Accurate Automatic 3D Annotation of Traffic Lights and Signs for Autonomous Driving

Sándor Kunsági-Máté Levente Pető Lehel Seres
Tamás Matuszka
aiMotive
{sandor.kunsagimate, levente.peto, lehel.seres, tamas.matuszka}@aimotive.com

Abstract

3D detection of traffic management objects, such as traffic lights and road signs, is vital for self-driving cars, particularly for address-to-address navigation where vehicles encounter numerous intersections with these static objects. This paper introduces a novel method for automatically generating accurate and temporally consistent 3D bounding box annotations for traffic lights and signs, effective up to a range of 200 meters. These annotations are suitable for training real-time models used in self-driving cars, which need a large amount of training data. The proposed method relies only on RGB images with 2D bounding boxes of traffic management objects, which can be automatically obtained using an off-the-shelf image-space detector neural network, along with GNSS/INS data, eliminating the need for LiDAR point cloud data.

1 Introduction

Autonomous driving is currently one of the most actively researched fields. Given the complexity of the problem, recent advancements focus on perceiving the entire three-dimensional environment around the vehicle. This comprehensive approach is essential because of the myriad traffic scenarios and interdependencies between objects, making two-dimensional object detection insufficient due to the lack of depth information. For instance, detecting a red light in a self-driving car's camera image does not necessarily mean the vehicle must stop. How far away is the traffic light? Is it relevant to the lane in which the ego vehicle is located? To answer these questions, the 3D positions of the objects have to be known.

Deep learning models currently used in self-driving cars require a vast amount of training data to ensure accurate predictions in all scenarios. As a consequence, there is a need to label every dynamic and static object with 3D bounding boxes and additional attributes over hundreds or thousands of hours of driving. However, manually creating these labels is expensive, time-consuming, and error-prone. While several datasets with 3D bounding box annotations are available for dynamic objects [18], [2], [10], [13], the number of available static object datasets with 3D annotations [8], especially those containing distant objects, is remarkably limited. As a result, there is a significant interest in automating the generation of such training data without human intervention. Although there are several large traffic light and sign datasets (LaRa [4], BSTLD [1], LISA [12], DTLD [9]), these contain only two-dimensional bounding box annotations. Our primary goal is to provide accurate 3D bounding boxes for traffic management objects, ensuring that the projected 2D bounding boxes in the camera image encompass objects from a wide range of viewing angles and distances. This step is crucial for all downstream tasks of the proposed method, such as classification or optical character recognition. Since the data recording process typically involves multiple sensors and a high frame rate, this requirement is easily met.

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

The main contribution of this work is a novel method that provides accurate positioning with an average mean distance of 0.2-0.3 meters and temporally consistent 3D bounding boxes of traffic management objects up to 200 meters away. Our method also determines additional attributes such as traffic light state, traffic light mask type, traffic sign type, and occlusion. The proposed solution is simple yet effective, relying solely on 2D images and Global Navigation Satellite System/Inertial Navigation System (GNSS/INS) data, without the need for expensive active sensors like LiDAR. Furthermore, we publish a representative dataset, automatically generated using our algorithm, under a CC BY-NC-SA 4.0 license, allowing the research community to use it for non-commercial research purposes[1]. To our knowledge, no publicly available large-scale dataset including distant objects currently exists that contains accurate 3D bounding boxes of traffic management objects, particularly traffic lights.

2 Related Work

Automatic 3D localization methods for static objects, particularly traffic signs, are already available with certain limitations. The three main approaches are the following: 1) using LiDAR point cloud data to identify the cluster associated with the object; 2) generating a synthetic point cloud through Structure-from-Motion and associating 2D image-space detections to the resulting 3D points; and 3) applying triangulation using camera images, GNSS, and orientation information.

Approach 1) is well-suited for traffic signs due to their highly reflective coating, which produces dense point groups in LiDAR data with high-intensity values that can be effectively clustered. Soilán et al. in [16] used this technique to localize traffic signs, reprojecting them onto 2D camera images to spatially and temporally synchronize with the point cloud data. While this method can yield accurate results, separating traffic signs close to each other is challenging. Another drawback, as they noted, is that in urban environments, the rate of false positive detections increases due to the higher number of reflective objects. A similar approach [11] was presented by Ghallabi et al., but in their case, no camera information was used and the method was only tested in a highway environment. Song and Myung described a method in [17] that also utilizes 2D image detection and LiDAR point cloud data. They first apply a deep learning model to camera images to predict 2D bounding boxes of traffic signs. These boxes are then used to filter relevant parts of the point cloud within a frustum, and DBSCAN clustering is applied to eliminate non-relevant point groups. However, this group of work depends heavily on the quality of the point cloud. For traffic signs located far from the observer or higher than the LiDAR detection range, few or no reflective points are detected, leading to low localization accuracy and an increased number of false negative detections. Additionally, this method is ineffective for traffic lights, as they are mostly black and have lower reflectivity. Moreover, most traffic lights are positioned higher than the detection range of LiDAR sensors.

Approach 2) is primarily used to create large-scale but low-resolution maps of traffic signs. Structure-from-Motion relies on identifying features in consecutive camera images, associating them, and estimating their 3D position through triangulation, thereby generating a synthetic point cloud from the images. Musa's solution [15] is based on this method and further improves localization accuracy using the GNSS coordinates of the images. Although the algorithm runs in real-time, its accuracy is around 2.75 meters, which is insufficient for automated ground truth data generation. Mapillary Traffic Sign Dataset (MTSD) [5] provides a world-scale map of traffic management objects using dashcam images and Structure-from-Motion. However, based on our experiments, the accuracy is also within several meters, and only latitude/longitude positions can be downloaded. No 3D bounding boxes are available that could be projected onto camera images. Therefore, this solution cannot be used for automated ground truth generation either.

The last group of methods relies on image-space detections, GNSS, and orientation information. Mentasti et al. developed a localization algorithm [14] for traffic lights, which they applied to the DriveU Traffic Light Dataset [9]. They estimated individual distances of traffic lights for each 2D detection using disparity maps, applied a tracking algorithm, and finally averaged the positions for each track ID. However, the 3D position estimation was not validated since the DriveU dataset only provides 2D bounding boxes of traffic lights. Fairfield and Urmson used a traffic light detection algorithm [7] that identifies brightly colored red, amber, and green blobs in the image. These detections are then associated between frames using image-to-image association and least squares

[1] https://github.com/aimotive/aimotive_tl_ts_dataset

triangulation. The orientation of the traffic light is estimated as the reciprocal heading of the mean car heading over all the image labels used to estimate the traffic light position. In traffic light online detection, the map positions are projected into the image plane, and a region of interest is defined, considering a larger area than the predicted bounding box. Finally, the classifier is applied to the image cutouts to find the light blobs and classify the colors. Since disparity-based depth estimation is known to be inaccurate in long distances and color-based blob detection is not applicable in the case of traffic signs, these methods cannot be applied to accurate 3D automatic annotation of traffic lights and signs.

To summarize, there is currently no comprehensive algorithm for automatically generating high-precision 3D bounding boxes (including distant objects) of traffic signs and lights with additional attributes. The existence of such an algorithm could have a significant impact on the development of image-based neural networks used by self-driving vehicles.

3 Automatic Annotation of Traffic Lights and Signs in 3D

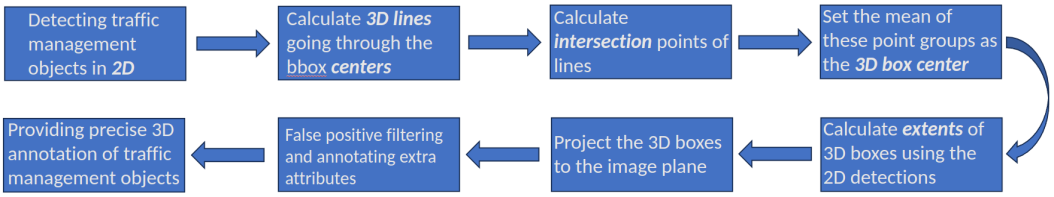

Figure 1: The main steps of the automatic annotation method.

Our proposed method, depicted by Figure 1, can be used for generating unlimited amounts of 3D training data for traffic management objects. This automatic annotation algorithm consists of five steps: 1) Mask2Former [3] image segmentation model is used to obtain the 2D positions of traffic lights and traffic signs; 2) 3D bounding box centers are localized by triangulating the lines of sight in the Earth-centered, Earth-fixed coordinate system (ECEF), resulting in a 3D map of traffic management objects; 3) 3D bounding box extent and orientation are estimated; 4) 3D boxes are transformed into the instantaneous coordinate systems (i.e., vehicle coordinate system) of each frame; and 5) 3D boxes are projected onto the camera image plane and 2D image cutouts of traffic management objects are classified. The outcome of the proposed method is a dataset containing 3D annotations of traffic lights and traffic signs for each frame, including information on color state, occlusion, traffic light mask type, and traffic sign type. We describe the details of the main steps of our method in the following subsections.

3.1 3D Localization

The first step in 3D localization involves acquiring 2D detections of traffic management objects in images captured by a single front camera. Then, the bounding boxes are calculated and the centers of the bounding boxes are stored. Only predicted 2D bounding boxes with high confidence are used, thereby excluding false positive detections. This step does not reduce the recall of 3D detection, as traffic management objects will typically be close to the ego vehicle's trajectory during recording and will appear large enough in the images over a sufficient time horizon to ensure highly confident 2D predictions.

The next step is to calculate the 3D positions of these static objects. To apply the triangulation technique, 2D observations of the same physical 3D point from multiple viewing angles are needed. Since traffic lights are relatively small and compact objects and traffic signs are planar, the center of the 2D bounding box can be treated as the projection of the same physical point with good approximation. Using the GNSS and orientation data of the observer along the ego vehicle's trajectory, as well as the 3D lines pointing towards the 2D bounding box centers, 3D positions of the object center in a global coordinate system through the triangulation technique illustrated in Figure 2 are determined.

Specifically, 3D lines that come closer than 10 centimeters to each other are collected. Then, the coordinates of the point closest to the lines are calculated by iterating over these line pairs. This process generates many candidate points for the centers of 3D boxes, which are then aggregated

using the DBSCAN clustering method [6]. A 3D point forms a cluster if there are at least 3-5 points within 5-10 centimeters of each other. After identifying these clusters, their average is taken as the final prediction of the 3D box center in ECEF coordinates. The distance filtering and clustering steps enhance the algorithm's robustness against random errors related to GNSS position, orientation, or camera calibration. It's important to note that this method does not require object tracking, as localization is calculated directly in the global coordinate system. This leverages the fact that the likelihood of incorrectly associating two 2D detections from different physical objects in 3D space, given such low distance threshold values in the triangulation process, is very low.

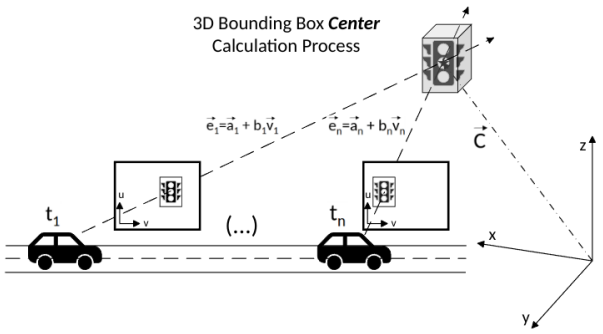

Figure 2: Calculation of 3D bounding box center.

3.2 Extent Calculation

The map with the bounding box centers of traffic management objects is provided after the localization step. However, the extent of the detected objects is still unknown. To determine this attribute of traffic lights, the intersections of the lines pointing towards the 2D bounding box corners with a vertically aligned plane that contains the center of the object and is perpendicular to our line of sight in the x-y plane are calculated. In this step, the cross-sections of the 3D bounding boxes from various viewing angles are measured. Finally, the widths and heights of these cross-sections are averaged to estimate the width, depth, and height of the 3D bounding boxes. Note that the width and depth are set to the same value, which is a good estimate for the commonly vertically aligned traffic lights. The visualization of the traffic light size estimation method is illustrated in Figure 3.

Traffic signs have a larger variety of shapes and can appear in shapes other than rectangles (e.g., circles, triangles). Therefore, instead of using the corners of the 2D bounding boxes, the intersections of the vertical plane and the lines pointing toward the edge points of the bounding box are calculated. Since traffic signs are planar objects, the maximum of the measured widths are taken and the depth is set to 10 centimeters.

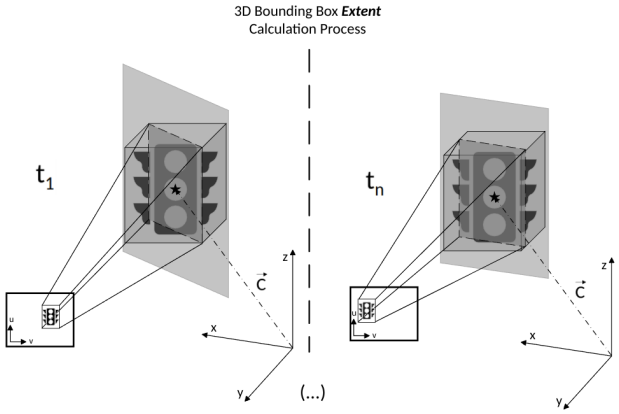

Figure 3: Calculation of 3D bounding box extent.

3.3 Orientation Estimation

Our proposed algorithm employs a heuristic approach to determine the orientation of traffic lights. The orientation estimation method identifies the frame where the vehicle is approximately 10 meters in front of the traffic light and assumes it is oriented opposite to the direction of travel. While this method generally provides accurate orientations for relevant traffic lights, it may be incorrect for cross-traffic ones. However, this does not affect the generation of 2D image cutouts for classification tasks, as the 2D projection of vertically aligned traffic light boxes remains relatively consistent regardless of different rotation angles around the Z axis (see Fig. 4).

For traffic signs, the algorithm uses the line-of-sight vector to the road sign in the frame where the measured width is maximal. The final orientation is the reverse of this vector, indicating the vehicle was closest to being directly opposite the corresponding traffic sign.

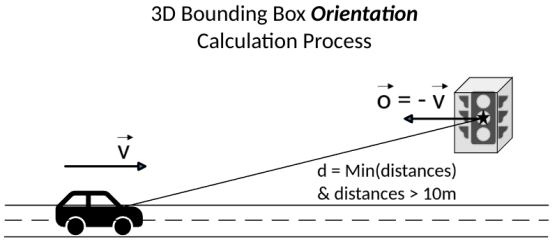

Figure 4: Calculation of 3D bounding box orientation.

3.4 Reducing False Positive Detections

At this stage, a map of 3D bounding boxes for traffic management objects with high positional accuracy (within 0.2-0.3 meters from the ground truth, see details in Section 5) is created, which can be used in various operational design domains such as rain, night, snow, etc. From this map, we generate 2D image cutouts of traffic management objects by projecting them onto the camera image plane, up to 200 meters from the ego vehicle position. Based on our experience, measurement errors in the triangulation technique can produce false positive boxes that are located on the same 3D lines as the true positive box. These false positives can be eliminated by associating their 2D projections with the original 2D bounding boxes. During this process, we first calculate the intersection-over-union (IoU) between the projections and the 2D bounding boxes, associating the average IoU value over the frames for each 3D bounding box. We then group 3D boxes that appear very close to each other, defined by an angle between their line of sight vectors below 0.25-0.3 degrees across several camera frames. Finally, we select the 3D box with the highest IoU value from each group as the final prediction.

3.5 Classification of Object Attributes

When considering the attributes of traffic management objects, we differentiate between time-dependent and time-independent properties. Time-dependent attributes, such as the traffic light color or the occlusion of traffic management objects, must be classified for each frame, which can be challenging when the object is far away from the ego vehicle. In contrast, time-independent attributes, such as the types of objects (e.g., forward arrow traffic light or yield, stop sign), do not change over time. Therefore, we can use high-resolution image cutouts when the ego vehicle is close to the objects. To automatically classify these attributes, we utilize standard convolutional neural networks.

4 3D Traffic Light and Road Sign Dataset

To facilitate research in static 3D object detection and address the challenges mentioned in Section 1, we have published a diverse training dataset of traffic lights and road signs, generated by our method described in Section 3. The recordings were captured in two countries (California, US, and Hungary) in urban and highway environments, and under different times of day and weather conditions. The dataset includes approximately 50,000 3D auto-annotated frames from 220 sequences, each 15

Figure 5: Samples from the dataset with 3D traffic sign and light annotations. The bounding boxes are automatically generated by our method. Traffic light states are color-coded.

[h!]

Table 1: Comparison of datasets.

	nuScenes [2]	Waymo [19]	DTLD (v2.0) [9]	MTSD [5]	Ours
Hours	5.5	6.4	NA	NA	1
Annotated frames	40k	230k	40k	52k	50k
Cameras	6	5	1 (stereo)	1	4
Traffic light 3D boxes	-	-	2D only (292k)	-	320k
Road Sign 3D boxes	-	3.2M	-	2D only (206k)	550k

seconds long, totaling 55 minutes of driving. In Table 1 we provide a comparison regarding to some commonly used autonomous driving datasets. Figure 5 visualizes sample annotations of the dataset. The sequences consist of images captured by four different cameras: wide and narrow front cameras, as well as left and right cross-traffic cameras. All of the camera frames have been anonymized using the DachcamCleaner software tool. Each frame includes a JSON annotation file for the traffic light and traffic sign 3D bounding boxes, which provides geometric information along with the traffic light state and mask, traffic sign type, object occlusion, and the text on traffic signs (extracted using the Google Vision API). The data distribution across the ODDs is shown in Figure 10. The majority of the dataset consists of urban scenes, with approximately 320,000 auto-annotated traffic lights and 550,000 traffic signs. The per-frame annotation distribution is depicted in Figure 9.

4.1 Compute resources

We ran the developed algorithm on a computer having Intel Core i9-10900X CPU (3.70GHz × 20) processor, and 32 GiB RAM memory. The measured runtime is about 210-270 milliseconds for a single frame. This means that the runtime of producing our dataset (containing 220 sequences each having 227 frames) took around 2.9-3.7 hours. The mentioned per frame runtime is valid for all subsequent experiments described in 5.

5 Evaluation

5.1 Validation Challenges

Precise localization of traffic management objects on a large scale is extremely challenging due to issues such as sensor limitations described in Section 2. This challenge explains why there is still no publicly available dataset with long-range 3D annotations for traffic signs and traffic lights. Although Mapillary provides global latitude and longitude coordinates for traffic signs, the accuracy is low, and there is no information about the vertical position, extent, or orientation to accurately place these objects in the local coordinate system of a driving scene. Popular autonomous driving datasets like nuScenes, KITTI, and Waymo present additional challenges. Among these, only Waymo [19] provides 3D bounding boxes for traffic signs and has GNSS information for the camera frames, which is necessary to evaluate our algorithm on a dataset. However, it contains annotations up to only 77-78 meters from the observer, and there is no information about the relevance of the traffic sign to the ego vehicle, hence we cannot directly measure precision or recall, but only a distance error between the associated ground truth-prediction pairs. Moreover, we are unaware of publicly available traffic light

Figure 6: Qualitative comparison of Waymo ground truth (green) and auto-annotated (red) 3D bounding boxes (**Left**: segment-14811410906788672189_373_113_393_113_with_camera_labels; **Right**: segment-10203656353524179475_7625_000_7645_000_with_camera_labels). The annotated traffic sign types in the ground truth and in the automatic annotation can be very different.

Table 2: Quantitative evaluation results of our automatic annotation method for **all** traffic signs of the Waymo validation dataset.

Metric	Result
Localization error	0.32 ± 0.22 meters
Orientation error	12.31 ± 2.00 degrees

datasets with 3D annotations, especially those containing distant objects. Given these difficulties, we have decided to validate our algorithm not only on the Waymo traffic sign dataset but also using manually annotated in-house benchmark datasets.

5.2 Validation of the method on the Waymo dataset

We evaluated our proposed method on the validation set of Waymo. Since our algorithm relies on egomotion-based triangulation, we filtered out segments where the traveled distance was less than 3 meters. Hence, we ended up with a final validation set containing 189 segments (each containing ≈ 200 frames). For the comparison of all detected traffic signs with all Waymo ground-truth boxes, we omitted classification metrics such as precision/recall due to the different definitions of the classes between Waymo and Mask2Former which we used to determine the existence of traffic signs in images. Fig. 6 depicts an example of the class definition mismatch. Our algorithm provides 3D bounding boxes only for traffic signs detected by the Mask2Former model. All metrics were calculated within the range of [-10m, 10m] lateral and [0m, 80m] longitudinal positions of the instantaneous coordinate system. The association distance threshold was set to 1 meter. Altogether **45,257** Waymo ground truth boxes have been associated with the bounding boxes generated by our method. The absolute mean distance between the centers is $\mathbf{0.32 \pm 0.22}$ **meters** and the mean absolute difference in the orientation is $\mathbf{12.31 \pm 2.00}$ **degrees** (see metrics in Table 2). The error distributions are shown in Fig. 11, where the performance was evaluated in 4 m x 10 m blocks.

We also provide validation results with respect to a relevant subset of traffic signs where we manually selected speed limit and stop signs from the mentioned 189 segments. In case of four traffic signs we did not approach them closer than 40 meters during the segment, and therefore our algorithm could not provide reliable bounding box estimation. Ignoring these objects, we measured the recall, position, and orientation error on 66 physically different traffic signs. Together, **5,511** ground truth boxes have been associated with our detections, where we detected **93.76 %** of traffic signs. The absolute mean distance between the centers is $\mathbf{0.28 \pm 0.23}$ meters and the mean absolute difference in orientation is $\mathbf{8.78 \pm 3.37}$ **degrees** (see Table 3). Detailed metrics can be seen in Fig. 12. These results indicate that the performance of our algorithm is even better if we consider only the traffic signs that are critical for self-driving.

Table 3: Quantitative evaluation results of our automatic annotation method for **speed limit and stop signs** of the Waymo validation dataset.

Metric	Result
Recall	93.76 %
Localization error	0.28 $\pm$ 0.23 meters
Orientation error	8.78 $\pm$ 3.37 degrees

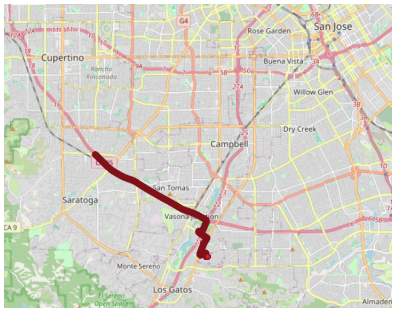

Figure 7: Visualization of the traffic sign validation route.

5.3 Validation of Automatic Traffic Sign Annotation on in-house dataset

We also validated the traffic sign automatic annotation performance on a 7-kilometer route in San José, California, which included both highway and urban sections (see the validation route in Figure 7). In total, 183 traffic signs were manually annotated with oriented 3D bounding boxes using LiDAR point cloud data. This manually created map was projected into the instantaneous coordinate systems of the vehicle, allowing for a detailed comparison with the automatic annotation. All metrics were calculated within the range of [-10m, 10m] lateral and [0m, 200m] longitudinal positions of the instantaneous coordinate system. The association distance threshold was set to 1 meter, and we calculated localization precision and recall related to the bounding box center. The automatic annotation method achieved **97.08%** precision and **95.33%** recall (see Table 4 for more detailed results). It is worth noting that the lower recall value resulted from only six missed traffic signs on the highway section, which included traffic signs with categories less relevant for self-driving (e.g. destination distance, interchange advance exit).

We also evaluated the localization errors of true positive detections using the absolute mean distance between the 3D bounding box centers and the annotations. Moreover, the absolute orientation error of the annotations is also evaluated. Our algorithm achieves low localization (**0.3 $\pm$ 0.16 meters**) and orientation (**11.09 $\pm$ 6.78 degrees**) errors that are similar to the values measured on the Waymo dataset. Detailed metrics are shown in Fig. 13 and Fig. 14.

5.4 Validation of Automatic Traffic Light Annotation on in-house dataset

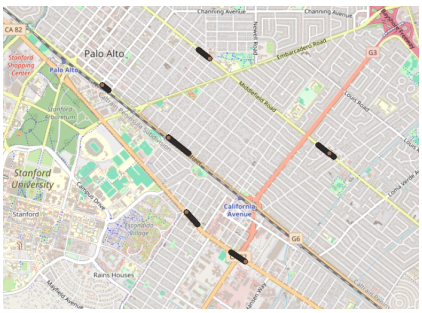

Figure 8: Visualization of the traffic light validation route.

Table 4: Quantitative evaluation results of our automatic annotation method for traffic signs on in-house dataset.

Metric	Result
Association precision	97.08 %
Association recall	95.33 %
Localization error	0.30 ± 0.16 meters
Orientation error	11.09 ± 6.78 degrees

Table 5: Quantitative evaluation results of our automatic annotation method for traffic lights on in-house dataset.

Metric	Result
Association precision	91.13 %
Association recall	95.87 %
Localization error	0.22 ± 0.20 meters
Orientation error	10.49 ± 9.39 degrees
Color state classification accuracy	94 %

We validated the automatic traffic light annotation algorithm at several intersections in Palo Alto, California. The validation route is approximately 1.3 kilometers long and includes 40 traffic lights (see the validation route in Figure 8). The 3D bounding boxes of the traffic lights, as well as their states, were manually annotated. Consequently, we measured both localization performance and traffic light state classification accuracy. In the association metrics, a true positive means the prediction is within 1 meter of the ground truth and the predicted class is correct. All metrics were calculated within the range of [-10m, 10m] lateral and [0m, 200m] longitudinal positions of the instantaneous coordinate system. Our method achieved **91.13%** precision and **95.87%** recall. The absolute localization error between the bounding box centers is **22 centimeters**, and the orientation absolute error is **10.49** $\pm$ **9.39 degrees**. The traffic light color state classification accuracy is **94%**. Detailed metrics are shown in Fig. 15 and Fig. 16.

6 Conclusion

Despite self-driving developments that have been conducted for several decades, there is still no publicly available large-scale dataset with 3D annotated traffic lights and traffic signs. This indicates that annotating traffic management objects is challenging, even with manual resources. This is especially true for traffic lights, which are difficult to detect in LiDAR point clouds even for humans, as their physical characteristics (e.g., small size, high placement, and black coating) make it challenging for the sensor to produce easily detectable reflections. In this work, we developed a fully automated method to generate temporally consistent 3D bounding boxes with high localization precision for traffic lights and traffic signs, which can be used to train image-based perception models for self-driving cars. Additionally, we released a public dataset generated by our algorithm, available under a CC BY-NC-SA 4.0 license, allowing the research community to use it for non-commercial research purposes[2].

Limitations The dataset is automatically annotated and, despite our extensive quality assurance process aimed at minimizing errors, it is still subject to annotation errors. Furthermore, the validation dataset size is limited which might hinder to measure the generalization ability of the proposed method.

Future work In the future, we aim to increase the manually annotated validation set's size continually. Furthermore, the traffic light detection precision shall be investigated on a larger sample.

[2]https://github.com/aimotive/aimotive_tl_ts_dataset

References

[1] Karsten Behrendt, Libor Novak, and Rami Botros. A deep learning approach to traffic lights: Detection, tracking, and classification. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1370–1377, 2017.

[2] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[3] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022.

[4] R. de Charette. Lara french traffic lights recognition (tlr) public benchmarks. In ,, 2015.

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

[6] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.

[7] Nathaniel Fairfield and Chris Urmson. Traffic light mapping and detection. In *2011 IEEE International Conference on Robotics and Automation*, pages 5421–5426, 2011.

[8] Felix Fent, Fabian Kuttenreich, Florian Ruch, Farija Rizwin, Stefan Juergens, Lorenz Lechermann, Christian Nissler, Andrea Perl, Ulrich Voll, Min Yan, et al. Man truckscenes: A multimodal dataset for autonomous trucking in diverse conditions. *arXiv preprint arXiv:2407.07462*, 2024.

[9] Andreas Fregin, Julian Muller, Ulrich Krebel, and Klaus Dietmayer. The driveu traffic light dataset: Introduction and comparison with existing datasets. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3376–3383, 2018.

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

[11] Farouk Ghallabi, Ghayath El-Haj-Shhade, Marie-Anne Mittet, and Fawzi Nashashibi. Lidar-based road signs detection for vehicle localization in an hd map. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 1484–1490, June 2019.

[12] Morten Bornø Jensen, Mark Philip Philipsen, Andreas Møgelmose, Thomas Baltzer Moeslund, and Mohan Manubhai Trivedi. Vision for looking at traffic lights: Issues, survey, and perspectives. *IEEE Transactions on Intelligent Transportation Systems*, 17(7):1800–1815, 2016.

[13] Tamas Matuszka, Ivan Barton, Ádám Butykai, Péter Hajas, Dávid Kiss, Domonkos Kovács, Sándor Kunsági-Máté, Péter Lengyel, Gábor Németh, Levente Pető, et al. aimotive dataset: A multimodal dataset for robust autonomous driving with long-range perception. In *International Conference on Learning Representations 2023 Workshop on Scene Representations for Autonomous Driving*.

[14] Simone Mentasti, Yusuf Can Simsek, and Matteo Matteucci. Traffic lights detection and tracking for hd map creation. *Frontiers in Robotics and AI*, 10, 2023.

[15] ABM Musa. Multi-view traffic sign localization with high absolute accuracy in real-time at the edge. In *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '22, New York, NY, USA, 2022. Association for Computing Machinery.

[16] Mario Soilán, Belén Riveiro, Joaquín Martínez-Sánchez, and Pedro Arias. Traffic sign detection in mls acquired point clouds for geometric and image-based semantic inventory. *ISPRS Journal of Photogrammetry and Remote Sensing*, 114:92–101, 2016.

[17] Wonho Song and Hyun Myung. 3d traffic sign detection using camera-lidar projection. In Ahmad Fakhri Ab. Nasir, Ahmad Najmuddin Ibrahim, Ismayuzri Ishak, Nafrizuan Mat Yahya, Muhammad Aizzat Zakaria, and Anwar P. P. Abdul Majeed, editors, *Recent Trends in Mechatronics Towards Industry 4.0*, pages 821–827, Singapore, 2022. Springer Singapore.

[18] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020.

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

A Appendix

A.1 Figures

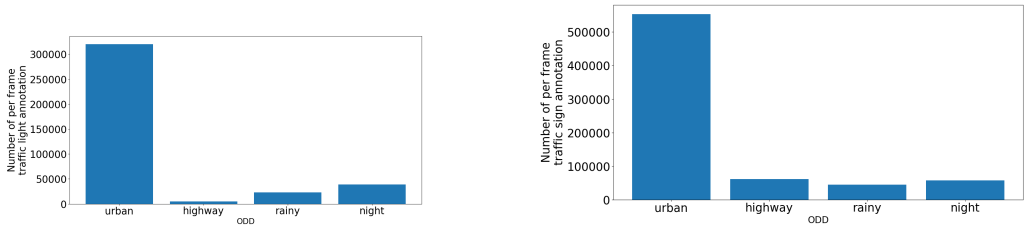

Figure 9: Data distribution of per frame annotations.

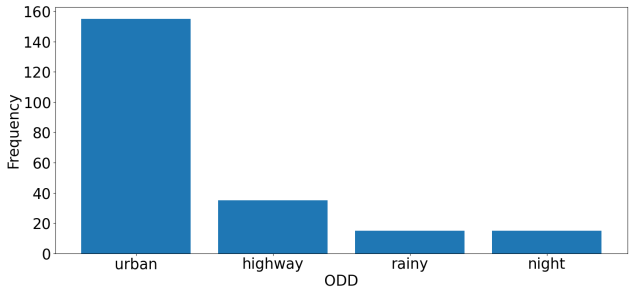

Figure 10: Data distribution across the different operational design domains.

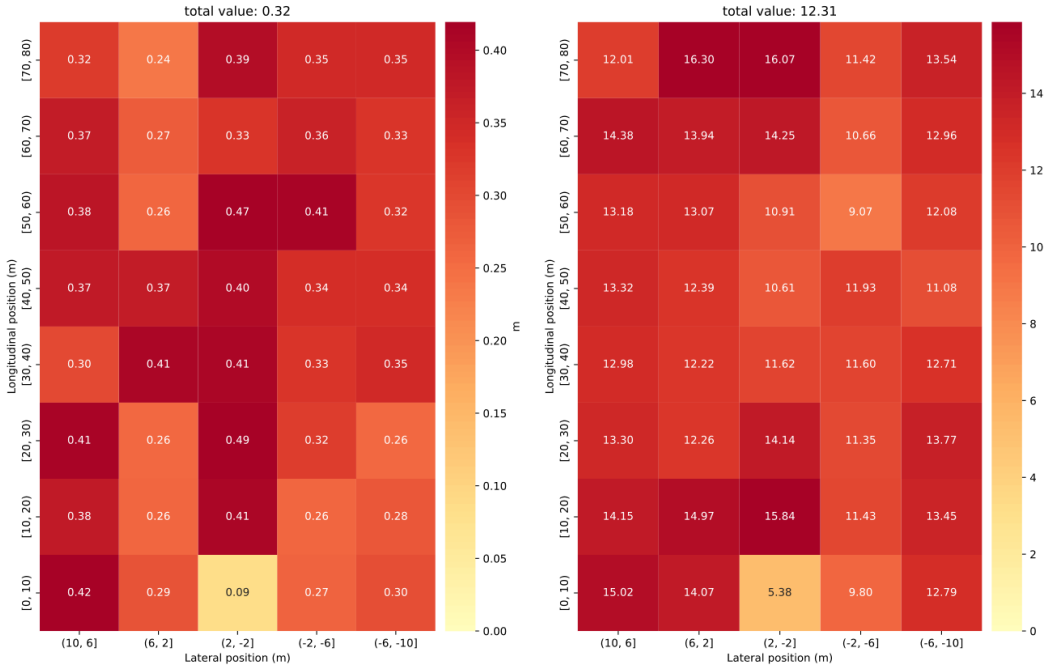

Figure 11: Evaluation results (in 4 m x 10 m blocks until 80 meters) of the proposed algorithm measured on **all** traffic sign boxes related to the Waymo validation set (**Left**: Mean error in bounding box center estimation (**0.32 meters**). **Right**: Mean absolute error in box orientation (**12.31 degrees**). **(best viewed by zooming in)**

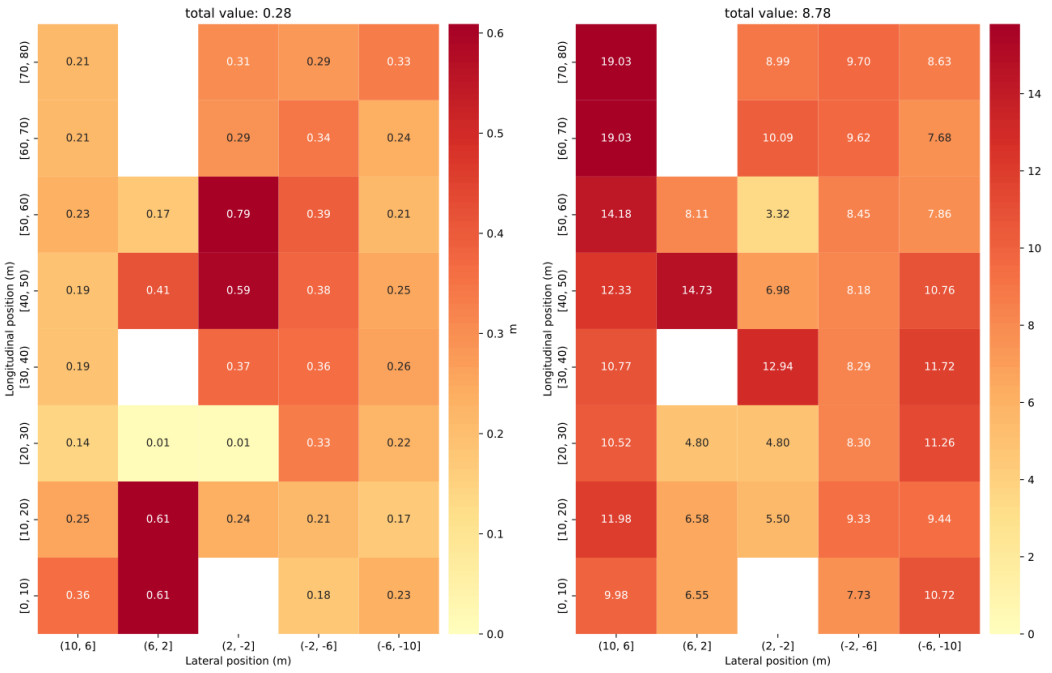

Figure 12: Evaluation results (in 4 m x 10 m blocks until 80 meters) of the proposed algorithm measured on the manually selected **speed limit and stop** signs related to the Waymo validation set. **Left**: Mean error in bounding box center estimation (**0.28 meters**). **Right**: Mean absolute error in box orientation (**8.78 degrees**).

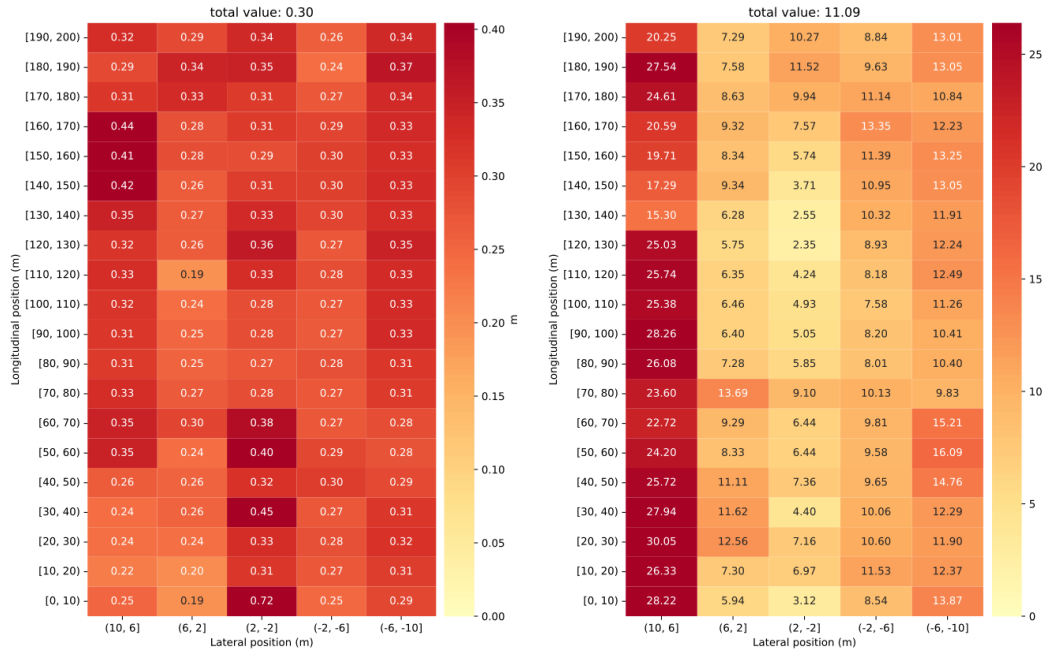

Figure 13: Evaluation results (in 4 m x 10 m blocks until 200 meters) of the proposed algorithm measured on our manually annotated **in-house** traffic sign dataset. **Left**: Mean error in bounding box center estimation (**0.3 meters**). **Right**: Mean absolute error in box orientation (**11.09 degrees**).

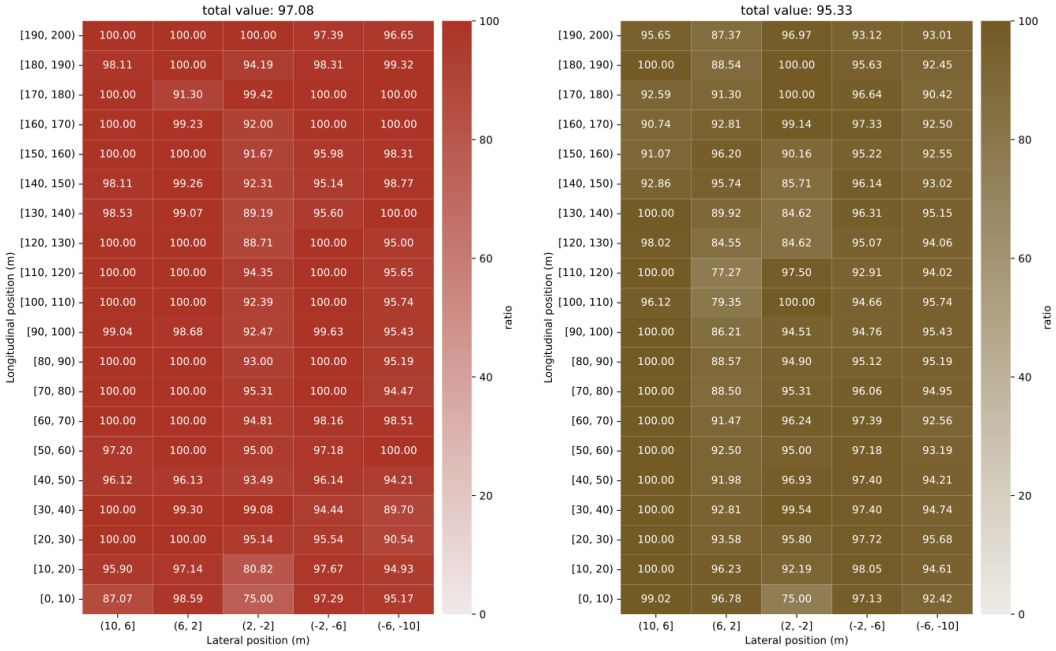

Figure 14: Precision and recall (in 4 m x 10 m blocks until 200 meters) of the proposed algorithm measured on our manually annotated **in-house** traffic sign dataset. **Left**: Precision (**97.08 %**). **Right**: Recall (**95.33 %**).

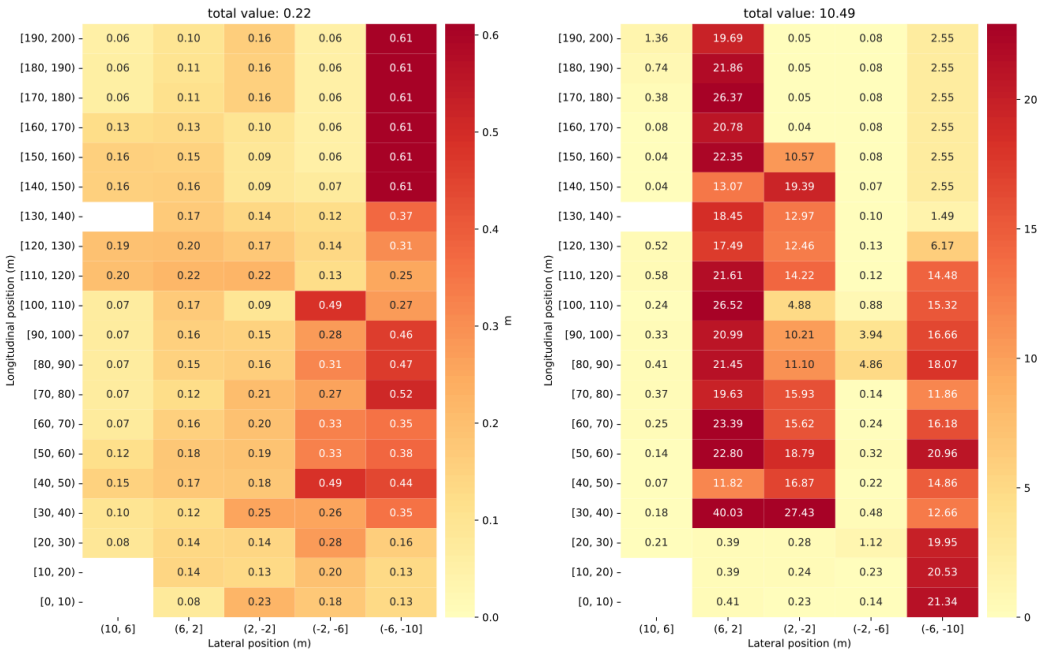

Figure 15: Evaluation results (in 4 m x 10 m blocks until 200 meters) of the proposed algorithm measured on our manually annotated **in-house** traffic light dataset. **Left**: Mean error in bounding box center estimation (**0.22 meters**). **Right**: Mean absolute error in box orientation (**10.49 degrees**).

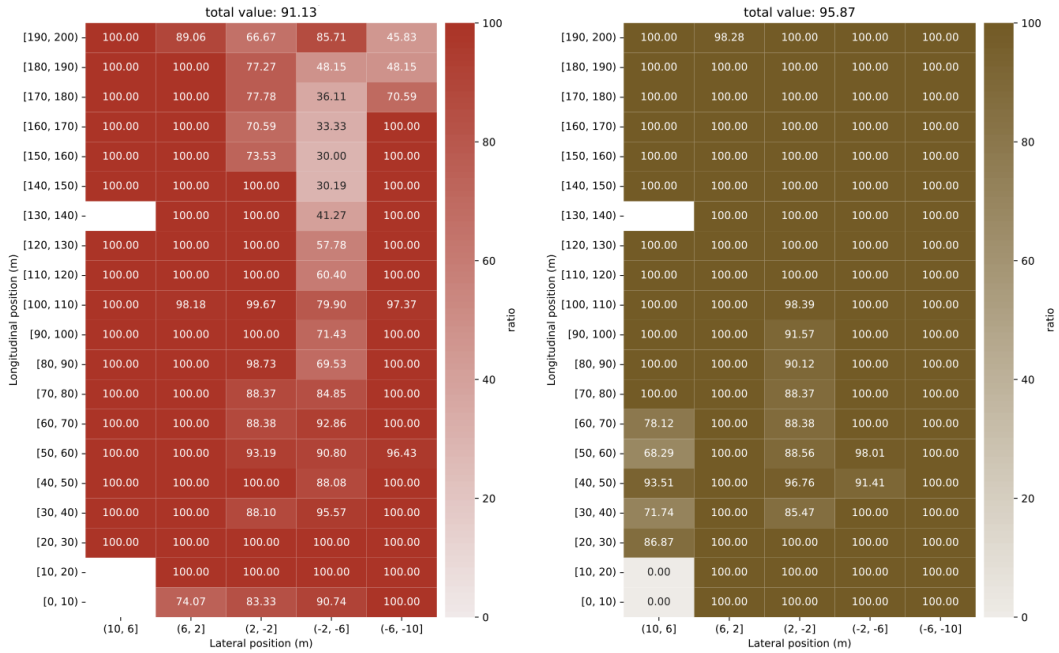

Figure 16: Precision and recall (in 4 m x 10 m blocks until 200 meters) of the proposed algorithm measured on our manually annotated **in-house** traffic light dataset. **Left**: Precision (**91.13 %**). **Right**: Recall (**95.87 %**).

A.2 aiMotive 3D Traffic Light and Traffic Sign Dataset Description

A.2.1 Dataset file structure description

In Fig. 17 you can see the file structure of the dataset. In the root directory the dataset is sorted into four different operational design domains (ODDs): highway, night, rainy and urban. In each ODD you can find sequence folders that contain 15 sec long records. Under each sequence folder there is a sensor directory containing the calibration, camera image and GNSS/INS data as well as the traffic_light and traffic_sign folders with the relevant 3D annotation data.

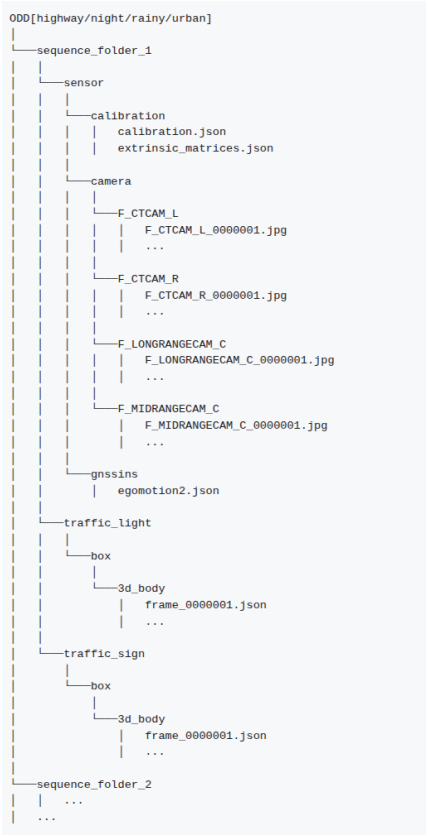

Figure 17: Description of aiMotive 3D Traffic Light and Traffic Sign Dataset file structure.

A.2.2 Sensor setup

The recording vehicle is equipped with four cameras, IMU and GPS. Details of the cameras are listed below:

- Sony IMX490 (front & cross-traffic cams)
- 30 to 40 Hz capture frequency
- 1/1.55” CMOS sensor of 2896 x 1876 resolution

Parameters of IMU and GPS:

- Novatel PwrPak7
- Up to 100 Hz measurement frequency
- Position accuracy of 100 mm (RTK)
- Heading accuracy of 0.08° (baseline = 2m)
- Roll & pitch accuracy of 0.02°

A.2.3 Coordinate systems

The reference coordinate system used for defining the annotated objects is called the body coordinate system. The body coordinate system is a coordinate system that is attached to the object holding the sensor system; for example, the vehicle body. The origin is the projected ground plane point under the center of the vehicle's rear axis at nominal vehicle body height and zero velocity. If looking towards the vehicle's forward direction from the driver's point of view, then:

- the X-axis points forward along the vehicle body,
- the Y-axis points left to the vehicle body,
- and the Z-axis points up along the vehicle body,
- where the measurement unit is defined in meters.

The body coordinate system is depicted in Fig. 18.

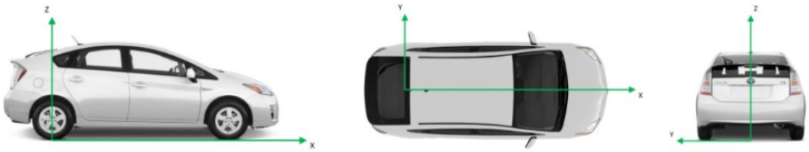

Figure 18: Illustration of the body coordinate system.

For cameras, the camera coordinate system can be used for projecting 3D points onto the camera image and vice versa. The origin is the camera's viewpoint and the axes are defined as follows:

- +X is right
- +Y is down
- +Z is forward (viewing into the scene),

where the measurement unit is defined in meters. The camera coordinate system is depicted in Fig 19.

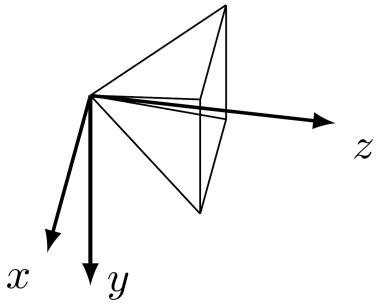

Figure 19: Illustration of the body coordinate system.

The sensor layout is illustrated in Fig. 20

A.2.4 Sensor synchronization

All of the recorded sensors are synchronized. The annotation files (named frame_#.json, where # refers to the camera frame identifier number) contain the traffic light and sign objects on the given camera frame. The camera sensors are rolling shutter-type sensors. This means that the exposure starts from the top of the sensor, going downwards, row by row.

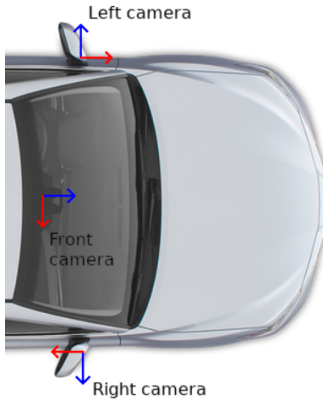

Figure 20: Illustration of the body coordinate system.

B NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes, we highlighted the advantages of our new method as well as the published dataset.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Yes, we discuss it in the Limitations section just after the conclusions.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We explained the steps of our proposed algorithm.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Yes, the dataset and code can be accessed here: https://github.com/aimotive/aimotive_tl_ts_dataset

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Yes, we made experiments on in-house and public benchmark datasets that can be found in Evaluation section.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Yes, we provided several error metrics of the method that can be found in the Evaluation section as well as in the appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Yes, we described the necessary compute resources in section 4.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Yes, it does.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Yes, they are.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Yes, we provided a detailed description of the published dataset and the used sensor setup in the appendix.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.