-ground contact points (red) are also at the outer edges of a vehicle's tires. 3D model provided by remtromol (2021).

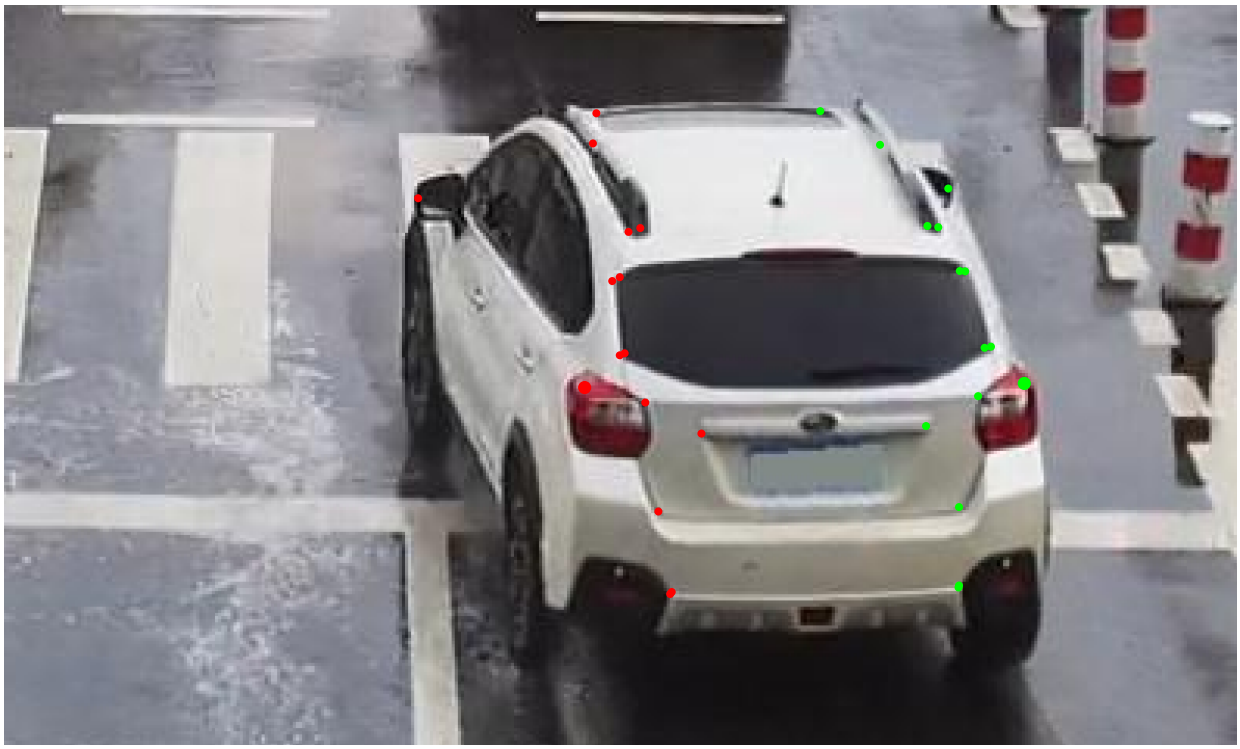


Figure 4.6: Symmetry points are indicated by red points on the driver's side and green points on the passenger's side. An extremal symmetry point pair is marked with a bigger size.

4.3 Time Spent on Annotations

Information on how long LiDAR-based 3D ground truthing takes for each dataset is rarely available. For instance, the authors of the KITTI (Geiger et al., 2013) dataset did not report the time required to annotate its object detection benchmark. In contrast, statistics from the authors of the SUN RGB-D dataset (Song et al., 2015) indicate that annotating each object instance takes approximately 114.3 seconds. It is important to note that the SUN RGB-D dataset (Song et al., 2015) contains annotations for indoor objects only. Our annotators required a similar duration, roughly 120 seconds per vehicle.

4.4 YORK-SYM Dataset

The result of the data curation process for the evaluation of our ground truthing method is a dataset referred to as YORK-SYM, which includes annotations for over 2000 vehicles. This dataset will be made available on our lab's project website.

Chapter 5

Inferring 3D points from 2D annotated points

The second part of our method is to infer the 3D locations of all of the annotated 2D points based on geometric constraints of a bilaterally symmetric object.

We can break down the process of inferring 3D locations of 2D points into three main steps:

- Estimation of symmetry plane orientation
- Estimation of symmetry plane location
- Estimation of 3D locations of symmetry points pairs

5.1 Camera projection

The general projection formula is:

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = K[R|t] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{5.1}$$

(X, Y, Z) is a point in 3D space, expressed in homogenous coordinates as $(X, Y, Z, 1)$. A homogeneous image point, $\tilde{p}$, expressed as $(\lambda x, \lambda y, \lambda)$ can be transformed to the image pixel coordinate of (x, y) by dividing by the third coordinate λ. $\bar{p}$ is the augmented form of the image point and is represented as $(x, y, 1)$. As λ

can take any value different from 0, a point in a 2D image can be the projection of infinitely many points in 3D. K is the intrinsic matrix. R and t are the rotation and translation of the **World coordinate frame** with respect to the **Camera coordinate frame**.

5.1.1 Coordinate frames

To simplify this projection equation, we can have the **World coordinate frame** centered at the origin of the **Camera coordinate frame**. Then translation t will be zero and the projection equation is simplified to:

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = KR \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{5.2}$$

The **Camera coordinate frame** is a right-handed coordinate frame with X-axis aligned with the horizontal axis of the image and pointing to the right, Y-axis aligned with the vertical axis of the image and pointing down, and Z-axis pointing away from the camera. Because traffic cameras are usually looking down toward the road (Figure 5.1), the Y-axis of the **Camera coordinate frame** is not orthogonal to the ground. The **World coordinate frame** is a right-handed coordinate frame with Y-axis aligned with gravity and pointing down to the ground. The X-Z plane of the **World coordinate frame** is horizontal.

For the DAIR-V2X dataset (Yu et al., 2022), there is also a **LiDAR coordinate frame**, which is centered at the LiDAR unit and has the Z-axis pointing up (opposite to gravity). The LiDAR unit is mounted next to the camera but has a different viewing angle. The transformation from a 3D point in the **LiDAR coordinate frame** to a 3D point in the **World coordinate frame** is:

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = [R|t]_{\text{LiDAR_to_world}} \begin{bmatrix} U \\ V \\ W \\ 1 \end{bmatrix} \tag{5.3}$$

where $[R|t]_{\text{LiDAR_to_world}}$ rotates and translates a 3D point (U, V, W) expressed in **LiDAR coordinate frame** into a **World coordinate frame** point (X, Y, Z) (Figure 5.1).

The LiDAR-based 3D bounding boxes of the DAIR-V2X dataset (Yu et al., 2022) are expressed in the **LiDAR coordinate frame**.

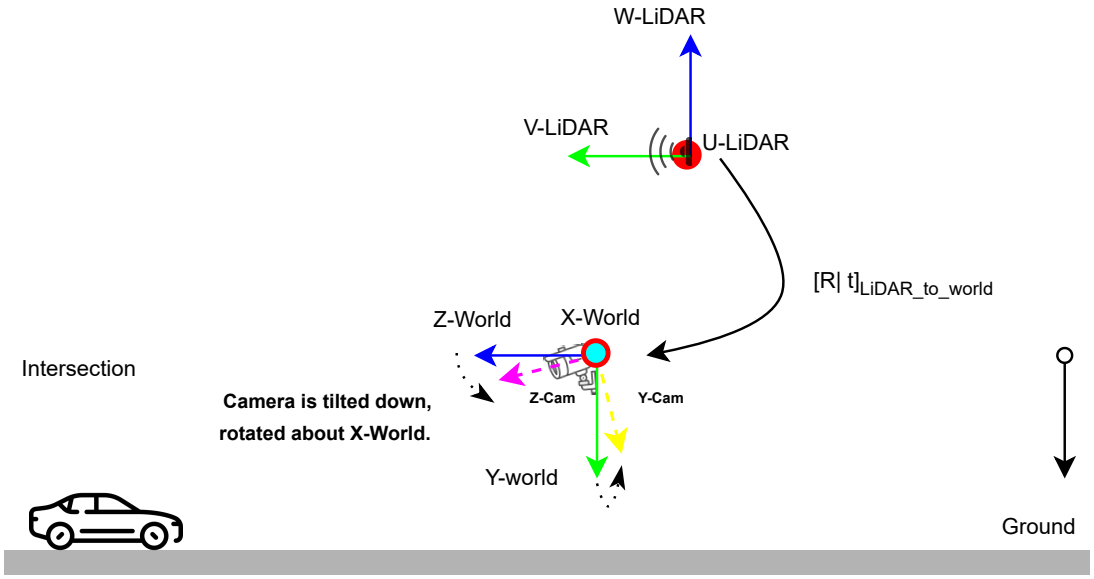


Figure 5.1: The camera is tilted down toward the road, showing the relationship between the **World coordinate frame** (Red-Green-Blue axes) and the **Camera coordinate frame** (Cyan-Yellow-Magenta axes). The figure also shows the transformation from the **LiDAR coordinate frame** to the **World coordinate frame** for the DAIR-V2X dataset (Yu et al., 2022).

5.2 Terrain model

Accurate localization of a vehicle is crucial to ground-truthing for both object detection and traffic analytics purposes. A 3D terrain model can be used to find the 3D locations of tire-ground contact points by finding the nearest 3D points to the 3D optical ray through each annotated tire-ground contact point.

The 3D location of a tire-ground contact point can be found by back-projection:

$$\mathbf{T}'_i = \lambda_{t'_i}(KR)^{-1}\bar{\mathbf{t}}'_i \tag{5.4}$$

where $\bar{\mathbf{t}}'_i = (x_i^{t\prime}, y_i^{t\prime}, 1)$ is the augmented form of an annotated tire-ground contact point $\mathbf{t}'_i$ and $\mathbf{T}'_i = (X_i^{t\prime}, Y_i^{t\prime}, Z_i^{t\prime})$ is the corresponding 3D location on a terrain model. To compute the scaling parameter $\lambda_{t'_i}$, we look up $Y_i^{t\prime}$ from the corresponding 2D pixel $\left(x_i^{t\prime}, y_i^{t\prime}\right)$ on the image using a raster containing the vertical depths of all pixels within a region of interest. We will explain how we derive this raster below.

For the DAIR-V2X dataset (Yu et al., 2022), we derive a terrain model by creating a raster of corresponding vertical depth Y for each pixel (Figure 5.2) from the provided 3D LiDAR point clouds by employing a **k-nearest neighbors algorithm** (This was required due to the unknown geographic location of the DAIR-V2X dataset (Yu et al., 2022). We note that most urban areas are densely mapped using airborne LiDAR

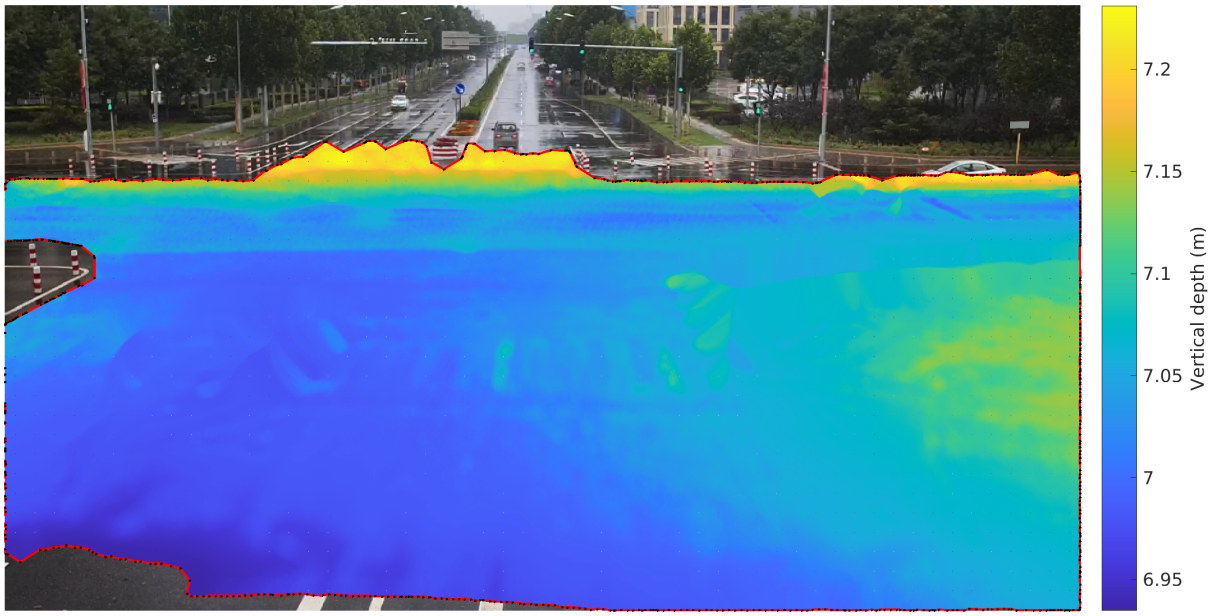


Figure 5.2: Raster of vertical depth Y_i for each pixel (x_i, y_i) in the region of interest. The region of interest, ROI, is highlighted by a red boundary.

(Chen et al., 2017) and/or photogrammetry and that good quality terrain models are generally available.). In any realistic deployment, the coordinates of an intersection are typically known and can be used to access the appropriate online terrain maps. Moreover, while only sparse LiDAR is needed to estimate the terrain map, dense LiDAR is required for 3D object detection. Many online sources offer global terrain data; for instance, 30m resolution global data is available from the U.S. Geological Survey (2024). Also, L3Harris Technologies (2024) provides higher resolution (2.5 – 5m) global data.

For each of the 6 intersections in the subset DAIR-V2X-C of this dataset, the first step of building a terrain model is to accumulate terrain LiDAR points across all the frames of the same intersection. We remove all the LiDAR points that belong to vehicles and infrastructure such as traffic poles and barriers. We use the 3D LiDAR-based bounding boxes to remove points from vehicles and a hand-crafted mask on 2D images to remove points from fixed infrastructures (Figure 5.3).

We found that most of the far-field LiDAR returns did not project from the ground but rather from moving vehicles (Figure 5.4). Since these points are not included in the 3D LiDAR-based bounding boxes, we adjusted the mask to eliminate these points (Figure 5.3). Only vehicles with the annotated tire-ground contact points located within this ROI (Figure 5.2) are evaluated.

As we observe that an accumulated LiDAR point cloud has different densities at different ground locations (Figure 5.5), we figure out the height Y_i^t corresponding to 2D pixel (x_i^t, y_i^t) by averaging the height of k nearest

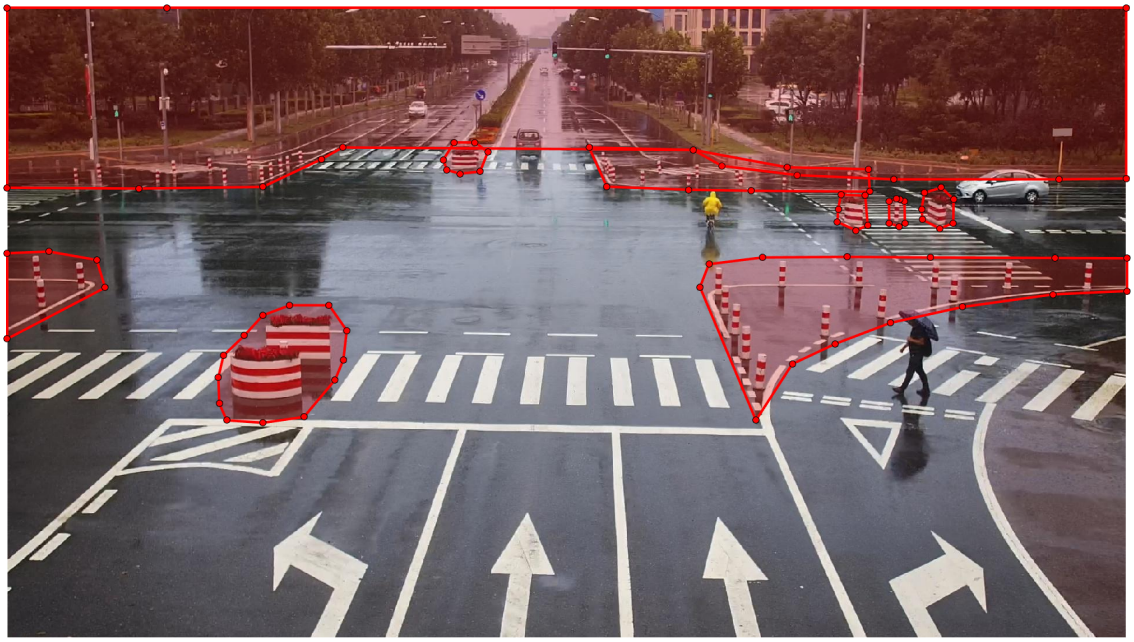


Figure 5.3: Hand-crafted image masks are employed to eliminate LiDAR returns from non-terrain infrastructure and vehicles in the far field. These are employed only to form the terrain model for the intersection.



Figure 5.4: LiDAR returns in the far field for this intersection of the DAIR-V2X dataset (Yu et al., 2022) contain points from vehicles' bodies. Notice the projected LiDAR points on the three vehicles in the far field.

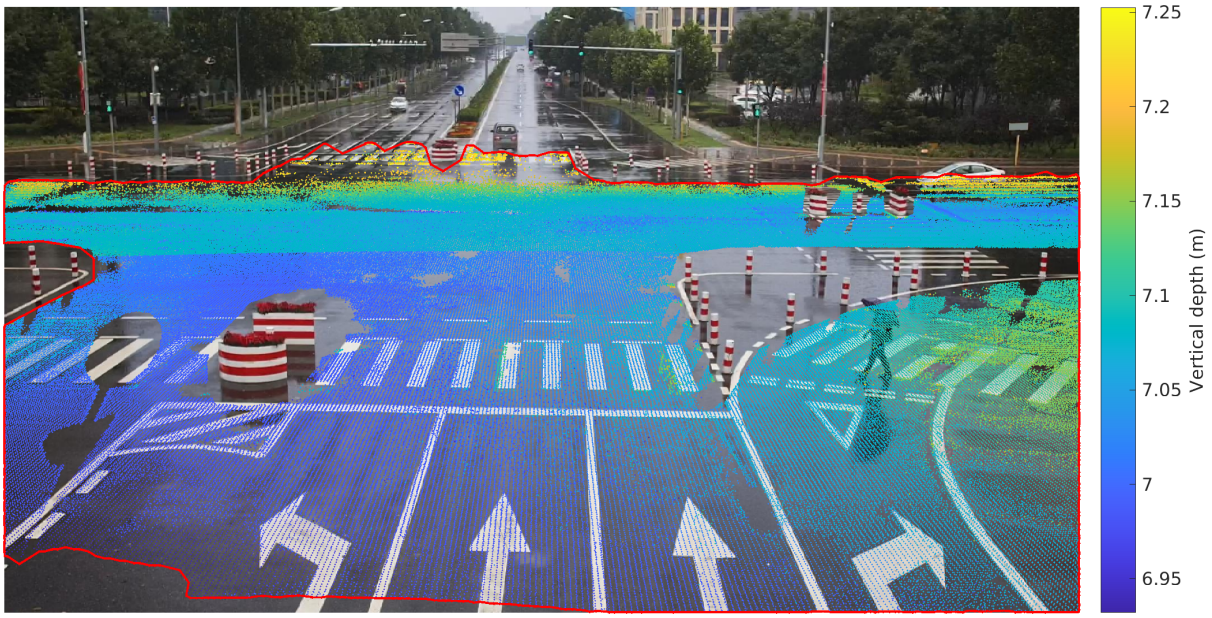


Figure 5.5: Accumulation of LiDAR points across frames of one of the intersections of the DAIR-V2X dataset (Yu et al., 2022) within the ROI (red) defined by the masking procedure.

3D points to its reprojection ray $\mathbf{l}_i$:

$$\mathbf{l}_i = (KR)^{-1}\bar{\mathbf{t}}_i \tag{5.5}$$

where $\bar{\mathbf{t}}_i = (x_i, y_i, 1)$ is the augmented form of a pixel $\mathbf{t}_i$. We use leave-one-out cross-validation to figure out the optimal k to use. Repeating this process for all pixels in the ROI, we will have a raster to look up Y values of vertical depth easily (Figure 5.2).

5.3 Estimating the orientation of the symmetry plane

Since we assume a fully-calibrated camera, we have the intrinsic matrix K and rotation matrix R. Using this information, we can find the vanishing points $\tilde{\mathbf{v}}_X$ and $\tilde{\mathbf{v}}_Z$ corresponding to the axes X and Z of the **World coordinate frame**:

$$\tilde{\mathbf{v}}_X = KR_1$$

$$\tilde{\mathbf{v}}_Z = KR_3$$

where R_1 and R_3 are the first and third columns of the rotation matrix R.

We then can find the horizon line. A line in homogeneous coordinates is represented by the cross product



Figure 5.6: Horizon line (in red) for a frame of the DAIR-V2X dataset (Yu et al., 2022).

of two homogeneous points that it connects:

$$\tilde{\mathbf{l}}_h = \tilde{\mathbf{v}}_X \times \tilde{\mathbf{v}}_Z \tag{5.6}$$

where $\tilde{\mathbf{l}}_h$ is the horizon line in homogeneous coordinates. A visualization of a horizon line is shown in Figure 5.6.

We let $\mathbf{p}' = \left\{\mathbf{p}'_1, \ldots, \mathbf{p}'_{n_p}\right\}$ be a set of n_p annotated symmetry point pairs. Each point pair $\mathbf{p}'_i$ consists of two points equally displaced to the left and to the right of the symmetry plane: $\mathbf{p}'_i = \{\mathbf{p}'_{il}, \mathbf{p}'_{ir}\} = \left\{\left(x_{il}^{p'}, y_{il}^{p'}\right), \left(x_{ir}^{p'}, y_{ir}^{p'}\right)\right\}$. A pair of symmetry points defines a symmetry line:

$$\tilde{\mathbf{l}}'_i = \tilde{\mathbf{p}}'_{il} \times \tilde{\mathbf{p}}'_{ir} \tag{5.7}$$

where $\tilde{\mathbf{l}}'_i$ is the symmetry line for pair i in homogeneous coordinates.

For a vehicle with zero pitch and roll and in the absence of annotation error, the set of symmetry lines $\left\{\tilde{\mathbf{l}}'_1, \ldots, \tilde{\mathbf{l}}'_{n_p}\right\}$ will intersect at a common vanishing point $\tilde{\mathbf{v}}_\mathrm{s}$ on the horizon. In practice, however, each symmetry line $\tilde{\mathbf{l}}'_i$ will, in general, intersect with the horizon line at some point $\tilde{\mathbf{v}}'_\mathrm{i} = \tilde{\mathbf{l}}'_i \times \tilde{\mathbf{l}}_h$ that is displaced from $\tilde{\mathbf{v}}_\mathrm{s}$ due to annotation error (Figure 5.7).

From this set of symmetry lines $\left\{\tilde{\mathbf{l}}'_1, \ldots, \tilde{\mathbf{l}}'_{n_p}\right\}$ in the image, to estimate $\tilde{\mathbf{v}}_\mathrm{s}$, we follow the method used by Li et al. (2012) to find the point $\tilde{\mathbf{v}}_\mathrm{s}^*$ on the horizon that minimizes the sum of squared distances to the

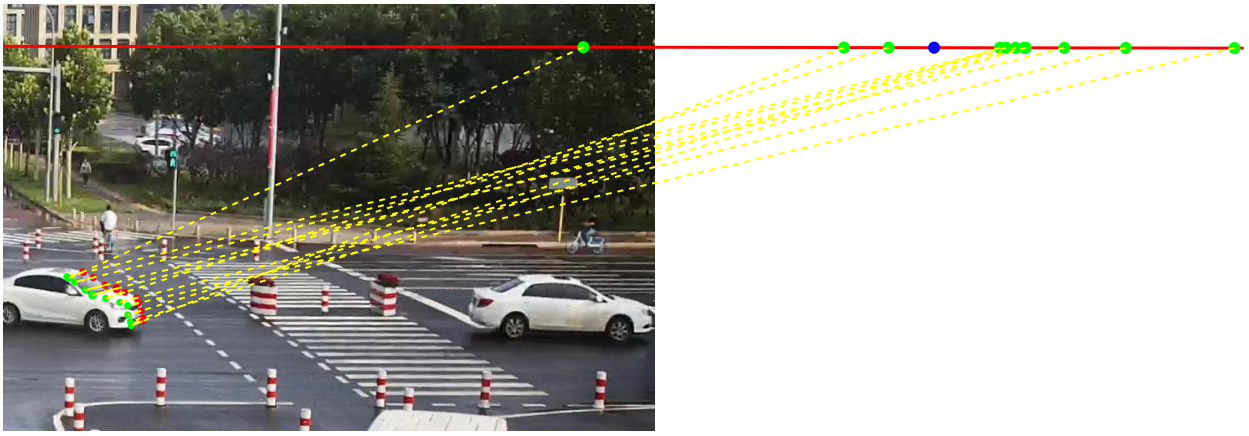


Figure 5.7: Each symmetry line $\tilde{\mathbf{l}}'_i$ (yellow) will, in general, intersect with the horizon line (red) at some point $\tilde{\mathbf{v}}'_{\mathrm{i}} = \tilde{\mathbf{l}}'_i \times \tilde{\mathbf{l}}_h$ (green) that is displaced from the true vanishing point $\tilde{\mathbf{v}}_{\mathrm{s}}$ due to annotation error. The point $\tilde{\mathbf{v}}^*_{\mathrm{s}}$ (blue), which minimizes the sum of squared distances to the symmetry lines $\left\{\tilde{\mathbf{l}}'_1, \ldots, \tilde{\mathbf{l}}'_{n_p}\right\}$, establishes the orientation of the symmetry plane of the vehicle. Symmetry points are in red on the driver's side and are in green on the passenger's side.

symmetry lines $\left\{\tilde{\mathbf{l}}'_1, \ldots, \tilde{\mathbf{l}}'_{n_p}\right\}$.

The normal $\mathbf{N}_s$ to the symmetry plane is then given by:

$$\mathbf{N}_s = (KR)^{-1}\tilde{\mathbf{v}}^*_{\mathrm{s}} \tag{5.8}$$

where $\tilde{\mathbf{v}}^*_{\mathrm{s}}$ is the homogeneous form of the vanishing point estimated from a set of symmetry lines $\left\{\tilde{\mathbf{l}}'_1, \ldots, \tilde{\mathbf{l}}'_{n_p}\right\}$. We can also calculate the unit normal to the symmetry plane:

$$\bar{\mathbf{N}}_s = \mathbf{N}_s / |\mathbf{N}_s| \tag{5.9}$$

5.4 Estimating the location of the symmetry plane

Having identified the orientation of the symmetry plane, we need to find its location in the normal direction $\mathbf{N}_s$. As discussed in chapter 4, annotated tire-ground contact points are points on a 2D image that correspond to the points in 3D where the outer edge of a tire touches the ground surface. Extremal symmetry point pairs are aligned vertically with the outer edges of a vehicle's tires. These points define two parallel planes of interest that we also need to find the locations for (Figure 5.8):

- Left extremal plane: the plane that contains the left extremal symmetry points (if present) and the left tire-ground contact points (if present).
- Right extremal plane: the plane that contains the right extremal symmetry points (if present) and the

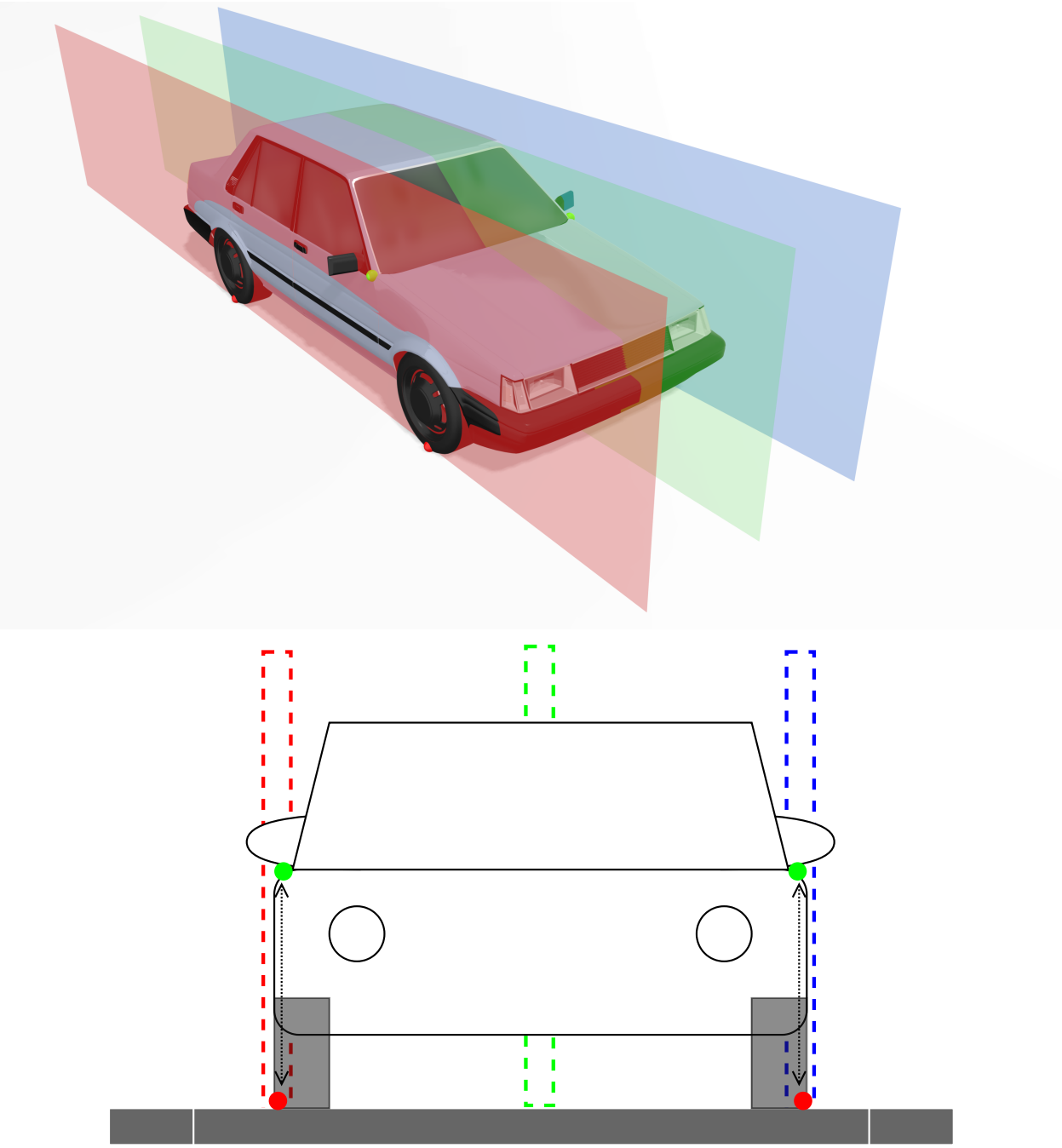


Figure 5.8: Left extremal plane, symmetry plane, right extremal plane are in red, green, blue respectively. Extremal symmetry points are in green and tire-ground contact points are in red. 3D model provided by remtromol (2021).

right tire-ground contact points (if present).

The distance between these two extremal planes is the overall width w of a vehicle. We explain the methods for estimating the plane locations under two scenarios in the following sections:

5.4.1 Scenario 1: At least one tire-ground contact point is annotated on both sides of the vehicle

As described in section 5.2, the terrain model provides us with a raster containing the height of each pixel in the ROI, for each annotated tire-ground contact point $\mathbf{t}_i = (x_i^{t\prime}, y_i^{t\prime})$, it is easy to look up the corresponding Y-axis coordinate (**World coordinate frame**'s vertical axis), or $Y_i^{t\prime}$, from this raster. Since there can be more than one tire-ground contact point annotated, we find the mean $Y^{t\prime}$ to be used for all annotated tire-ground contact points.

Having $Y^{t\prime}$, we then find the rectified 3D location $X_i^{t\prime}, Y^{t\prime}, Z_i^{t\prime}$ of an annotated tire-ground contact point $\mathbf{t}_i' = (x_i^{t\prime}, y_i^{t\prime})$ using the back-projection by solving for $\lambda_{t_i'}$, $X_i^{t\prime}$, $Z_i^{t\prime}$ and then repeat for all tire-ground contact points:

$$\begin{bmatrix} X_i^{t\prime} \\ Y^{t\prime} \\ Z_i^{t\prime} \end{bmatrix} = \lambda_{t_i'} (KR)^{-1} \begin{bmatrix} x_i^{t\prime} \\ y_i^{t\prime} \\ 1 \end{bmatrix} \tag{5.10}$$

where $X_i^{t\prime}$ and $Z_i^{t\prime}$ are the ground coordinates of the tire-ground contact point and $\lambda_{t_i'}$ is a scaling factor.

Having found the 3D locations of all the annotated tire-ground contact points, we can find the left extremal and the right extremal plane. Let $\mathbf{T}_l' = (X_l^{t\prime}, Y^{t\prime}, Z_l^{t\prime})$ be the mean of the 3D coordinates of the annotated tire-ground contact points on the left extremal plane. Because the normal to the symmetry plane is also normal to the left extremal plane, the left extremal plane can be represented in homogeneous coordinates as $\tilde{\mathbf{M}}_l = \left(a_s, b_s, c_s, \bar{d}_l\right)$ where the vector (a_s, b_s, c_s) is the unit normal to the symmetry plane $\bar{\mathbf{N}}_s$, and $\bar{d}_l$ is the distance from the origin to the plane.

To identify $\bar{d}_l$, we exploit the fact that $\bar{\mathbf{T}}_l'$ must lie on the left extremal plane, and thus:

$$\bar{\mathbf{T}}_l' \cdot \tilde{\mathbf{M}}_\mathbf{l} = \mathbf{0} \tag{5.11}$$

where $\bar{\mathbf{T}}_l'$ is the augmented form of $\mathbf{T}_l'$. Since we take the mean of the 3D locations of the annotated tire-ground contact points on the left extremal plane. If two left tire-ground contact points have been annotated, they will not generally lie exactly on the estimated left extremal plane $\tilde{\mathbf{M}}_l$. Therefore, we have to rectify these 3D locations to make them lie on the plane by finding their projections on the plane.

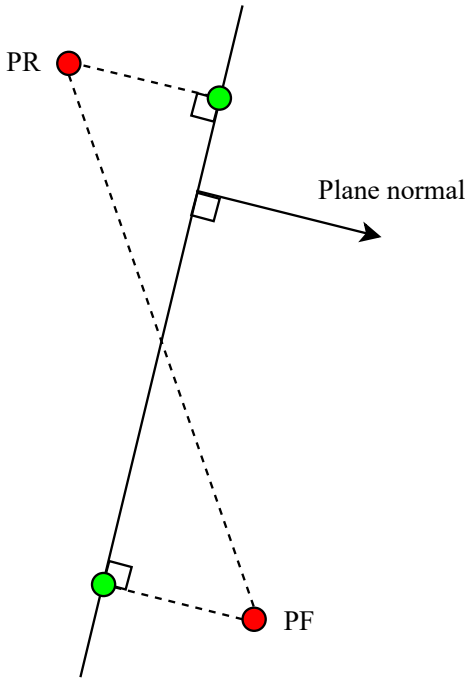


Figure 5.9: In this plan view, PR is the 3D location of the passenger rear tire-ground contact point and PF is the 3D location of the passenger front tire-ground contact point. We rectify the 3D locations of tire-ground contact points by finding their projections on the extremal plane. Original 3D point locations are in red and rectified 3D point locations are in green.

Let $\hat{\mathbf{T}}'_{il}$ be the 3D initially-estimated location of an annotated tire-ground contact point $\mathbf{t}'_{il}$ on the left extremal plane, and let $\mathbf{T}'_{il}$ be the rectified 3D location of the same point. We have two equations:

$$\mathbf{T}'_{il} = \hat{\mathbf{T}}'_{il} + \lambda \bar{\mathbf{N}}_s \tag{5.12}$$

$$\bar{\mathbf{T}}'_{il} \cdot \tilde{\mathbf{M}}_\mathbf{l} = 0 \tag{5.13}$$

substituting Equation 5.12 into Equation 5.13, it is straightforward to solve for λ. We now have a set of 3D points that are located on the left extremal plane. This process of rectification is shown in Figure 5.9.

Repeating this process for the right extremal plane, we can find the plane $\tilde{\mathbf{M}}_r = \left(a_s, b_s, c_s, \bar{d}_r\right)$ and the 3D locations of all tire-ground contact points located on it. We represent the symmetry plane, in homogeneous coordinates, as $\tilde{\mathbf{M}}_s = \left(a_s, b_s, c_s, \bar{d}_s\right)$ where the vector (a_s, b_s, c_s) is the unit normal to the symmetry plane $\bar{\mathbf{N}}_s$ and $\bar{d}_s$ can be found by:

$$\bar{d}_s = \frac{1}{2}(\bar{d}_l + \bar{d}_r) \tag{5.14}$$

because the left extremal plane and the right extremal plane are equally displaced from the symmetry plane.



Figure 5.10: This vehicle has tire-ground contact points (magenta) annotated only for the front and rear tires on the passenger's side. Symmetry points are in red on the driver's side and are in green on the passenger's side. An extremal symmetry point pair is shown with a bigger size.

We can also calculate the overall width w of a vehicle by:

$$w = |\bar{d}_l - \bar{d}_r| \tag{5.15}$$

5.4.2 Scenario 2: Tire-ground contact points on only one side of vehicle

An example of this scenario is shown in Figure 5.10. In this case, we need to use the annotations of **extremal** symmetry point pairs, which are the 3D symmetry points that are at the same distance from the symmetry plane as tire-ground contact points, lying on the extremal planes (see subsection 4.2.1).

Without loss of generality, we assume that one or more tire-ground contact points are annotated on the left extremal plane of the vehicle. We can identify this plane in homogeneous coordinates $\tilde{\mathbf{M}}_l = \left(a_l, b_l, c_l, \bar{d}_l\right)$ by using the same process as in subsection 5.4.1. Let the annotated **extremal** symmetry point pairs be $\mathbf{p}_i^{*\prime} = \{\mathbf{p}_{il}^{*\prime}, \mathbf{p}_{ir}^{*\prime}\} = \left\{\left(x_{il}^{p^{*\prime}}, y_{il}^{p^{*\prime}}\right), \left(x_{ir}^{p^{*\prime}}, y_{ir}^{p^{*\prime}}\right)\right\}$. We can then find the corresponding 3D locations $\mathbf{P}_{il}^{*}$ of point

$\mathbf{p}_{il}^{*\prime}$ using two equations:

$$\mathbf{P}_{il}^{*}(\lambda) = \lambda(KR)^{-1}\begin{bmatrix} x_{il}^{p^{*\prime}} \\ y_{il}^{p^{*\prime}} \\ 1 \end{bmatrix} \tag{5.16}$$

$$\bar{\mathbf{P}}_{il}^{*}(\lambda) \cdot \tilde{\mathbf{M}}_{\mathbf{l}} = 0 \tag{5.17}$$

where λ is a scaling parameter that can be solved for easily. We still need to find the 3D locations $\mathbf{P}_{ir}^{*}$ of the right extremal symmetry points $\mathbf{p}_{ir}^{*}$. These must lie on the right extremal plane $\tilde{\mathbf{M}}_r = \left(a_s, b_s, c_s, \bar{d}_r\right)$ parallel to $\tilde{\mathbf{M}}_l$. The line connecting $\mathbf{P}_{ir}^{*}$ and $\mathbf{P}_{il}^{*}$ must be orthogonal to the symmetry plane, hence:

$$\mathbf{P}_{ir}^{*} = \mathbf{P}_{il}^{*} + w_i \bar{\mathbf{N}}_s \tag{5.18}$$

where w_i is an unknown scalar. We also have another constraint: $\mathbf{P}_{ir}^{*}$ must lie on the optical ray through $\mathbf{p}_{ir}^{*}$. However, given the annotation error, the projection $\mathbf{p}_{ir}^{*}$ of each right extremal symmetry point will not align exactly with the annotated point $\mathbf{p}_{ir}^{*\prime}$. Thus, we find w_i that minimizes the 3D distance to the projection line through the annotated point $\hat{\mathbf{p}}_{ir}^{*\prime}$.

First, we represent the projection direction $\hat{\mathbf{D}}_{ir}^{*\prime}$ corresponding to $\hat{\mathbf{p}}_{ir}^{*\prime}$ in the **World coordinate frame** as:

$$\hat{\mathbf{D}}_{ir}^{*\prime} = (KR)^{-1}\begin{bmatrix} \hat{x}_{ir}^{p^{*\prime}} \\ \hat{y}_{ir}^{p^{*\prime}} \\ 1 \end{bmatrix} \tag{5.19}$$

Now, we have two lines: $\mathbf{l}_1$ is the line that goes through $\mathbf{P}_{il}^{*}$ in the direction of $\bar{\mathbf{N}}_s$, and $\mathbf{l}_2$ is the line that goes through the optical center of the camera in the projection direction $\hat{\mathbf{D}}_{ir}^{*\prime}$ corresponding to $\hat{\mathbf{p}}_{ir}^{*\prime}$. The optimal $\mathbf{P}_{ir}^{*}$, which is the point on $\mathbf{l}_1$ that is nearest to $\mathbf{l}_2$, is given by:

$$\mathbf{P}_{ir}^{*} = \mathbf{P}_{il}^{*} + \frac{-\mathbf{P}_{il}^{*} \cdot \hat{\mathbf{N}}_r^{*}}{\mathbf{N}_s \cdot \hat{\mathbf{N}}_r^{*}} \bar{\mathbf{N}}_s \tag{5.20}$$

(Weisstein, 2022) and thus the solution for w_i is given by:

$$w_i = \frac{-\mathbf{P}_{il}^{*} \cdot \hat{\mathbf{N}}_r^{*}}{\bar{\mathbf{N}}_s \cdot \hat{\mathbf{N}}_r^{*}} \tag{5.21}$$

where $\hat{\mathbf{N}}_r^{*} = \hat{\mathbf{D}}_{ir}^{*\prime} \times \left(\bar{\mathbf{N}}_s \times \hat{\mathbf{D}}_{ir}^{*\prime}\right)$.

Since there can be more than one **extremal** symmetry point pair, we can find the optimal w by taking

the average of all the estimated w_i:

$$w = \frac{1}{n}\sum_{i=1}^{n} w_i \tag{5.22}$$

where n is the number of **extremal** symmetry point pairs. Since $\bar{\mathbf{N}}_s$ is a vector of unit norm, this separation w between the two extremal planes is the overall width of a vehicle. Using $\bar{d}_l$ and optimal w, we can find $\bar{d}_r$. This completes the algebraic identification of $\tilde{\mathbf{M}}_r$. Having identified both $\tilde{\mathbf{M}}_r$ and $\tilde{\mathbf{M}}_l$, we can find the symmetry plane $\tilde{\mathbf{M}}_s$ using Equation 5.14 by taking the mean of signed distances $\bar{d}_l$ and $\bar{d}_r$.

5.5 Estimating the 3D locations of symmetry point pairs

We still have to estimate the 3D locations of non-extremal symmetry point pairs $\mathbf{p}_i = \{\mathbf{p}_{il}, \mathbf{p}_{ir}\} = \{(x^p_{il}, y^p_{il}), (x^p_{ir}, y^p_{ir})\}$ in the image.

In section 5.3, we estimate the common vanishing point of a vehicle $\mathbf{v}_\mathrm{s}$ from a set of vanishing lines. Given annotation error, annotated symmetry point pairs $\mathbf{p}'_i = \{\mathbf{p}'_{il}, \mathbf{p}'_{ir}\}$ will not generally lie exactly on a symmetry line. To rectify these points to symmetry lines, for each pair $\mathbf{p}_i$, we define the rectified vanishing line as the line that goes through the estimated vanishing point $\mathbf{v}_\mathrm{s}$ and minimizes the sum of squared distances to the annotated points $\{\mathbf{p}'_{il}, \mathbf{p}'_{ir}\}$.

Let $\bar{\mathbf{p}}'_i$ be the centroid of $\mathbf{p}'_{il}$ and $\mathbf{p}'_{ir}$. The rectified symmetry points $\mathbf{p_{il}}$ and $\mathbf{p_{ir}}$ are the projections of $\mathbf{p}'_{il}$ and $\mathbf{p}'_{ir}$ onto each rectified symmetry line that goes through the vanishing point $\mathbf{v}_\mathrm{s}$ and the centroid $\bar{\mathbf{p}}'_i$:

$$\mathbf{p_{il}} = \bar{\mathbf{p}}'_i + \frac{(\mathbf{p}'_{il} - \bar{\mathbf{p}}'_i)^\mathsf{T}(\mathbf{v}_s - \bar{\mathbf{p}}'_i)(\mathbf{v}_s - \bar{\mathbf{p}}'_i)}{|\mathbf{v}_s - \bar{\mathbf{p}}'_i|^2} \tag{5.23}$$

$$\mathbf{p_{ir}} = \bar{\mathbf{p}}'_i + \frac{(\mathbf{p}'_{ir} - \bar{\mathbf{p}}'_i)^\mathsf{T}(\mathbf{v}_s - \bar{\mathbf{p}}'_i)(\mathbf{v}_s - \bar{\mathbf{p}}'_i)}{|\mathbf{v}_s - \bar{\mathbf{p}}'_i|^2} \tag{5.24}$$

The 3D locations $\mathbf{P}_i = \{\mathbf{P}_{il}, \mathbf{P}_{ir}\} = \{(X^p_{il}, Y^p_{il}, Z^p_{il}), (X^p_{ir}, Y^p_{ir}, Z^p_{ir})\}$ of rectified symmetry points $\mathbf{p_{il}}$ and $\mathbf{p_{ir}}$ can be found using the back-projection of their augmented forms $\bar{\mathbf{p}}_{il}$ and $\bar{\mathbf{p}}_{ir}$:

$$\mathbf{P}_{il} = \lambda_l (KR)^{-1}\bar{\mathbf{p}}_{il} \tag{5.25}$$

$$\mathbf{P}_{ir} = \lambda_r (KR)^{-1}\bar{\mathbf{p}}_{ir} \tag{5.26}$$

where λ_l and λ_r are scaling parameters that we can find using two constraints:

1. The line $\mathbf{P}_{il}\mathbf{P}_{ir}$ must be normal to the symmetry plane.

2. The mean of $\mathbf{P}_{il}$ and $\mathbf{P}_{ir}$ must lie on the symmetry plane.

Here we adopt the steps used by Li et al. (2012). First, from condition 1, we have:

$$\mathbf{P}_{il} - \mathbf{P}_{ir} = m\bar{\mathbf{N}}_s \tag{5.27}$$

where m is an unknown non-zero scalar. Substituting Equation 5.25 and Equation 5.26 into Equation 5.27, we have:

$$\lambda_l(KR)^{-1}\bar{\mathbf{p}}_{il} - \lambda_r(KR)^{-1}\bar{\mathbf{p}}_{ir} = m\bar{\mathbf{N}}_s \tag{5.28}$$

Multiplying by KR on both sides, we now have:

$$\lambda_l\bar{\mathbf{p}}_{il} - \lambda_r\bar{\mathbf{p}}_{ir} = m\lambda_s\bar{\mathbf{v}}_s \tag{5.29}$$

where λ_s is a non-zero scaling parameter.

Since $\bar{\mathbf{p}}_{il} = (\mathbf{p}_{il}, 1)$, $\bar{\mathbf{p}}_{ir} = (\mathbf{p}_{ir}, 1)$, and $\bar{\mathbf{v}}_s = (\mathbf{v}_s, 1)$, Equation 5.29 can be simplified to:

$$\lambda_l(\mathbf{p}_{il} - \mathbf{v}_s) = \lambda_r(\mathbf{p}_{ir} - \mathbf{v}_s) \tag{5.30}$$

From condition 2, we have:

$$\frac{\mathbf{P}_{il} + \mathbf{P}_{ir}}{2} \cdot \bar{\mathbf{N}}_s + \bar{d}_s = 0 \tag{5.31}$$

where $\bar{d}_s$ is the distance to the symmetry plane $\mathbf{M}_s = (\bar{\mathbf{N}}_s, \bar{d}_s)$.

Substituting Equation 5.25, Equation 5.26 into Equation 5.31 and along with Equation 5.30, we have two linear equations with two unknowns λ_l and λ_r. It is straightforward to solve for the two unknowns λ_l and λ_r, which identify $\mathbf{P}_{il}$ and $\mathbf{P}_{ir}$ respectively. This forms an algebraic estimate of the 3D point cloud corresponding to all annotated 2D points. In the next chapter, chapter 6, we will explore least-squares refinement of this estimate.

Chapter 6

Refining the algebraic estimates

In chapter 5, we have derived algebraic estimates of the symmetry plane and the 3D locations of all annotated points. To improve the accuracy of these estimates, we can refine them iteratively by minimizing the reprojection errors $\mathbf{E}$ of all n_p annotated symmetry and tire-ground contact points:

$$\mathbf{E} = \sum_{i=1}^{n_p} (\mathbf{p}'_i - \hat{\mathbf{p}}_i)^2 \tag{6.1}$$

where $\mathbf{p}'_i = (x'_i, y'_i)$ is an annotated point, and $\hat{\mathbf{p}}_i = (\hat{x}_i, \hat{y}_i)$ is the reprojection of the 3D point $\mathbf{P}_i$ that satisfies the constraints of our model:

1. Tire-ground contact points must remain on the ground.
2. Left and right extremal planes must be parallel to the symmetry plane and equally displaced from the symmetry plane.
3. Left tire-ground contact points and left extremal symmetry points must lie in the left extremal plane.
4. Right tire-ground contact points and right extremal symmetry points must lie in the right extremal plane.
5. Symmetry point pairs must be equally displaced from the symmetry plane.
6. Symmetry lines connecting symmetry point pairs must be orthogonal to the symmetry plane.

For each vehicle, we employ an **Vehicle-centered coordinate frame** (Figure 6.1) with the origin located at the estimated 3D location of one of the annotated tire-ground contact points (Driver front, Driver

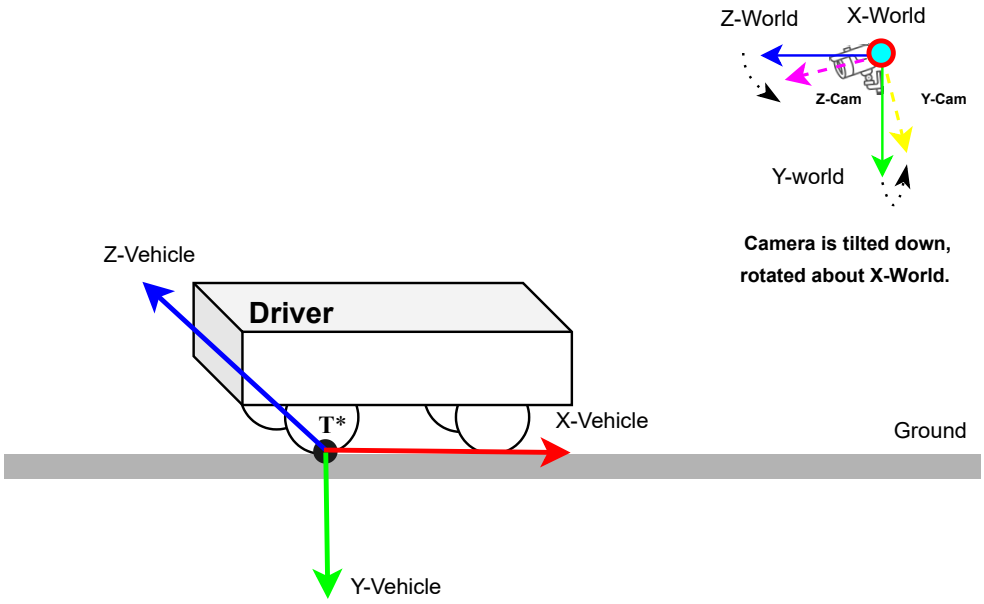


Figure 6.1: The **Vehicle-centered coordinate frame** with the origin $\mathbf{T}^*$ (black circle) located at one of the tire-ground contact points (Driver front in this example). The Z-axis of the vehicle's coordinate frame is constrained to point from the **driver's side** to the **passenger's side** and align with the normal to the symmetry plane $\mathbf{N}_s$.

rear, Passenger front, or Passenger rear), which we label $\mathbf{T}^*$. Since we always have at least one annotated tire-ground contact point, we will always have an origin for this vehicle-centered coordinate frame.

For this coordinate frame, the Y-axis is pointing down, consistent with the **World coordinate frame**. Regardless of where the coordinate frame's origin is located, the Z-axis is constrained to point from the **driver's side** to the **passenger's side** and align with the normal to the symmetry plane $\mathbf{N}_s$. The X-axis points along the vehicle's body with the positive direction pointing towards the vehicle's rear (Figure 6.1). This removes the ambiguity of X-Z axes and determines the transformation between the **Vehicle-centered coordinate frame** and the **World coordinate frame**.

In this vehicle-centered frame, we optimize:

DOF	Variable
$n_t - 1$	The X-value of all n_t tire-ground contact points except for the tire chosen as the origin of the vehicle coordinate frame
1	The orientation of the symmetry plane
1	The location of the symmetry plane $\bar{d}_s$
1	The vehicle's overall width w given by the distance between the two extremal planes
$3n_p$	The 3D coordinates of n_p non-extremal symmetry point pairs
$2n_e$	The Y-value and X-value of n_e extremal symmetry point pairs

Additionally, we must optimize the X-value and the Z-value of $\mathbf{T}^*$, the ground coordinates of the vehicle. Using the Levenberg–Marquardt algorithm (Marquardt, 1963), we can optimize the above degrees of freedom to minimize the sum of squared reprojection errors $\mathbf{E}$.

6.1 Incorporating statistics from the Canadian Vehicle Specifications (CVS) database to improve prediction quality

The Canadian Vehicle Specifications (CVS) database (Canada, 2023) has been critical in improving the performance of our method. We acknowledge that this dataset is compiled by a Canadian authority, while the DAIR-V2X (Yu et al., 2022) dataset includes data from intersections in China. However, upon inspection of detected makes and models (see section 7.2), many make/models of vehicles from DAIR-V2X (Yu et al., 2022) are included in the Canadian dataset. Additionally, the availability of open-source, reliable datasets that include a large number of vehicle makes and models is very limited. We have used it in the following ways:

6.1.1 Producing a complete bounding box from a set of 3D predicted symmetry point pairs and tire-ground contact points

After the nonlinear optimization process, we obtain a set of 3D predicted points and critical measurements such as width and wheelbase (if available). Since we may not always have points on the vehicle's top, front, and rear surfaces, the length and height of the 3D vehicle bounding box might be underconstrained by the extracted 3D point cloud. (Note that this issue also occurs with sparse LiDAR-based point clouds.) We can more accurately predict the remaining unknown dimensions using the overall width and wheelbase with a Multivariate Gaussian model estimated from the Canadian Vehicle Specifications (CVS) database (Canada,

Table 6.1: Descriptive statistics for key dimensions in the Canadian Vehicle Specifications database (Canada, 2023) (in cm). **OH:** Overall height. **OW:** Overall width. **WB:** Wheelbase. **RH:** Rear overhang. **FH:** Front overhang.

Dimension	Mean	Minimum	Maximum	Standard Deviation
OH	168.34	110.0	305.0	32.64
OW	191.67	156.0	293.0	12.29
WB	305.63	187.0	447.0	49.3
RH	107.25	39.0	202.0	18.93
FH	93.18	42.0	110.0	8.06

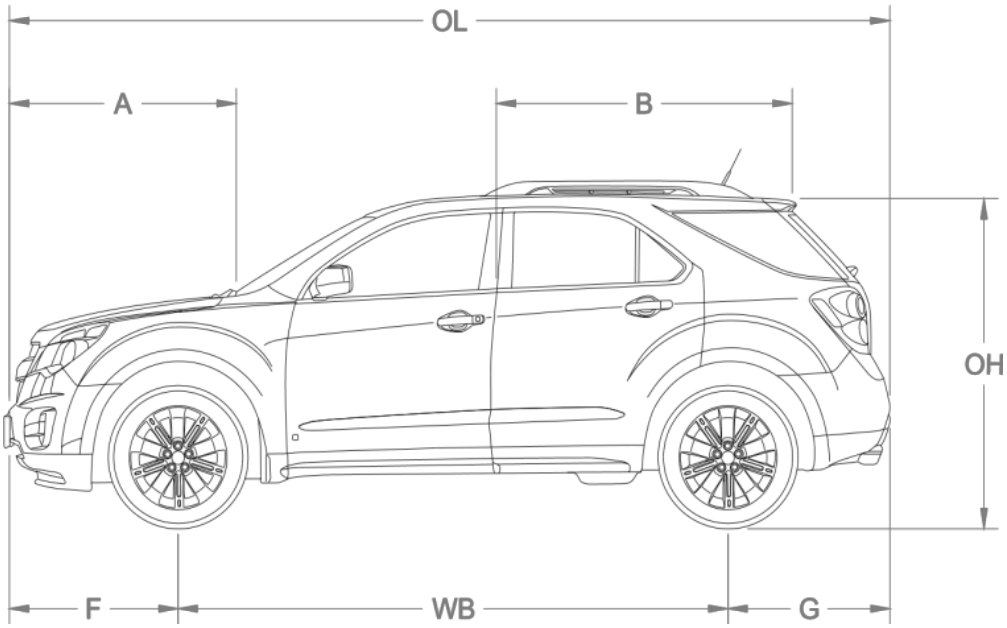
Table 6.2: Pearson correlation coefficients for key dimensions. **OH:** Overall height. **OW:** Overall width. **WB:** Wheelbase. **RH:** Rear overhang. **FH:** Front overhang.

	OH	**OW**	**WB**	**RH**	**FH**
OH	1.00	0.59	0.76	0.61	0.43
OW	0.59	1.00	0.74	0.60	0.56
WB	0.76	0.74	1.00	0.70	0.49
RH	0.61	0.60	0.70	1.00	0.54
FH	0.43	0.56	0.49	0.54	1.00

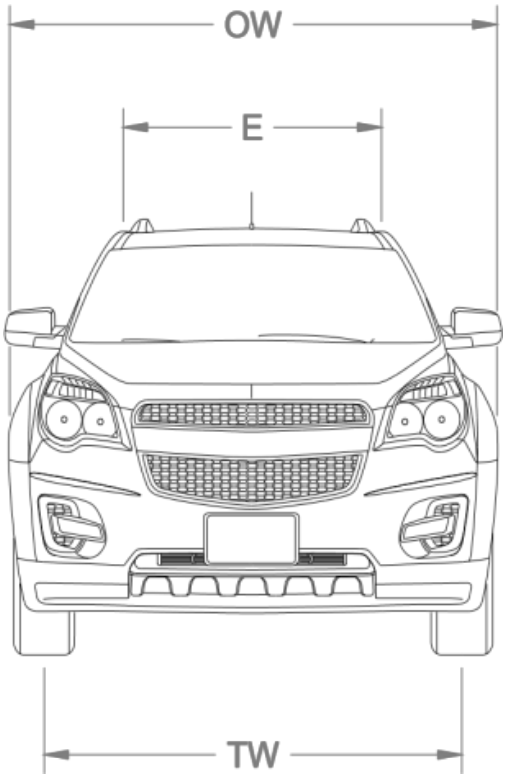
2023) (Figure 6.2). The database includes passenger vehicles sold in Canada between the years 2011-2023. It specifies the major dimensions of 11,707 vehicles, including passenger cars, vans, SUVs, hatchbacks, station wagons, and full-size pickup trucks, excluding buses, motorcycles, and long-haul trucks. Descriptive statistics for critical dimensions are provided (Table 6.1).

We model these dimensions as joint Gaussian with a full covariance matrix. The high correlations between these dimensions (Table 6.2) testify to the potential of using the database to complete missing dimensions for incomplete point clouds. Please note that the overall width, or OW, from the database, does not include side-view mirrors. This aligns with our predicted overall width, which measures the distance between the left and right extremal planes (Figure 6.3).

We initially use the 3D point cloud to estimate only the position, orientation, and overall width and wheelbase (if available) of the vehicle, and estimate the remaining parameters (overall height, rear overhang, and front overhang) using the optimal mean-square estimate from the corresponding conditional Gaussian model (Bishop, 2006). If any 3D points fall outside one or more dimensions of this initial bounding box estimate, the bounding box is extended to include them. See Figure 6.4 for a comparison of a predicted 3D bounding box before and after the incorporation of the Multivariate Gaussian model for dimensions.



(a) Overall Length (OL), Front End Length (A), Rear End Length (B), Overall Height (OH), Front Overhang (F), Wheelbase (WB), Rear Overhang (G)



(b) Overall Width (OW), Roof Width (E), Track Width (TW)

Figure 6.2: Dimensions provided in the Canadian Vehicle Specifications (CVS) database (Canada, 2023)



Figure 6.3: Overall width, or OW, from the Canadian Vehicle Specifications (CVS) database (Canada, 2023) does not contain side-view mirrors. Adapted from Canada (2023).

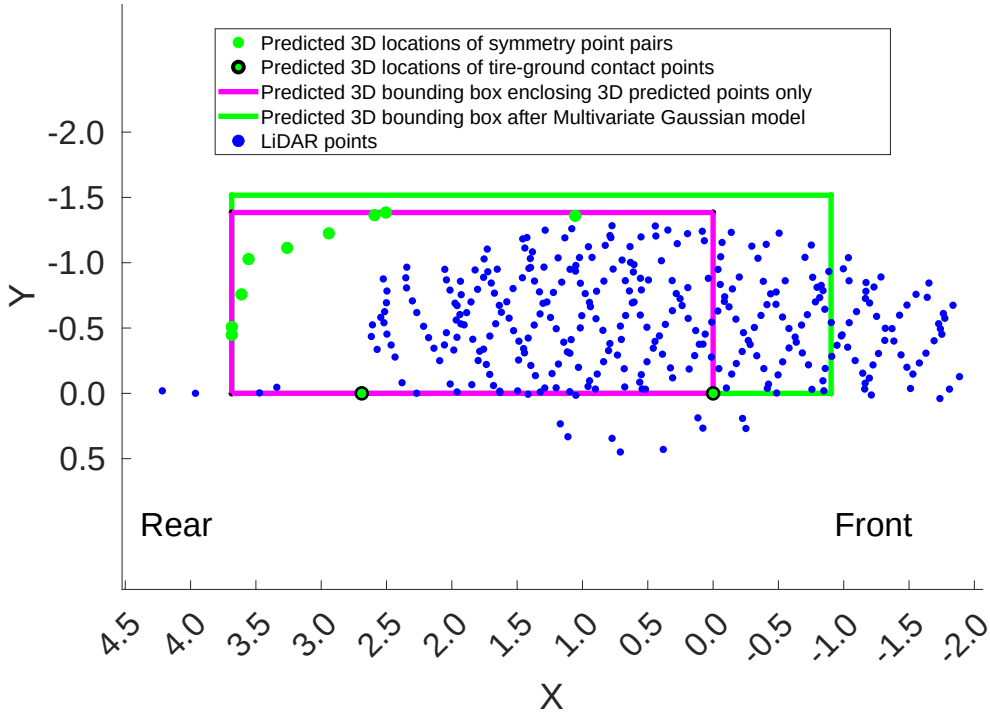


Figure 6.4: Side view of a vehicle: Comparison of a predicted 3D bounding box before and after incorporating the Multivariate Gaussian model of dimensions.

Table 6.3: Maximum constraints adopted using estimated statistics from the Canadian Vehicle Specifications (CVS) database (Canada, 2023).

Dimension	**Maximum (m)**
Width including side-view mirrors	3.53
Height	3.05
Front Overhang	1.10
Wheelbase	4.47
Rear Overhang	2.02

6.1.2 Using statistics from the Canadian Database as constraints during the non-linear optimization process

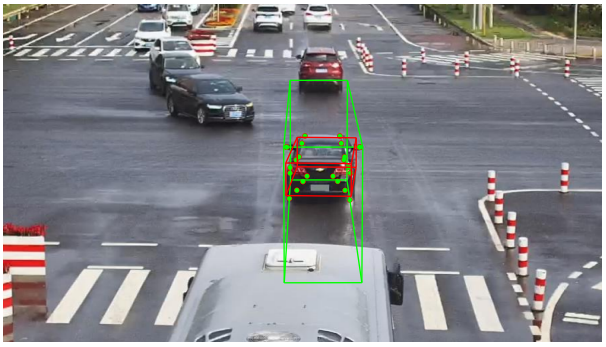
Upon inspecting some initial predictions, we observed that our predicted boxes exhibit several failure cases. One or more pairs of symmetry points are estimated to be too far from the center of the vehicle itself, causing the predicted 3D bounding boxes to be excessively long, which is not realistic for the vehicles in this dataset. Similarly, if estimated too high from the ground, the predicted 3D bounding boxes become unrealistically tall.

These errors occur most frequently when the symmetry plane is nearly parallel to the line of sight, and the estimation of the scaling parameter for symmetry point pairs becomes ill-conditioned (Figure 6.5, Figure 6.6).

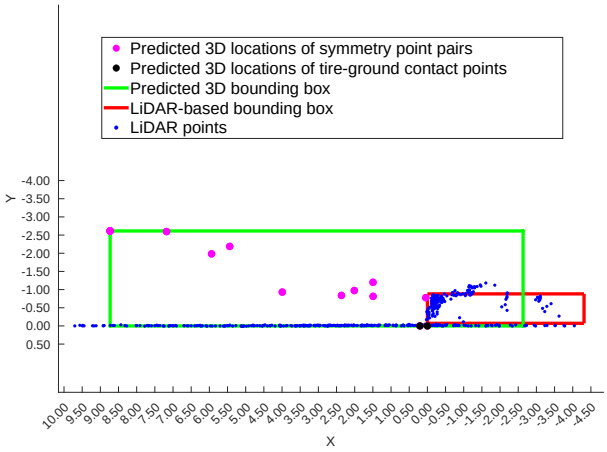
To reduce this sensitivity to the annotations, we constrain the non-linear optimization with bounds estimated from the Canadian Vehicle Specifications (CVS) database (Canada, 2023). These maximums are obtained directly from the descriptive statistics of the dataset (Table 6.1). While symmetry point pairs can consist of side-view mirrors, the overall width (OW) from the database does not include side-view mirrors. Therefore, we added a maximum of 0.6m for a pair of side-view mirrors to the maximum OW of 2.93m, producing the maximum width constraint of 3.53m (Table 6.3). These constraints (Table 6.3) steer the optimization process to produce more realistic vehicle dimensions (Figure 6.5, Figure 6.6). We implement this using **MATLAB's lsqnonlin**.

6.2 Resolving systemic bias in annotations

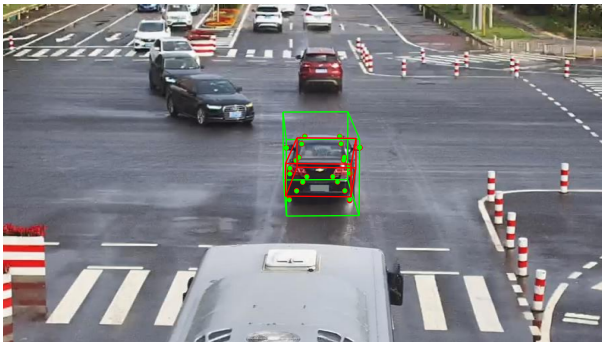
Upon initial evaluation of our predictions, we discovered a systemic underestimation of the Overall Width (OW) when compared to the ground truth make and model OW. We detected the ground truth make and model using Google Reverse Search (see section 7.2). To further investigate this systemic bias, we divide the vehicles in the dataset into four categories:



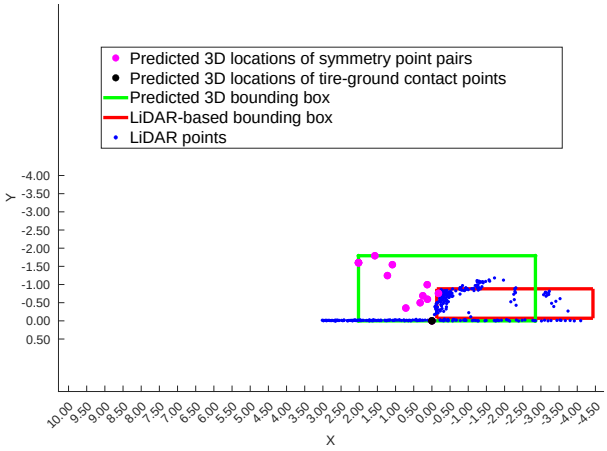
(a) Before - Projection of the predicted bounding box on the 2D image. Annotated points (red) and reprojected points (green).



(b) Before - Side view of the predicted bounding box.



(c) After - Projection of predicted bounding box on the 2D image. Annotated points (red) and reprojected points (green).

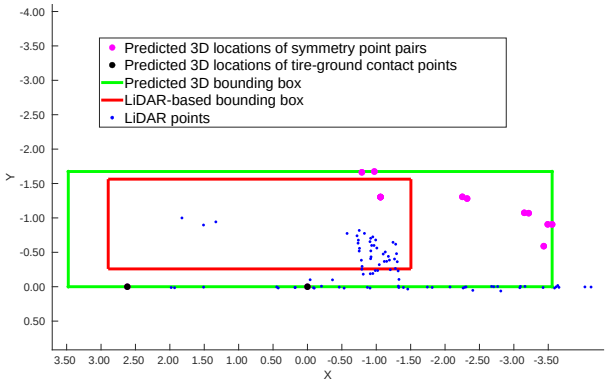


(d) After - Side view of the predicted bounding box.

Figure 6.5: An example of enhanced prediction following the application of constraints to the predicted dimensions: Prior to the enforcement of these constraints, the predicted 3D bounding box was excessively long due to the unconstrained Rear Overhang (RH).



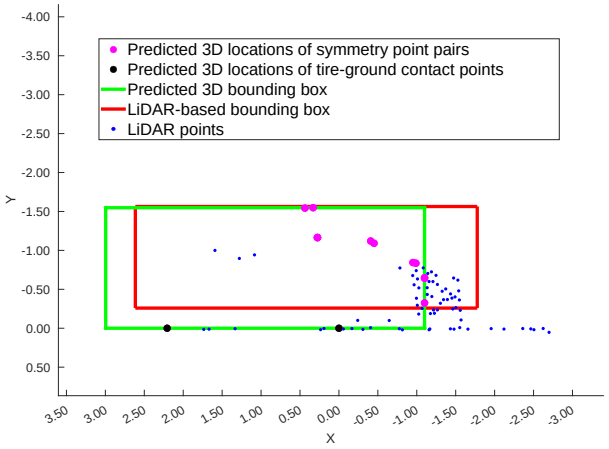
(a) Before - Projection of the predicted bounding box on the 2D image. Annotated points (red) and reprojected points (green).



(b) Before - Side view of the predicted bounding box.



(c) After - Projection of predicted bounding box on the 2D image. Annotated points (red) and reprojected points (green).



(d) After - Side view of the predicted bounding box.

Figure 6.6: An example of enhanced prediction following the application of constraints to the predicted dimensions: Prior to the enforcement of these constraints, the predicted 3D bounding box was excessively long due to the unconstrained Front Overhang (FH).

- **Category 1:** Vehicles with tire-ground contact points annotated on both sides.
- **Category 2:** Vehicles with tire-ground contact points annotated on only one side and side-view mirror points annotated. Some of these vehicles have front and rear tire-ground contact points annotated on the same side of the vehicles, allowing for wheelbase measurement.
- **Category 3:** Vehicles with two tire-ground contact points annotated on one side, with no side-view mirrors annotated.
- **Category 4:** Vehicles with one tire-ground contact point annotated on one side, with no side-view mirrors annotated.

The biases relative to the ground truth make and model OW across these four categories are shown in Table 6.4. The biases are most severe in Categories 2, 3, and 4. This might in some cases be due to mislocalization of the extremal symmetry points on the far side of a vehicle. Examples of Category 3 are shown in Figure 6.7.

Table 6.4: Comparison of predicted OW against ground truth make/model OW across four categories of annotation (see text).

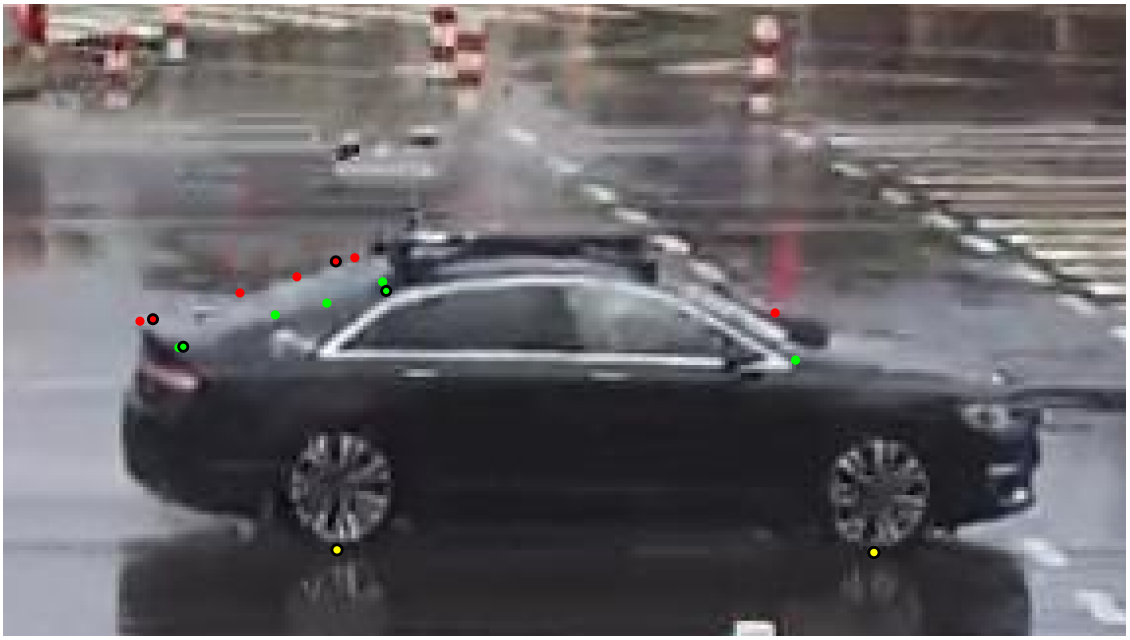
	Category 1		**Category 2**		**Category 3**		**Category 4**	
	Before	**After**	**Before**	**After**	**Before**	**After**	**Before**	**After**
Bias (m)	-0.17	-0.05	-0.28	0.01	-0.31	0.05	-0.42	-0.16
MAE (m)	0.18	0.15	0.29	0.07	0.34	0.06	0.42	0.16
Number of vehicles	396		57		243		1	

Upon investigation, we found that the annotations of the tire-ground contact points and mirror symmetry points are more reliable than those of the extremal symmetry point pairs. Therefore, we adopt the strategies below to find OW', which is defined as the rectified OW:

- **Category 1:** OW is kept without rectification for these vehicles, except for vehicles with side-view mirror pairs annotated. We derive a mean mirror offset of 0.25m for two mirrors, estimated from 1022 vehicles with side-view mirrors and tire-ground contact points on both sides. OW' is set to the distance between mirror pairs minus the offset of 0.25m for two mirrors.
- **Category 2:** OW' equals the distance between mirror pairs minus 0.125m, except for vehicles with wheelbase annotated. For these vehicles, OW' is estimated from the joint Gaussian model using the Wheelbase (see subsection 6.1.1). We find under these circumstances that the mirror annotations are less reliable, presumably because the vehicle is more side-facing, and so opt to instead use the 3D distance between the front and back tire-ground contact points as an estimate of the wheelbase.



(a) Ground truth make and model $OW = 1.82$m, LiDAR-derived $OW = 1.84$m, symmetry-based $OW = 1.73$m and symmetry-based $OW' = 1.85$m after the rectification (see text).



(b) Ground truth make and model $OW = 1.86$m, LiDAR-derived $OW = 2.00$m, symmetry-based $OW = 1.28$m and symmetry-based $OW' = 1.89$m after the rectification (see text).

Figure 6.7: Cases in Category 3 where the annotation on the far side of the car causes the underestimation of overall width OW.

- **Category 3:** OW' is estimated from the joint Gaussian model using the Wheelbase (see subsection 6.1.1).
- **Category 4:** OW' is calculated using the annotated symmetry points but then adjusted. The adjustment required is learned in an unsupervised fashion from examples in categories 1-3 by calculating a mean adjustment (0.25m) across all vehicles. We found that the annotated far point of extremal symmetry pairs tended to be biased inward, perhaps due to the self-occlusion of the point by the vehicle surface.
- OW' is used when the $OW' > OW$ due to the observed underestimation.

Upon applying the above adjustments, the biases are improved across all four categories (see Table 6.4).

Chapter 7

Evaluation

First, we categorize the annotated vehicles into three groups based on the distance from the center of 3D LiDAR-based bounding boxes to the camera:

1. Near-field: ≤ 30 meters
2. Mid-field: > 30 meters and ≤ 50 meters
3. Far-field: > 50 meters and ≤ 150 meters

As discussed in section 5.2, we only evaluate the vehicles that have annotated tire-ground contact points within the region of interest where we have a reliable terrain model (81% of all annotated vehicles).

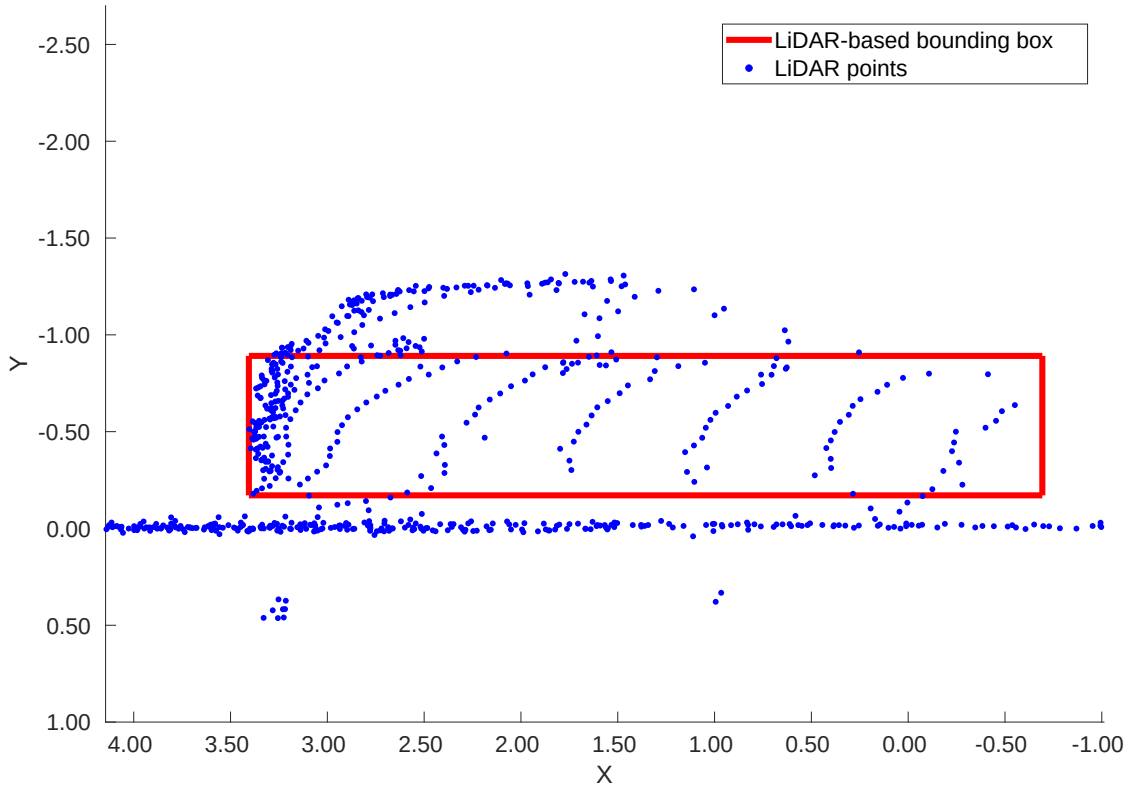
The LiDAR-based bounding boxes provided in the DAIR-V2X (Yu et al., 2022) dataset have several issues:

1. Errors in time synchronization between the LiDAR sensor and the camera.
2. Calibration errors between the LiDAR sensor and the camera.
3. 3D LiDAR-based bounding boxes do not cover the entire vehicle body, from the highest points down to the tire-ground contact points (Figure 7.1).

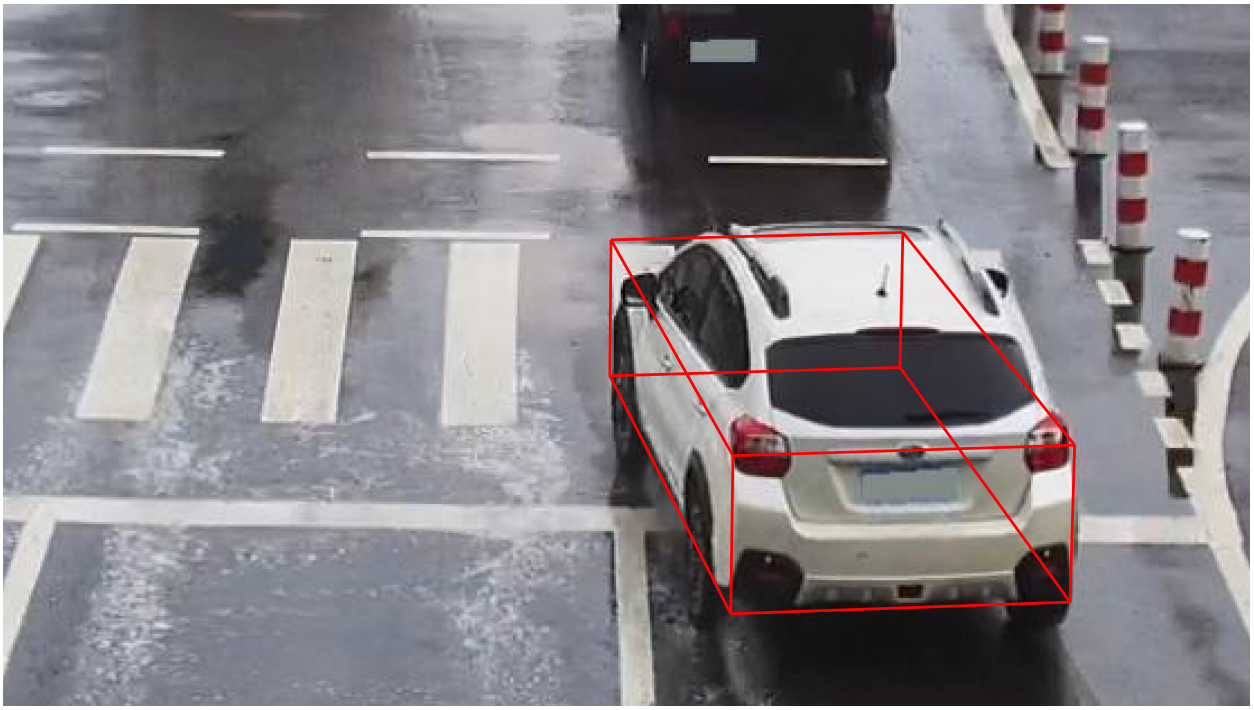
Partly due to these reasons, we evaluate our 3D point clouds using several methods outlined below.

7.1 Ground-truthing for 3D object detection

Since the estimated 3D point cloud typically covers only some parts of a vehicle, to have more accurate vehicle dimensions, we made use of the Canadian Vehicle Specifications (CVS) database (Canada, 2023) (see section 6.1). This database contains the vehicle dimensions of vehicles dated back to 2011.



(a) The side view of a vehicle showing the 3D LiDAR-based bounding box does not cover the entire vehicle body.



(b) Visualization of the same 3D bounding box on the vehicle's image.

Figure 7.1: The 3D LiDAR-based bounding box does not cover the entire vehicle body as tire-ground contact points are excluded.

Using a multivariate Gaussian model estimated from the database, we use the estimated 3D point cloud to estimate the seven parameters of a vehicle's 3D bounding box and compare them against the ground truth. These parameters are the location of the 3D bounding box in the X, Y, and Z directions, length, width, height, and heading angle.

7.1.1 3D object detection metrics

Accuracy can be summarized by the common metric of intersection-over-union, or **IoU**, which can be used to compare our predicted 3D bounding boxes against the LiDAR-based bounding boxes. We will consider both the 3D **IoU** and the 2D **IoU** of the bird's-eye view projections of these 3D bounding boxes, $\mathbf{IoU}_{BEV}$.

Denoting $\mathbf{B}_p$ and $\mathbf{B}_g$ as the 2D bird's-eye view projections of the predicted and ground truth LiDAR-based bounding box respectively, $\mathbf{IoU}_{BEV}$ is defined as:

$$\mathbf{IoU}_{BEV} = \frac{\text{Area of overlap of } \mathbf{B}_p \text{ and } \mathbf{B}_g}{\text{Area of union of } \mathbf{B}_p \text{ and } \mathbf{B}_g}$$

To calculate this BEV intersection, we use the method of Zhou et al. (2019). As the vertical axes of two bounding boxes must be parallel to each other, we can calculate the height overlap easily, and then the **IoU** can be calculated as:

$$\mathbf{IoU} = \frac{\text{Area of BEV overlap} \times \text{Height overlap}}{\text{Volume of } \mathbf{B}_p + \text{ Volume of } \mathbf{B}_g - \text{Area of BEV overlap } \times \text{Height overlap}}$$

We observe that most LiDAR-based bounding boxes in the DAIR-V2X dataset (Yu et al., 2022) do not appear to extend fully to the ground (Figure 7.1). This issue is expected to lower the 3D **IoU**.

Dimensions of 3D bounding boxes

To understand the performance in more detail, we compare the predicted 3D dimensions, w, l, h, against the LiDAR-derived dimensions: width, length, and height of vehicles.

Ground distance to 3D bounding boxes

We calculate the horizontal distance from the camera center to the center of the base of predicted 3D bounding boxes and 3D LiDAR-based bounding boxes and report the Mean Absolute Error and the Bias.

Orientation of vehicles

To evaluate the accuracy of our method for the heading angle of a vehicle, we can compare the orientation of the symmetry plane and report the Mean Absolute Error.

7.1.2 Results and analysis

Table 7.1: Deviations between 3D bounding boxes estimated from symmetry and LiDAR data

Metric	Value
Number of vehicles	2078
Mean reprojection error (pixels)	2.20
IoU$_{BEV}$	0.50
IoU	0.35
MAE of distance to center of LiDAR-derived bounding box (m)	1.18
Bias of distance to center of LiDAR-derived bounding box (m)	-0.68
Mean heading angle error (degrees)	3.55
MAE Height (m)	0.51
MAE Length (m)	0.66
MAE Width (m)	0.27

Table 7.1 shows the deviation between our symmetry-based 3D bounding box estimates and estimates derived from LiDAR. We caution against interpreting these deviations as errors in the symmetry-based method, as there is evidence that the LiDAR-based ground truth is itself not without error (see subsection 7.1.1). However, it is certainly useful to assess consistency between the two methods.

IoU: Across all the vehicles from six intersections, we achieved an **IoU**$_{BEV}$ of 0.50, and an **IoU** of 0.35, which are above the common **IoU** threshold of 0.25 for 3D object detection.

Heading Angle Error: The mean heading angle error is 3.55 degrees, which proves that by using 2D annotations of symmetry point pairs, we can accurately estimate the orientation of a vehicle. This proves that the bilateral symmetry constraint is a strong constraint that should be leveraged to estimate the orientation of a vehicle with high accuracy.

MAE with LiDAR-Derived Bounding Box Dimensions: Length has the highest MAE. This is because for vehicles heading towards or away from the camera we do not have points annotated at the rear or front extremities of a vehicle.

Height also has a significant MAE. This is expected because LiDAR-derived bounding boxes often do not cover the entire vehicle body (see subsection 7.1.1). Width has the lowest MAE because LiDAR estimates are fairly accurate and symmetry-based estimates are fairly accurate as well, and corrected as in section 6.2.

Table 7.2: Deviations between 3D bounding boxes estimated from symmetry and LiDAR data as a function of distance: Near-field ($\leq$ 30m), Mid-field (30 m - 50 m), Far-field (50m - 150m)

Metric	**Near-Field**	**Mid-Field**	**Far-Field**
Number of vehicles	179	734	1165
Mean reprojection error (pixels)	5.15	2.60	1.50
$\mathbf{IoU}_{BEV}$	0.50	0.56	0.46
IoU	0.24	0.37	0.35
MAE of distance to center of LiDAR-derived bbox (m)	1.23	0.95	1.32
Bias of distance to center of LiDAR-derived bbox (m)	-1.05	-0.51	-0.73
Mean heading angle error (degrees)	2.60	2.83	4.15
MAE Height (m)	0.95	0.58	0.39
MAE Length (m)	1.05	0.68	0.58
MAE Width (m)	0.27	0.26	0.27

7.1.3 Results and analysis based on difficulty levels

As explained earlier, we divide the vehicles into three categories, Near-field ($\leq$ 30m), Mid-field (30 m - 50 m), Far-field (50m - 150m), based on the horizontal distance from the camera to the center of the base of the LiDAR-derived 3D bounding box. Table 7.2 shows deviations between symmetry and LiDAR-derived 3D vehicle estimates in each category. We find little systematic change in deviations as distance to the vehicle increases, with some exceptions noted below. The first exception is that the deviation in mean heading angle error does progressively increase with increasing distance. This could be due to increasing errors in both symmetry-derived and LiDAR-derived ground truth. The second exception is that the lowest **IoU** occurs at the Near-field level. Additionally, the distance bias, height, and length estimation *improves* with increasing distance, which initially seems counterintuitive. This may be due in part to LiDAR errors in the near field (Figure 7.2). LiDAR-derived 3D bounding boxes do not always cover the entire vehicle height. This results in a much higher MAE Height of 0.95 meters in the Near-field category, which reflects the difference between the complete height coverage of our predicted 3D bounding boxes and the partial height coverage by the LiDAR-derived 3D bounding boxes (Figure 7.2).

7.2 Evaluating against real ground truth

While it is reassuring to see a good agreement between our symmetry-based ground truth and LiDAR-derived ground truth, this analysis does not provide a very accurate estimate of the errors produced by our method. One limitation of the evaluation methods discussed above is that LiDAR-based bounding boxes may themselves be inaccurate due to a variety of reasons such as calibration errors. In general, they are rarely verified against a more reliable source of ground truth. To address this limitation, by identifying a vehicle's make and model, we can evaluate our estimated length, width, and height against the true values from a



Figure 7.2: An example of a 3D LiDAR-based bounding box that does not cover the entire vehicle body. Our predicted 3D bounding boxes cover the entire vehicle's bodies while LiDAR-derived bounding boxes from the DAIR-V2X (Yu et al., 2022) dataset do not, causing **IoU** to be lower.

vehicle dimensions database such as the Canadian Vehicle Specifications (CVS) database (Canada, 2023). This is, to our knowledge, the first time validation against absolute ground truth for vehicle dimensions has been attempted.

The process of obtaining ground truth make and model was as follows:

1. For each track of vehicles, we use the images contained within the track to reverse search for make and model on Google Image (Figure 7.3). Only the tracks with a consistency of make and model detected using multiple frames within the track are retained. We exclude tracks for which Google Image reverse search returns with ambiguity. This happens when no clear make and model are detected (Figure 7.4).

2. For the identified make and model, we obtain the ground truth dimensions using the Canadian Vehicle Specifications (CVS) database (Canada, 2023).

The predicted and LiDAR-derived bounding box dimensions are evaluated against the ground truth make and model dimensions. Metrics we used to compare them:

Bias of dimensions:

$$\text{Bias (m)} = \frac{1}{N}\sum_{i=1}^{N}(P_i - G_i) \tag{7.1}$$

Mean Absolute Error of dimensions:

$$\text{MAE (m)} = \frac{1}{N}\sum_{i=1}^{N}(|P_i - G_i|) \tag{7.2}$$

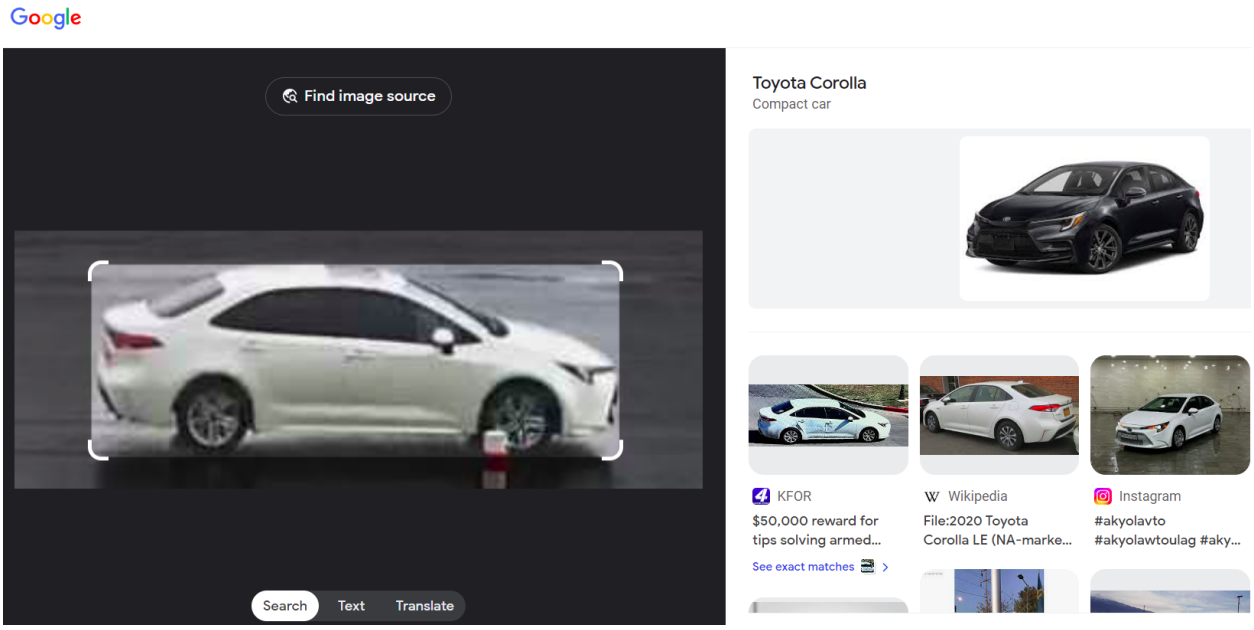
Mean percent Absolute Error:

$$\text{MAE (\%)} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{|P_i - G_i|}{G_i}\right) \times 100 \tag{7.3}$$

where P_i is the predicted dimension and G_i is the ground truth dimension and N is the number of vehicles.

Evaluation against LiDAR-derived bounding box without inter-frame propagation

While our method only uses the information from a single frame to make our predictions, LiDAR-derived bounding boxes are usually propagated across frames (see subsection 2.1.1). It is not fair to compare our predictions with the propagated LiDAR-derived bounding boxes, as they carry information from frames when the vehicles are closer to the sensors. We could also take the same approach, but that is beyond the scope of this study.



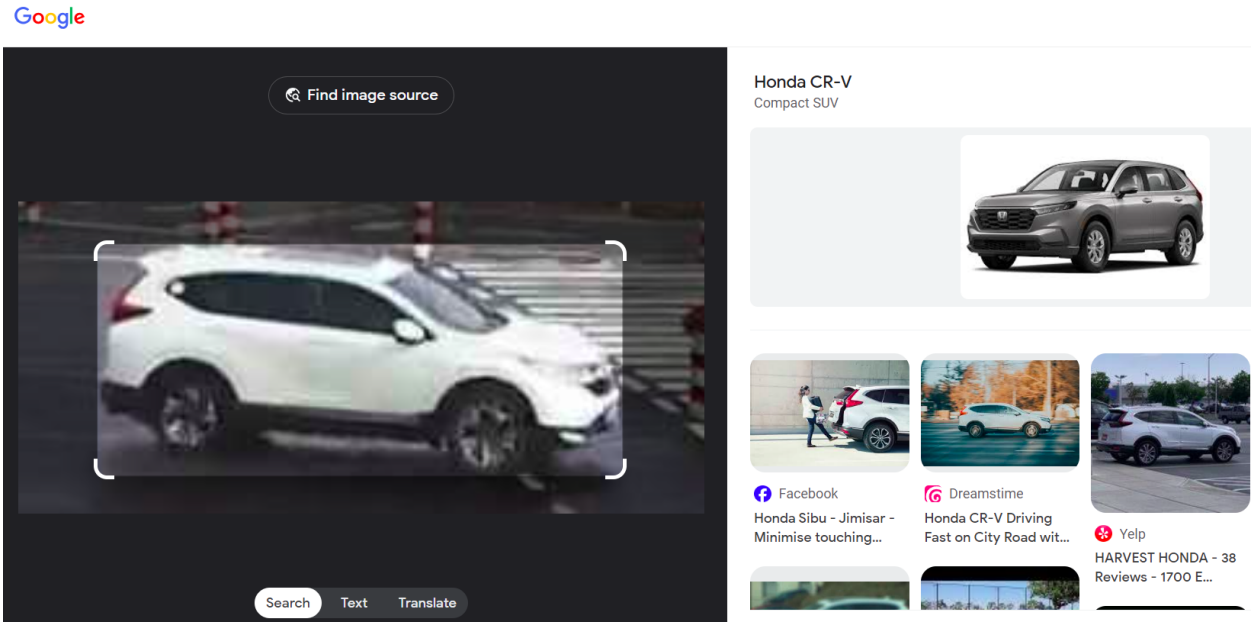
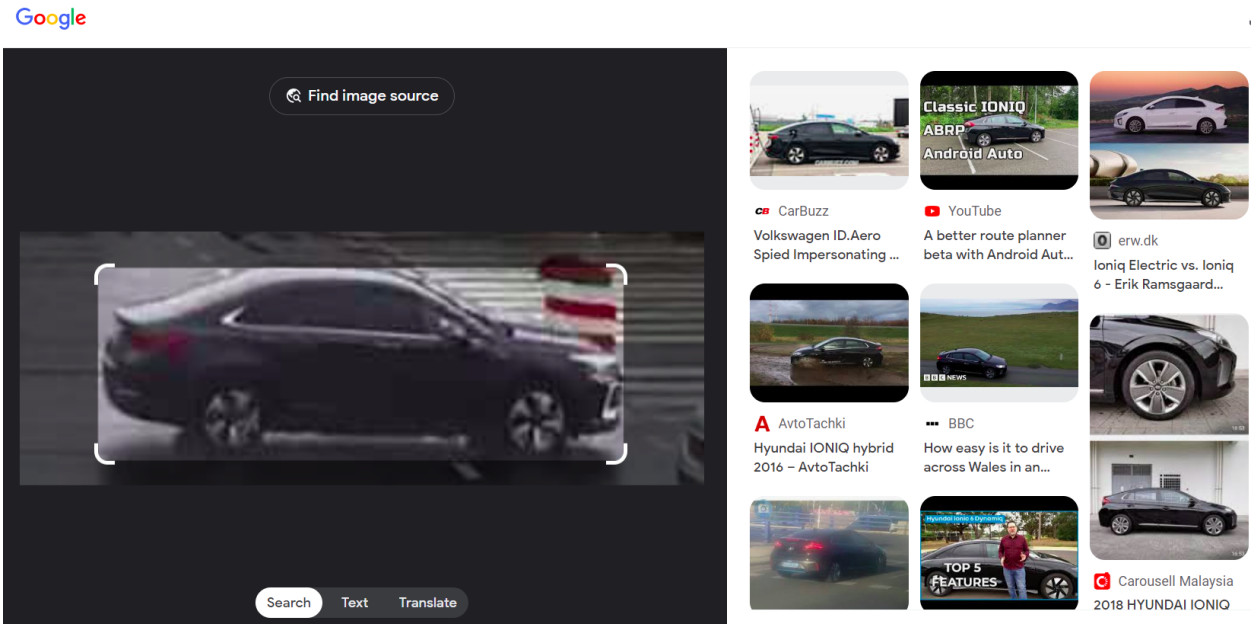


Figure 7.3: Examples of vehicle make and model using Google Image reverse search.



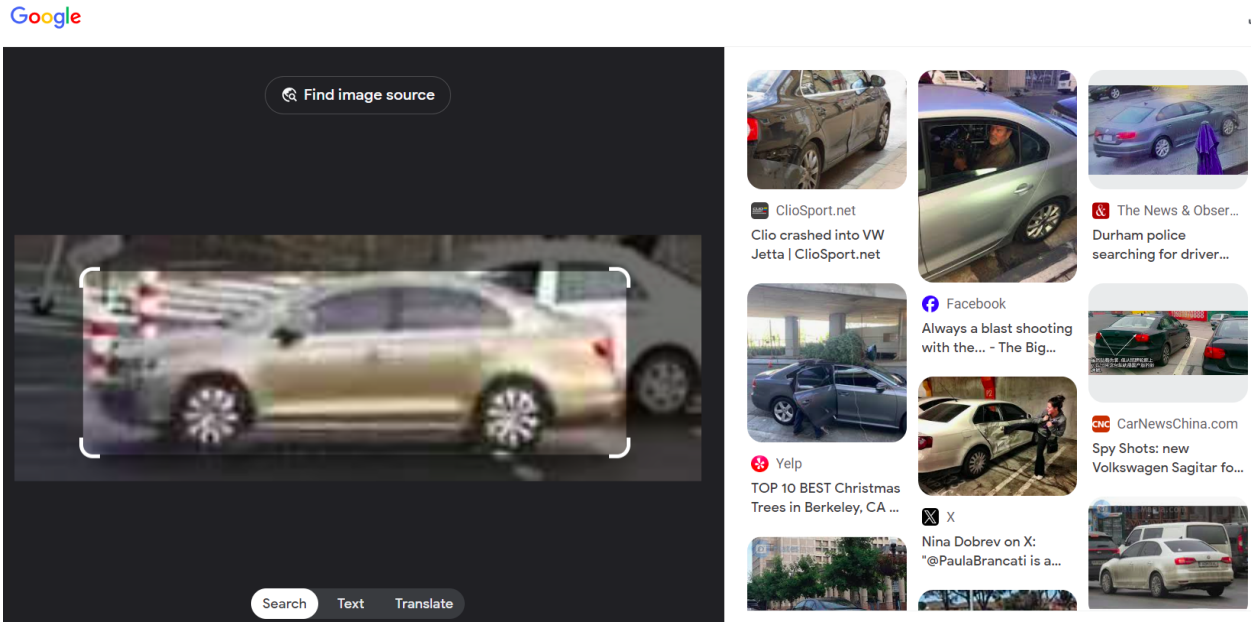


Figure 7.4: Two cases when Google Image reverse search returns ambiguous results.

To make a fair comparison, we create a non-propagated LiDAR bounding box estimate by taking the following steps:

- Collect all points within the original LiDAR-derived bounding box.
- Use a 2D minimal bounding box function (Diener, 2024) to create a new minimally-containing BEV footprint.
- Set the vertical extent of the bounding box to range from the lowest point to the highest point within the original box.
- The smaller horizontal dimension of the box is taken as the width, and the longer dimension as the length.

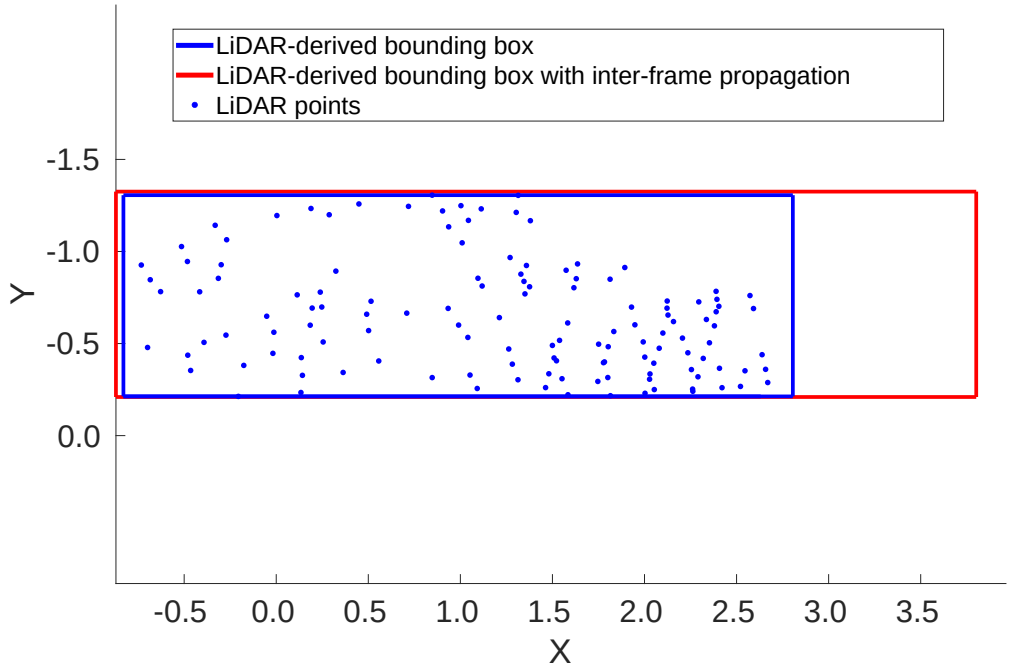
An example is shown in Figure 7.5. Table 7.3 shows that without inter-frame propagation, the dimensions from LiDAR-derived ground-truthing are significantly underestimated when compared to the detected make and model's dimensions. The most significant change is for length where the bias jumps from -0.40m for LiDAR with inter-frame propagation to -1.37m for LIDAR without inter-frame propagation, suggesting that LiDAR-based ground-truthing depends heavily on frames where LiDAR returns are not sparse.

Table 7.3: Comparison of estimated vehicle dimensions of OUR method, LiDAR (LD), and LiDAR with inter-frame propagation (LD_{prop}) against real ground truth. N = **697** vehicles. **569** out of **697** vehicles have wheelbase estimates.

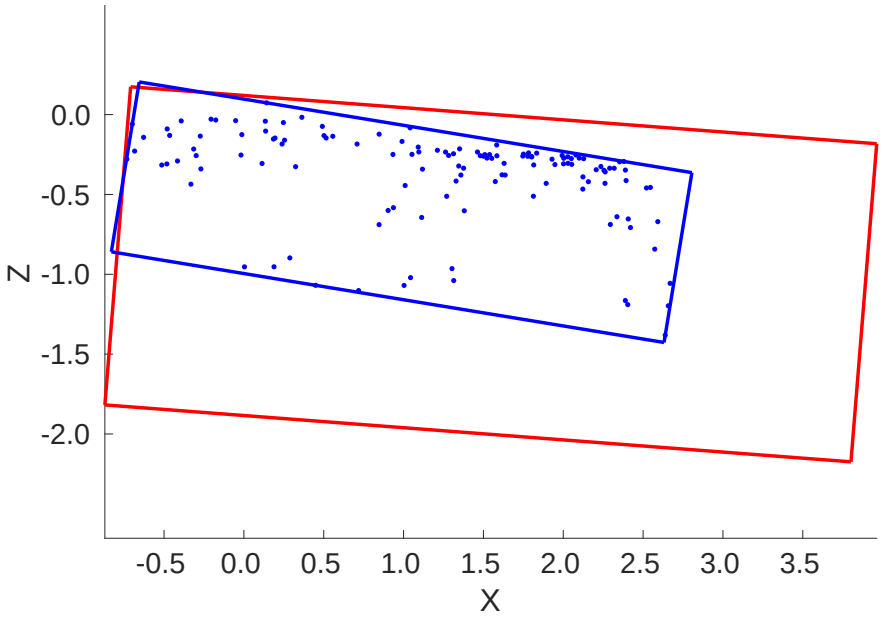
	Bias (m)			MAE (m)			MAE (%)		
	LD	LD_{prop}	OUR	LD	LD_{prop}	OUR	LD	LD_{prop}	OUR
Height	-0.61	-0.49	0.06	0.61	0.49	0.15	39.7	31.4	10.0
Length	-1.37	-0.40	0.24	1.37	0.42	0.45	29.0	8.8	9.5
Width	-0.40	-0.05	-0.01	0.41	0.17	0.11	22.2	9.2	6.2
Wheelbase	–	–	-0.08	–	–	0.23	–	–	8.12

Evaluation against LiDAR-derived bounding box with inter-frame propagation

Table 7.3 shows that symmetry-based estimates of vehicle length are comparable in accuracy to LiDAR-based estimates with inter-frame propagation, obtained directly from the DAIR-V2X dataset (Yu et al., 2022). Symmetry-based estimates of vehicle height are substantially better (0.15m vs 0.49m MAE), and symmetry-based estimates of vehicle width are also superior (0.11m vs 0.17m MAE). Note that extrapolating LiDAR estimates from near-field frames where the vehicle assumes an advantageous pose greatly improves the accuracy of LiDAR-derived estimates. We believe that a similar strategy could be applied to our symmetry-based approach to further enhance accuracy, and we look forward to exploring this in future work.



(a) Side view of the vehicle



(b) BEV of the vehicle

Figure 7.5: LiDAR-derived bounding box (blue) and LiDAR-derived bounding box with inter-frame propagation (red).

Unlike the LiDAR-based approach, our method often generates good estimates of the wheelbase with an MAE of 0.23 meters. This is an advantage of our method over the LiDAR method.

7.3 Evaluating consistency within a track

We aim to evaluate whether our predictions are consistent within each track, using the same track information generated in chapter 3.

To analyze the consistency, we have the following metrics:

Mean Standard Deviation of a dimension within a track:

$$\text{MSD (m)} = \frac{1}{N}\sum_{i=1}^{N}\sigma_i \tag{7.4}$$

Mean Percent Standard Deviation of a dimension:

$$\text{MPSD (\%)} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\sigma_i}{\mu_i}\right)\times 100 \tag{7.5}$$

where N is the number of tracks, σ_i is the standard deviation of the estimated dimension within a track, and μ_i is the mean of the estimated dimension within that track.

Table 7.4: Consistency of vehicle dimension estimates of OUR method, LiDAR (LD), and LiDAR with inter-frame propagation (LD_{prop}) over time. MSD: Mean standard deviation. MPSD: Mean percent standard deviation. **N = 375** tracks containing **2050** vehicle instances.

	MSD (m)			MPSD (%)		
	LD	LD_{prop}	OUR	LD	LD_{prop}	OUR
Height	0.08	0.06	0.08	9.4 %	5.8 %	5.0 %
Length	0.47	0.15	0.35	15.5 %	3.4 %	7.2 %
Width	0.20	0.12	0.11	15.9 %	7.0 %	6.4 %

Table 7.4 shows the consistency over time of LiDAR-based and symmetry-based vehicle dimension estimates. For symmetry-based estimates, we find that the mean standard deviation varies between 5.0% and 7.2%, depending on the dimension. Without inter-frame propagation across frames, the mean standard deviation of LiDAR-derived estimates varies between 9.4% and 15.9%. Inter-frame propagation reduces this to 3.4% to 7.0%, in the same range as our single-frame symmetry-based estimates. This further substantiates the heavy reliance on LiDAR-based ground truthing on inter-frame propagation.

7.4 Ablation studies

7.4.1 Method to merge annotations from two annotators

Given that we obtain annotations from two different annotators, we evaluate a strategy for merging annotations. In this strategy, we follow these steps when merging the annotations:

- Use all annotated symmetry pairs.
- Average the locations of tire-ground contact points in 2D images if they are close to each other within a threshold of 10 pixels. We exclude annotations where tire-contact points are annotated for the same type of tire-ground contact, but are further than 10 pixels apart, as this suggests erroneous annotations.

Referring to Table 7.5, we can see that this merging strategy proves effective in modeling errors. The remaining results in this Evaluation section are the results of using merged annotations from two annotators.

Table 7.5: Comparison of performance when using different annotations

Metric	**Person 1**	**Person 2**	**Merged**
Number of vehicles	2009	2009	2009
Mean reprojection error (pixels)	2.33	2.51	2.20
IOU BEV	0.49	0.51	0.50
IOU	0.34	0.36	0.35
MAE of distance to center of LiDAR-derived bounding box (m)	1.23	1.13	1.16
Bias of distance to center of LiDAR-derived bounding box (m)	-0.77	-0.61	-0.68
Mean heading angle error (degrees)	3.99	3.37	3.56
MAE Height (m)	0.49	0.49	0.50
MAE Length (m)	0.62	0.65	0.64
MAE Width (m)	0.25	0.28	0.26

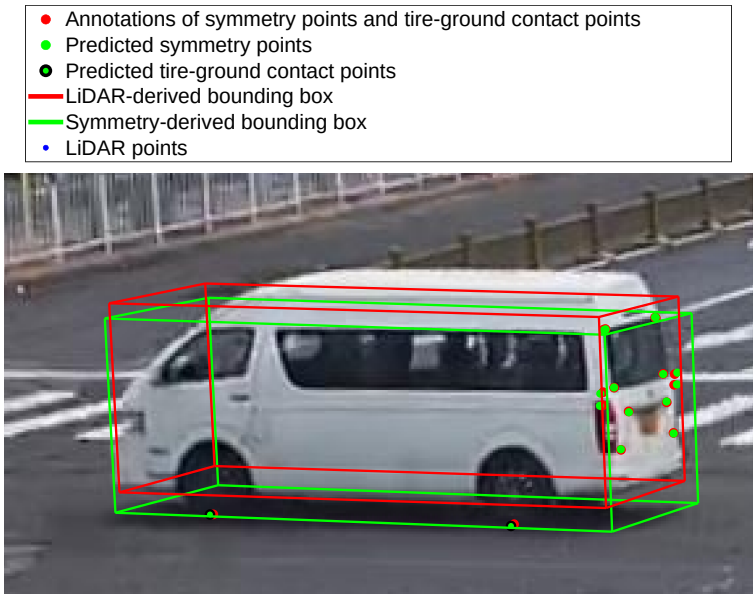
7.5 Qualitative results

Below, we present example qualitative results obtained using our symmetry-based method for 3D vehicle ground truthing.

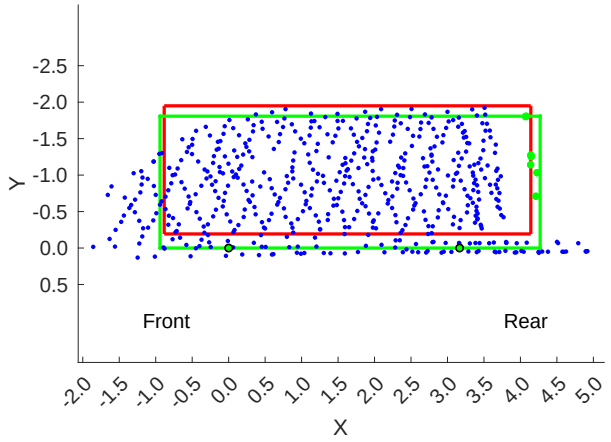
Figure 7.6, Figure 7.7, and Figure 7.8 show our best, median, and worst results on the DAIR-V2X dataset, respectively, as measured by IoU with LiDAR-based ground truth. In the best case shown in Figure 7.6, both LiDAR- and symmetry-based bounding box estimates seem to be accurate. The median case shown in Figure 7.7 provides a good illustration of errors that occur when the symmetry plane approaches the optical axis of the camera. In this case, a relatively small error in the estimated position of the symmetry plane results in a relatively large underestimation of the distance to the symmetry points at the rear of the vehicle. Note, however, that the LiDAR-based ground truth also significantly underestimates the height of

the vehicle. In the worst-case example shown in Figure 7.8, the reprojection error is small, and the estimated heading angle is correct, but the symmetry-based vehicle estimate is displaced to be further away than the LiDAR-based estimate. This is likely due to a small error in the relative tilt calibration of the two sensors, which, given the oblique angle of the road surface relative to the optical axis at longer distances, can easily lead to larger errors. In this case, the offset is on the order of 5m, which, at this viewing distance of 55.6m and camera height of 7.06m, could be caused by a relative tilt error of only 0.6 deg.

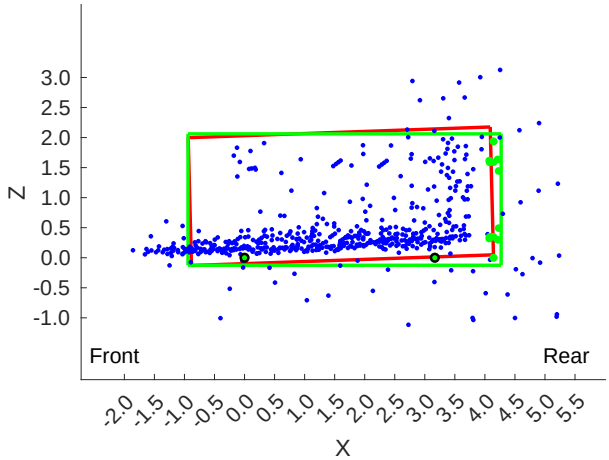
This inter-sensor calibration error should not necessarily be seen as a flaw in the monocular symmetry-based approach, since it could equally be seen as an error in the LiDAR calibration. For this dataset, we used these LiDAR data to derive a terrain model, and so inter-sensor calibration error will also displace our symmetry-based estimates relative to the terrain. But in general, we would expect terrain models to be derived from independent aerial photogrammetry or LiDAR surveys, which means that both monocular and LiDAR estimates of 3D bounding boxes could be subject to calibration errors. Note again the large height error of the LiDAR-based ground truth.



(a) Camera projection. Note that annotated points are largely occluded by reprojected points, reflecting a good quality of fit.

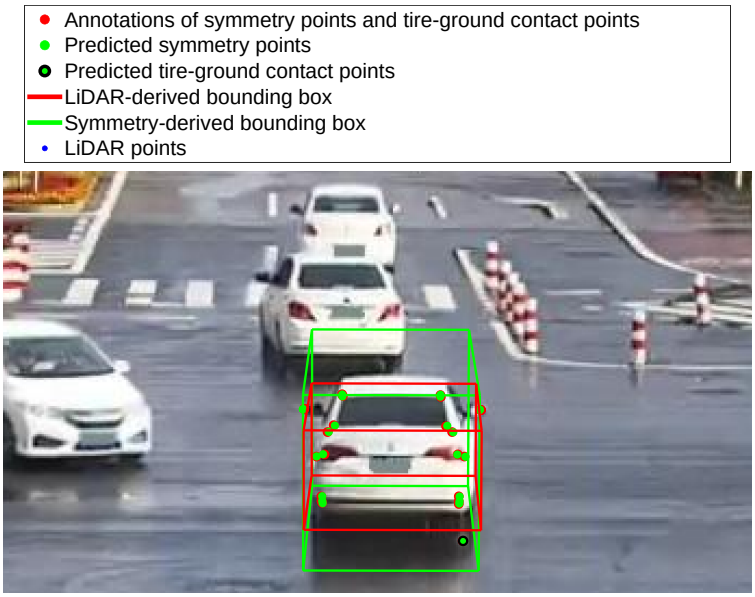


(b) Side view.

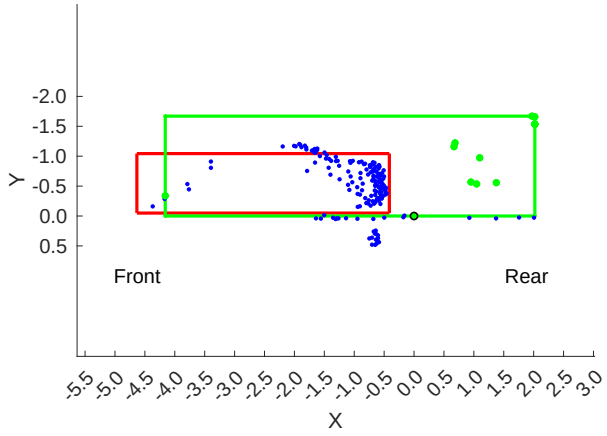


(c) Bird's-eye view.

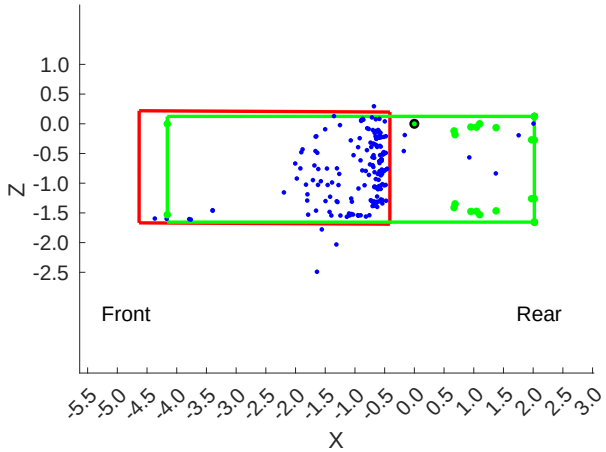
Figure 7.6: An instance achieving the highest IoU with LiDAR-based ground truth (**IoU = 0.76**).



(a) Camera projection. Note that annotated points are largely occluded by reprojected points, reflecting a good quality of fit.

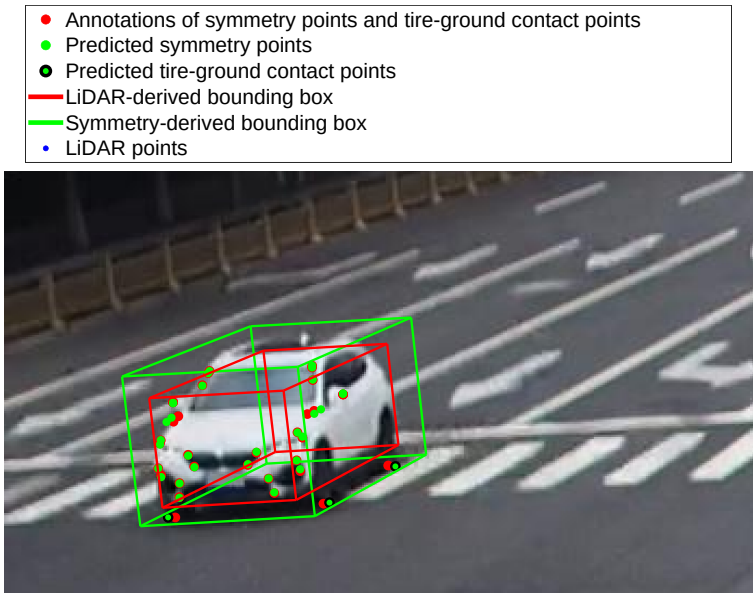


(b) Side view.

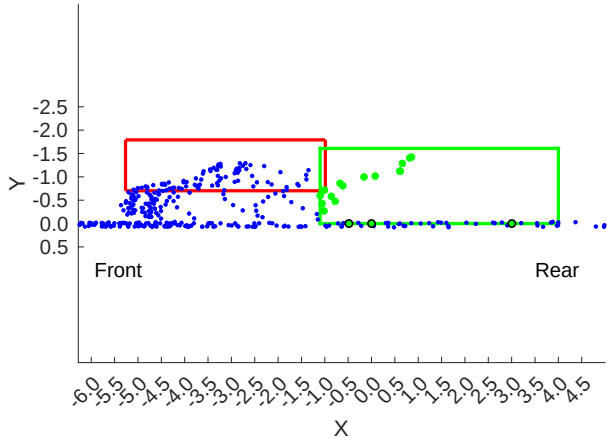


(c) Bird's-eye view.

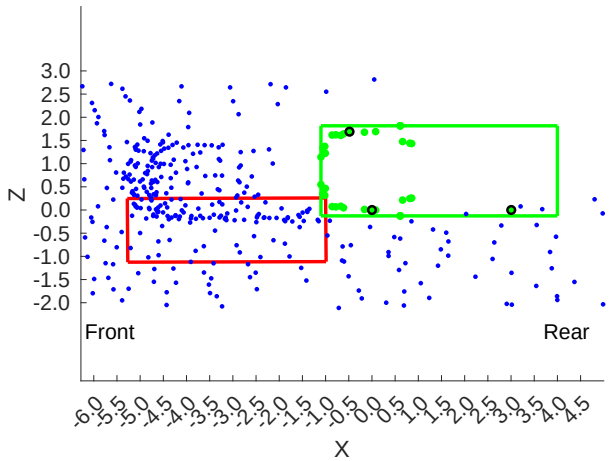
Figure 7.7: An instance achieving the median IoU with LiDAR-based ground truth (**IoU = 0.34**).



(a) Camera projection. Note that annotated points are largely occluded by reprojected points, reflecting a good quality of fit.



(b) Side view.



(c) Bird's-eye view.

Figure 7.8: An instance achieving the lowest IoU with LiDAR-based ground truth (**IoU = 0.002**).

Chapter 8

Discussion

8.1 Limitations

Our method has several limitations:

- Annotations of points on the vehicles are done manually. Noisy annotations can lead to decreased accuracy.
- The predicted locations of 3D points in our method are sensitive to the annotations, especially when the estimated symmetry plane is almost parallel to the line of sight.
- For some vehicles, identifying pairs of extremal symmetry points that align vertically with tire-ground contact points can be challenging, potentially reducing the performance of our algorithm. We have partially addressed this issue through a rectification process (see section 6.2).
- The quality of terrain data can impact the accuracy of symmetry-based ground truthing. These errors will have the greatest impact on the estimated location of vehicles, particularly for those in the far field where the view angle is highly oblique to the terrain surface. Terrain data are generally available. Once the location of deployment is known, online sources offer global terrain data; for instance, 30m resolution global data is available from the U.S. Geological Survey (2024). Also, L3Harris Technologies (2024) provides higher resolution (2.5 – 5m) global data.
- Errors in camera calibration will lead to errors in symmetry-based 3D vehicle estimates.
- We assume that a vehicle does not tilt or roll. Since the LiDAR-derived 3D ground truth bounding boxes for both datasets are constrained to have zero roll and zero tilt, relaxing our assumption is likely

to increase the error relative to this ground truth. Nevertheless, this remains an important topic for future research, as in real-world scenarios, this assumption may not always hold. Depending on road surfaces and vehicle movements, vehicles can sometimes exhibit tilt or roll.

- While RGB sensing generally provides higher spatial resolution than LiDAR, the number of pixels on a vehicle will also of course decline with distance and could lead to decreased accuracy in the far field. Interestingly, we did not witness this degradation for most metrics, presumably because resolution remained sufficiently high out to 150m, although we did see some decline in the heading accuracy, which seems most sensitive to resolution.

- Occlusions can hinder the identification of a sufficient number of symmetry points and/or tire-ground contact points for accurate 3D estimation.

8.2 Conclusion

We have developed and evaluated a novel symmetry-based approach to vehicle ground truthing that employs a single RGB camera and terrain data rather than LiDAR. We found the method to vastly outperform single-frame LiDAR ground truth, and to be comparable in most respects to multi-frame propagated LiDAR ground truth. Annotations are relatively quick (about 2 minutes per vehicle) and quite accurate even in the far field, at least out to 150m in our experiments. This novel approach may thus be useful when far-field ground truth is important and/or LiDAR is not available due to cost or other restrictions. Our work demonstrates that LiDAR-free 3D ground truth is feasible and worthwhile for traffic analytics, offering reliable and explainable results at a fraction of the cost of LiDAR systems. The 3D bounding boxes generated by our method can then be used to train monocular 3D object detection algorithms where LiDAR data are not available.

8.3 Future Work

LiDAR-derived bounding box dimensions can be significantly affected by calibration errors, as seen in the height dimension for DAIR-V2X Yu et al. (2022). After filtering out these low-quality bounding boxes, it is evident that low-quality 3D LiDAR-derived bounding boxes may affect the evaluation of our method. Thus, securing high-quality, reliable 3D ground truth is critical in further evaluating the accuracy of our system.

There are future improvements that can be made to further enhance the symmetry-based ground truthing method. Building on the observation that dimension constraints from the Canadian Vehicle Specifications (CVS) database Canada (2023) help in reducing failure cases while achieving improved performance, we

can use such databases to generate priors over vehicle dimensions that are then combined with likelihoods over these dimensions derived from the symmetry-based point cloud estimates. Additionally, it would be interesting to track objects and use automatic methods to propagate annotations and 3D models over time. We can adopt the consistency constraint across vehicles within the same track to leverage privileged views of a moving vehicle.

Our current method also assumes that the vehicle is horizontal, as do LiDAR-based methods. Future work will explore whether allowing modest amounts of roll and pitch can improve accuracy. This could be partially addressed by taking the terrain slope into account. Occlusion is a problem for both LiDAR- and symmetry-based ground truthing. This can often be addressed by identifying frames where a vehicle is largely unoccluded and then propagating the model over time. Currently, we analyze only unoccluded vehicles. Future work will explore the degree to which prior models can be used to complete partially occluded vehicles.

Bibliography

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., and Beijbom, O. (2020). nuScenes: A Multimodal Dataset for Autonomous Driving. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11618–11628, Seattle, WA, USA. IEEE.

Canada, T. (2023). Canadian Vehicle Specifications (CVS) Database. Available online: https://open.canada.ca/data/dataset/913f8940-036a-45f2-a5f2-19bde76c1252.

Chang, A. X., Funkhouser, T. A., Guibas, L. J., Hanrahan, P., Huang, Q.-X., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., and Yu, F. (2015). Shapenet: An information-rich 3d model repository. *ArXiv*, abs/1512.03012.

Chen, Z., Gao, B., and Devereux, B. (2017). State-of-the-Art: DTM Generation Using Airborne LIDAR Data. *Sensors*, 17(1):150.

Collins, R. and Weiss, R. (1990). Vanishing Point Calculation as a Statistical Inference on the Unit Sphere. In *[1990] Proceedings Third International Conference on Computer Vision*, pages 400–403.

Corral-Soto, E. R. and Elder, J. H. (2017). Slot Cars: 3D Modelling for Improved Visual Traffic Analytics. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 889–897, Honolulu, HI, USA. IEEE.

Coughlan, J. and Yuille, A. (1999). Manhattan World: Compass Direction from a Single Image by Bayesian Inference. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, pages 941–947 vol.2, Kerkyra, Greece. IEEE.

CVAT.ai (2023). Computer Vision Annotation Tool (CVAT).

Denis, P., Elder, J. H., and Estrada, F. J. (2008). Efficient Edge-Based Methods for Estimating Manhattan Frames in Urban Imagery. In Forsyth, D., Torr, P., and Zisserman, A., editors, *Computer Vision – ECCV 2008*, volume 5303, pages 197–210. Springer Berlin Heidelberg, Berlin, Heidelberg.

Diener, J. (2024). 2D minimal bounding box. https://www.mathworks.com/matlabcentral/fileexchange/31126-2d-minimal-bounding-box. MATLAB Central File Exchange. Retrieved May 11, 2024.

Du, Y., Zhao, Z., Song, Y., Zhao, Y., Su, F., Gong, T., and Meng, H. (2022). StrongSORT: Make DeepSORT Great Again. https://arxiv.org/abs/2202.13514v2.

Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. (2013). Vision Meets Robotics: The KITTI Dataset. *The International Journal of Robotics Research*, 32(11):1231–1237.

Jocher, G., Chaurasia, A., and Qiu, J. (2023). YOLO by Ultralytics.

L3Harris Technologies (2024). Aw3d standard by l3harris. L3Harris Technologies. Accessed: 2024-06-02.

Lambert, J., Carballo, A., Cano, A. M., Narksri, P., Wong, D., Takeuchi, E., and Takeda, K. (2020). Performance Analysis of 10 Models of 3D LiDARs for Automated Driving. *IEEE Access*, 8:131699–131722.

Li, Y., Sawada, T., Latecki, L. J., Steinman, R. M., and Pizlo, Z. (2012). A Tutorial Explaining a Machine Vision Model that Emulates Human Performance When It Recovers Natural 3D Scenes from 2D Images. *Journal of Mathematical Psychology*, 56(4):217–231.

Lin, Y., Pintea, S. L., and van Gemert, J. C. (2021). Nerd++: Improved 3d-mirror symmetry learning from a single image. In *British Machine Vision Conference*.

Marquardt, D. W. (1963). An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441.

Pizlo, Z., Li, Y., Sawada, T., and Steinman, R. (2014). *Making a Machine That Sees Like Us*. Oxford University Press.

Qian, Y., Ramalingam, S., and Elder, J. H. (2019). LS3D: Single-View Gestalt 3D Surface Reconstruction from Manhattan Line Segments. In Jawahar, C., Li, H., Mori, G., and Schindler, K., editors, *Computer Vision – ACCV 2018*, volume 11364, pages 399–416. Springer International Publishing, Cham.

remtromol (2021). 3D Model Toyota Corolla (KE80) on Sketchfab. Available online: https://skfb.ly/o87Cs.

ScaleAI (2023). Scale AI Video Annotation. Scale AI, Inc.

Song, S., Lichtenberg, S. P., and Xiao, J. (2015). Sun rgb-d: A rgb-d scene understanding benchmark suite. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 567–576.

Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., Vasudevan, V., Han, W., Ngiam, J., Zhao, H., Timofeev, A., Ettinger, S., Krivokon, M., Gao, A., Joshi, A., Zhao, S., Cheng, S., Zhang, Y., Shlens, J., Chen, Z., and Anguelov, D. (2019). Scalability in Perception for Autonomous Driving: Waymo Open Dataset. https://arxiv.org/abs/1912.04838v7.

Sun, X., Wu, J., Zhang, X., Zhang, Z., Zhang, C., Xue, T., Tenenbaum, J. B., and Freeman, W. T. (2018). Pix3d: Dataset and methods for single-image 3d shape modeling. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2974–2983.

U.S. Geological Survey (2024). Aster global digital elevation model (gdem version 3). U.S. Geological Survey. Accessed: 2024-06-02.

Wang, B., Wu, V., Wu, B., and Keutzer, K. (2019). LATTE: Accelerating LiDAR Point Cloud Annotation via Sensor Fusion, One-Click Annotation, and Tracking. https://arxiv.org/abs/1904.09085v1.

Wang, T., He, C., Wang, Z., Shi, J., and Lin, D. (2020). FLAVA: Find, Localize, Adjust and Verify to Annotate LiDAR-Based Point Clouds. https://arxiv.org/abs/2011.10174v1.

Weisstein, E. W. (2022). Line-Line Distance. https://mathworld.wolfram.com/. Access at: https://mathworld.wolfram.com/Line-LineDistance.html.

Wilson, J. P. (2012). Digital Terrain Modeling. *Geomorphology*, 137(1):107–121.

Yu, H., Luo, Y., Shu, M., Huo, Y., Yang, Z., Shi, Y., Guo, Z., Li, H., Hu, X., Yuan, J., and Nie, Z. (2022). DAIR-V2X: A Large-Scale Dataset for Vehicle-Infrastructure Cooperative 3D Object Detection. https://arxiv.org/abs/2204.05575v1.

Zhou, D., Fang, J., Song, X., Guan, C., Yin, J., Dai, Y., and Yang, R. (2019). IoU Loss for 2D/3D Object Detection. https://arxiv.org/abs/1908.03851v1.

Zhou, Y., Liu, S., and Ma, Y. (2021). Nerd: Neural 3d reflection symmetry detector. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15935–15944.

Zimmer, W., Rangesh, A., and Trivedi, M. (2019). 3D BAT: A Semi-Automatic, Web-based 3D Annotation Toolbox for Full-Surround, Multi-Modal Data Streams. https://arxiv.org/abs/1905.00525v1.