

An Infrastructure-based Localization Method for Articulated Vehicles

Alberto Justo^{1,3}, Iker Pacho^{1,2}, Javier Araluce¹, Jesus Murgoitio¹, and Luis Miguel Bergasa³

¹ TECNALIA, Basque Research and Technology Alliance (BRTA), 48160 Derio, Spain

² Department of Automatic Control and Systems Engineering, University of the Basque Country (UPV/EHU), 48013 Bilbao, Spain

³ Department of Electronics, University of Alcalá, 28805 Alcalá de Henares, Spain

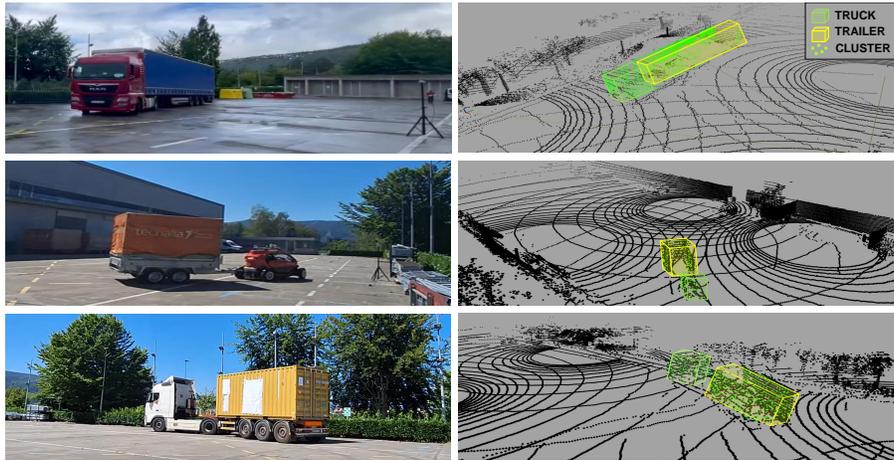


Fig. 1: Infrastructure-based Localization Method for Articulated Vehicles: The left side shows three different articulated vehicles evaluated in this research, while the right side displays LiDAR point clouds with predicted 3D Boxes for the same scenario.

Abstract. Automating articulated vehicles in valet parking maneuvers is becoming more significant, due to the growing need for efficient freight transportation and logistics. Thus, this paper introduces a novel Infrastructure to Vehicle (I2V) localization approach for articulated vehicles. The proposed method focuses on accurately classifying trucks and trailers and hitch angle estimation between them. Validations with real-world LiDAR data on different articulated vehicles show that our solution improves safety and efficiency in automated docking scenarios without vehicle modifications.

Keywords: Articulated Vehicles · Localization · Hitch Angle Estimation · Infrastructure to Vehicle

1 Introduction

The use of articulated vehicles, defined as a lead unit attached to a trailer, is a common practice in the transportation industry, particularly for the movement of goods and materials. Road vehicles in the European Union constitute the 25 % of the whole freight transportation, near 4 million vehicles [11]. Moreover, according to the Transport and Logistics Observatory (Observatorio de Transporte y Logística OTLE) in Spain, there are over 1.5 million articulated vehicles circulating each year [24]. In the context of Connected and Automated Vehicles (CAVs), particularly in road freight transportation systems, articulated vehicles present different challenges to be accomplished. Articulated vehicles are suited for applications like automated docking, where space optimization and precise control are crucial in urban environments and freight hubs [15]. The implementation of automated and connected technologies in load transport can lead to more efficient logistics, reduced emissions, and lower operational costs. Currently, companies like Aurora [2], Driveblocks [10] or Waabi [29] are integrating automated trucking technologies into different driving scenarios.

Given these circumstances, articulated vehicles experience more unstable situations than one unit vehicles. It is essential to consider modes such as jackknifing, trailer shaking or rollover, to ensure a safer control of the vehicle [19]. Due to different attachable trailers and the articulated system problem, there is a need to identify each unit’s location and orientation. Also, a fundamental requirement is to ensure the hitch angle of the vehicle, defined as the angle between the longitudinal axes of the trailer and the lead unit [15]. Current solutions use a variety of different sensors [9,16,23] and model-based methods [13,14] to solve these challenges. However, these approaches imply making significant modifications on the vehicle, which can be costly and time-consuming in terms of sensor installation and maintenance. Infrastructure-based solutions offer a more cost-effective and comprehensive alternative for controlled scenarios that work with different vehicles without adapting the vehicle [3]. As a result, infrastructure-based solutions enhance safety and efficiency, while avoiding the logistical challenges and expenses typically associated with vehicle-based systems.

For this reason, we propose in this paper our Infrastructure-to-Vehicle (I2V) localization solution for articulated vehicles, represented in Figure 3. To the best of our knowledge, this approach has not been implemented before. The main contributions of our paper are the following:

- We developed a novel LiDAR-based I2V localization approach for articulated vehicles, using KD-Tree Euclidean Clustering [20] and Principal Component Analysis (PCA) [22]. This system is integrated with Simple Online and Realtime Tracking (SORT) [5].
- We implemented a hitch angle estimation (HAE), based on truck and trailer detection and tracking within geometric calculation.
- We validated our approach with three articulated vehicles in real-world scenarios, within different LiDAR setups.

2 Related Works

2.1 Infrastructure-based Localization

Although 3D object detection has experienced rapid development, the use of roadside sensors for 3D object detection and tracking remains a relatively new area. This approach holds significant potential to overcome challenges such as occlusions, perception failures from vehicle-based sensors, and limited detection range [4]. The use of roadside sensors can lead to more accurate object detection, precise localization, and improved overall situational awareness [1, 8].

In addition to these advantages, pointcloud map-based solutions, like OctoMap [31] or Normal Distributions Transform (NDT [26]) are becoming increasingly popular to aid filtering processes. They provide structured representations of 3D space that can be used to easily rule out noise, static objects and outliers. However, the dynamic nature of articulated vehicle environments means that maps must be frequently updated to reflect changes. For example, the positions of vehicles, containers, and equipment can change rapidly in logistics yards or construction sites. This needs continuous map updates to maintain the accuracy and reliability of the localization and detection systems. These updates can be computationally demanding and require sophisticated algorithms to ensure that the map remains consistent and accurate over time [6].

Currently, datasets such as [32, 34, 35] are among the only real-world infrastructure datasets available. Moreover, few current infrastructure-based 3D object detection models can distinguish between truck and trailer in articulated vehicles [33]. As of today, there is no available dataset focused on 3D detection of different articulated vehicles.

2.2 Hitch Angle Estimation

Hitch Angle Estimation (HAE) is a critical component in the control and safety systems of articulated vehicles [15]. It refers to the determination of the angle between the towing vehicle and the trailer, as shown in Figure 2. HAE is essential for various applications, including stability control, path planning and maneuvering in constrained environments [15, 16]. Accurate HAE helps preventing accidents and ensures smooth vehicle operation, particularly in challenging scenarios like sharp turns or reversing. This is a critical point in automated docking for freight hubs, where accurate maneuvers are necessary due to limited space.

Vehicle-based hitch angle estimation methods utilize various sensors and models, each with distinct advantages and limitations. In terms of sensors, most common solutions [7, 27, 28] are based on GNSS-IMU sensors, widely used for their simplicity and real-time capabilities, but can suffer from drifts and external localization errors. In locations like freight hubs, container ports, and similar industrial environments, GNSS signals can be obstructed by large structures or metal containers, creating challenges for reliable localization. Here, infrastructure-based systems provide alternative localization methods, enhance

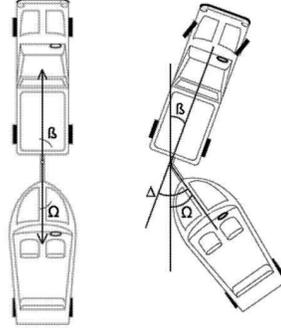


Fig. 2: Illustration of truck (β), trailer (α) and hitch (Δ) angles [28]

safety, and improve operational efficiency. By strategically placing fixed sensors in these environments, we can support logistics operations where traditional vehicle-based systems may fail. This holistic approach improves system robustness and leads to more advanced applications in autonomous driving and smart infrastructure management.

Camera-based systems [9] offer direct visual observation of the hitch angle, yet environmental factors and lighting variations can compromise their performance. The combination of LiDAR and RADAR solutions [21, 23] provide accurate measurements and are less affected by weather, but they are costly and require multiple modifications on the vehicle. Furthermore, these systems are more focused only on the HAE, but not on the localization of both parts of the articulated vehicle. On the other hand, our system provides both alternate solutions without need of vehicle modifications nor increasing its cost. Moreover, it gives the possibility to work with different vehicles. From models perspective, kinematic and dynamic systems are normally used. Kinematic models [16] estimate hitch angles based on the geometric properties of the vehicle. They are computationally efficient but may not fully capture the effects of dynamic factors, like tire slip or road surface variations. On the other hand, dynamic models [21] offer a comprehensive understanding by incorporating forces and moments. However, they demand significant computational resources and detailed vehicle parameter data. Infrastructure-based methods are not currently able to calculate vehicle dynamics, but there is still gap for collaborative solutions to be done. However, in situations like low-speed maneuvers at a proper frame rate, I2V solutions can compensate this shortcoming in some way. I2V systems may enhance these methods by providing additional perspectives and data. This external data can improve the accuracy and reliability of hitch angle estimation, offering redundancy and helping to overcome challenges like sensor drift or occlusions.

3 Methodology

In this paper, we present our infrastructure, LiDAR-based framework for localization of articulated vehicles, shown in Figure 3. Our method involves sensor calibration, point cloud preprocessing, vehicle detection, classification, tracking and hitch angle estimation.

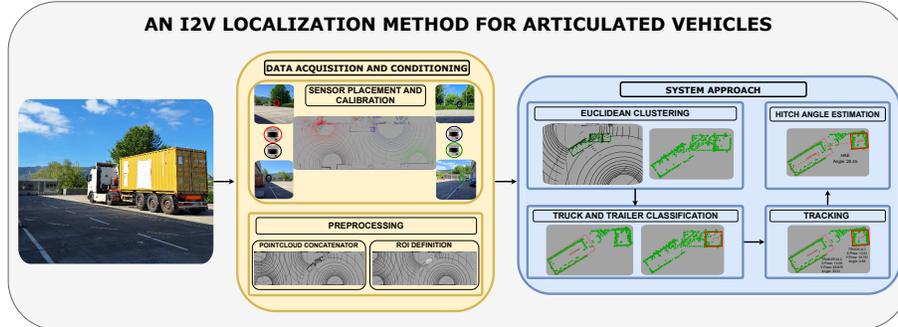


Fig. 3: I2V Localization Method for Articulated Vehicles in docking manouvers.

This pipeline can be modified for different types of articulated vehicles. All our framework modules support real-time parameter updates, allowing dynamic adjustments without restarting. This feature is facilitated through a parameter callback mechanism for each module. It updates internal state variables whenever any parameter is changed.

3.1 Sensor Calibration

We transform the point clouds from different sensors to a common coordinate frame, ensuring all data points are represented in the same spatial context. This involves applying rotation and translation operations. Later on, we use the Iterative Closest Point (ICP [30]) algorithm for accurate alignment. It minimizes the distance between point clouds by iteratively refining the transformation parameters (rotation and translation). The key steps include:

- **Initial Guess.** We apply an initial transformation to the source point cloud, done previously by measuring distances and rotations between LiDAR frames. Moreover, we fine-tune those measurements through our parameter callback mechanism.
- **Closest Point Search.** We find the closest point in the target cloud for each point in the source cloud.
- **Transformation Estimation.** We estimate the transformation that minimizes the distance between the matched points.
- **Transformation Update.** We update the transformation parameters and repeat the process until convergence.

3.2 Preprocessing

We differ two components in the preprocessing part: a pointcloud concatenator and a region definition module. Both are illustrated in Figure 3.

Pointcloud Concatenator. The pointcloud concatenator implements voxel grid definition, centroid aggregation, timestamp synchronization, and final concatenation to create a unified point cloud. This module is based on the pointcloud preprocessor from Autoware Universe [17], which uses the following steps:

- **Voxel Grid Definition:** Given a point cloud $P = \{\mathbf{p}_i = (x_i, y_i, z_i)\}$, the space is divided into a grid of 3D voxels with dimensions $(\Delta x, \Delta y, \Delta z)$. Each point \mathbf{p}_i is assigned to a voxel V_{ijk} like in Equation 1.

$$i = \left\lfloor \frac{x_i - x_{\min}}{\Delta x} \right\rfloor, \quad j = \left\lfloor \frac{y_i - y_{\min}}{\Delta y} \right\rfloor, \quad k = \left\lfloor \frac{z_i - z_{\min}}{\Delta z} \right\rfloor \quad (1)$$

- **Aggregation:** For each voxel V_{ijk} , the points p_i are aggregated by computing the centroid. This is shown in Equation 2.

$$\mathbf{C}_{ijk} = \left(\frac{1}{N} \sum_{p \in V_{ijk}} x_p, \frac{1}{N} \sum_{p \in V_{ijk}} y_p, \frac{1}{N} \sum_{p \in V_{ijk}} z_p \right) \quad (2)$$

N is the number of points in voxel V_{ijk} . Each calculated centroid is then registered for later synchronization and concatenation.

- **Synchronization.** We align point cloud data packets based on their timestamps to ensure that data from different sources or time frames correspond to the same moment in time. In cases where exact timestamp matches are not available, interpolation is used to estimate intermediate data points, as shown in Equation 3.

$$\mathbf{P}_{int} = \mathbf{P}_i + \frac{(t - t_i)}{(t_{i+1} - t_i)} (\mathbf{P}_{i+1} - \mathbf{P}_i) \quad (3)$$

For points p_i and p_{i+1} with timestamps t_i and t_{i+1} , an interpolated point p_{int} at time t is calculated. In cases where point cloud data cannot be sufficiently synchronized, such as when timestamps fall outside a predefined tolerance window, the data is discarded. This prevents inconsistencies in the merged point cloud and ensures accurate environmental representation.

- **Concatenation.** We combine the synchronized and aligned point clouds into a single unified point cloud. Since there is a common reference frame already for all point clouds, it makes the concatenation easier.

Region Definition Module. This module acquires the merged point cloud data with two clear goals: define a region of interest (ROI) and identify and remove ground points. For that purpose, we implement passthrough and RANSAC

[12] filters. Passthrough filter acts as a spatial filter, allowing us to specify the bounds of the region we are interested in by setting minimum and maximum values for each dimension (x, y, z). RANSAC helps us to fit models to data with a high degree of outliers, in this case, ground points. Since our test field has very little slope, it significantly enhances the performance of the RANSAC algorithm. By applying these filters after concatenation, we avoid inconsistencies that could happen from processing separate pointclouds independently.

3.3 Euclidean Clustering Detection

This module, based on euclidean distance, is suitable to find cubic volumes. Given a concatenated point cloud $\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n\}$, we employ a KD-Tree algorithm to organize the points for rapid spatial queries. We segment the pointcloud into clusters based on a predefined distance tolerance ϵ , as shown in Equation 4. Here, P_i is the cluster of points that achieve the euclidean distance condition under tolerance ϵ .

$$P_i = \{\mathbf{p}_j \in \mathbf{P} \mid \text{dist}(\mathbf{p}_j, \mathbf{p}_k) < \epsilon\} \quad (4)$$

Moreover, we make an assumption: articulated vehicle dimensions are predefined. During the extraction process, each candidate cluster is evaluated against the predefined dimensional constraint. It allows rapid modifications, as the articulated vehicle dimensional parameters can be modified in real-time as needed. This dimensional constraint includes a tolerance, referred to as volume tolerance δV .

To implement the volume tolerance, we compute the volume of each detected cluster and compare it against the assumed volume V_e with the tolerance margin. A cluster is considered valid if its volume V_i satisfies Equation 5.

$$V_{\min} \leq V_i \leq V_{\max} \quad (5)$$

$V_{\max} = V_e + \delta V$ is the maximum acceptable volume, $V_{\min} = V_e - \delta V$ is the minimum acceptable volume. This ensures that only clusters with a realistic volume, accounting for both upper and lower bounds, are retained as an articulated vehicle candidate.

3.4 Truck and Trailer Classification

Once the whole articulated vehicle is detected, we make a second Euclidean Clustering detection for the trailer side, in a similar way to the aforementioned in Equations 4 and 5. Then, in order to detect the truck, we subtract vehicle and trailer pointclouds. We chose to apply this approach since trailer detection is more feasible due to bigger number of contained points. The distinction between trailer and truck can be achieved in a more intuitive way with this approach, as volume differences between both parts are notable. For each cluster P_i , its centroid \mathbf{C} is computed as shown in Equation 6:

$$\mathbf{C} = \frac{1}{n} \sum_{i=1}^n \mathbf{p}_i \quad (6)$$

Since centroid calculation may be biased in situations where not all planes of the whole vehicle are detected, we apply a correction in both x and y positions. It is determined by the distance between the centroid, the closest plane in x and y separately and the known dimensions of the vehicle.

$$\mathbf{C}_{\text{adjusted}} = \mathbf{C} + \left(\frac{D_x - d_x}{2}, \frac{D_y - d_y}{2}, 0 \right) \quad (7)$$

Here, D_x and D_y are the known dimensions of the vehicle in the x and y directions, respectively, and d_x and d_y are the distances from the centroid to the closest planes in the x and y directions, respectively. Thus, we can determine truck and trailer position with more accuracy, regardless of the amount of occluded planes.

3.5 Tracking

In order to identify and smooth our articulated vehicle detections, we use SORT (Simple Online and Realtime Tracking [5]) algorithm. It is designed for real-time applications, particularly in the context of multi-object tracking (MOT). SORT associates detections across consecutive frames using a combination of well-established techniques, such as the Kalman Filter and the Hungarian algorithm [18]. We use the Hungarian algorithm to solve the assignment problem by computing a cost matrix based on Euclidean distance between current detections and past trackers. Associations are determined using a 1 meter threshold to estimate truck and trailer IDs. The Kalman Filter in SORT is used to estimate the state of tracked objects, incorporating both prediction and measurement updates. The state prediction is given by Equation 8.

$$\mathbf{x}_{k|k-1} = \mathbf{F}\mathbf{x}_{k-1|k-1} \quad (8)$$

$\mathbf{x}_{k|k-1}$ is the predicted state vector at time k , \mathbf{F} is the state transition matrix and $\mathbf{x}_{k-1|k-1}$ is the previous state estimate. The correction step updates the state estimate based on new measurements, shown in Equation 9.

$$\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\mathbf{x}_{k|k-1}) \quad (9)$$

$\mathbf{x}_{k|k}$ is the updated state estimate, \mathbf{K}_k is the Kalman gain, \mathbf{z}_k is the measurement vector, and \mathbf{H} is the observation model matrix. The Kalman gain \mathbf{K}_k is computed as shown in Equation 10.

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}^\top(\mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^\top + \mathbf{R})^{-1} \quad (10)$$

$\mathbf{P}_{k|k-1}$ is the predicted covariance matrix, and \mathbf{R} is the measurement noise covariance matrix.

To sum up, SORT algorithm integration ensures robust tracking by effectively managing detection-to-track associations and updating state estimates.

3.6 Hitch Angle Estimation

Our pipeline for angle determination consists of Principal Component Analysis (PCA) for initial estimation of each cluster, SORT for smoothing, and HAE for the relative angle between the truck and trailer. After adjusting the centroid of the detected cluster, we apply Principal Component Analysis (PCA) using a history of the previous centroid positions. This helps us estimate the main direction in which the articulated vehicle is moving. To do this, we construct a matrix \mathbf{C} , where each row represents a historical centroid position $\mathbf{c}_i = (x_i, y_i, z_i)$. Next, we calculate the covariance matrix \mathbf{Q} of \mathbf{C} as mentioned in Equation 11. $\bar{\mathbf{c}}$ is the mean of the centroid positions, shown in Equation 12.

$$\mathbf{Q} = \frac{1}{n-1} \sum_{i=1}^n (\mathbf{c}_i - \bar{\mathbf{c}})(\mathbf{c}_i - \bar{\mathbf{c}})^\top \quad (11)$$

$$\bar{\mathbf{c}} = \frac{1}{n} \sum_{i=1}^n \mathbf{c}_i \quad (12)$$

PCA finds the eigenvectors and eigenvalues of \mathbf{Q} , where the eigenvector corresponding to the largest eigenvalue represents the principal direction of movement of the vehicle. This principal direction is projected onto the horizontal plane to find the yaw angle. To ensure stability in the orientation estimation, constraints are applied to limit rapid changes in the direction. Specifically, we constrain the change in orientation between consecutive frames to avoid unrealistic jumps. After estimating the orientation using PCA, we use the SORT algorithm just to smooth the yaw angle. This step reduces noise and fluctuations in the orientation estimation. The hitch angle Δ between the truck and trailer is determined by the difference in their respective yaw angles, as shown in Equation 13.

$$\Delta = \beta - \alpha \quad (13)$$

α represents the yaw angle of the truck, and β represents the yaw angle of the trailer [28].

4 Evaluation

4.1 Experimental Setup

The experiments were conducted in the test field at Astondo Bidea in Derio, Spain, shown in Figure 4. To achieve comprehensive coverage of the articulated vehicles and minimize occlusions, four Ouster OS1-32 LiDAR sensors were placed around the test field. The goal was to maximize the field of view and ensure accurate data capture from multiple angles.

We evaluated our method using three different articulated vehicles, as detailed in Table 1. Regarding the Renault Twizy, we modified its back structure in order to attach multiple kinds of trailers. It provided us an initial articulated

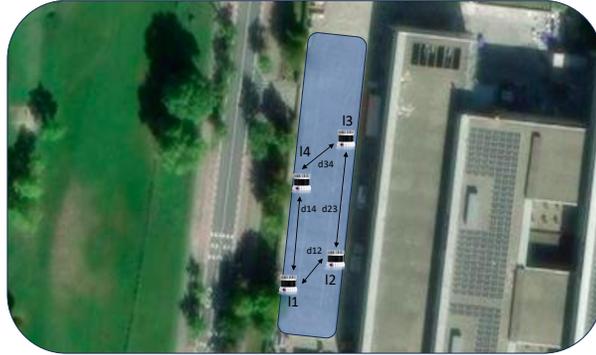


Fig. 4: Our test field for a realistic setting of automated docking. Here, Ouster LiDARs l1, l2, l3 and l4 are deployed with the following counterclockwise setup: $d_{12} = 19.2\text{m}$, $d_{23} = 40.5\text{m}$, $d_{34} = 19.2\text{m}$, $d_{14} = 15\text{m}$.

vehicle where to make our developments and tests. However, the main purpose of our application is to be applied in bigger articulated vehicles. The other two freight vehicles, Man TGX and Volvo FH, were rented for these experiments. We hired professional drivers for multiple day recording and testing. During the first 5 minutes before testing, there was an initial manual calibration of the whole pipeline for each vehicle. Then, we recorded multiple maneuvers through the test field, typical in freight hub situations. The recorded frames were manually labeled for a posterior offline evaluation with the calibrated system. For labeling, we used LabelCloud [25]. Given the project requirements where this research has been developed, our solution aims to be deployed in computational lightweight devices. The computational backbone of our experiments features a i7-12850HX with 32 GB of RAM.

Table 1: Dimensions of the different articulated vehicles used in the experiments.

| Vehicle Type | Component | Height [m] | Width [m] | Length [m] | Volume [m^3] |
|---------------|-----------|------------|-----------|------------|------------------|
| Renault Twizy | Truck | 1.50 | 1.57 | 2.00 | 4.71 |
| | Trailer | 2.50 | 1.75 | 2.50 | 10.94 |
| Man TGX | Truck | 2.86 | 2.77 | 2.77 | 21.94 |
| | Trailer | 2.55 | 2.77 | 13.68 | 96.63 |
| Volvo FH | Truck | 2.50 | 2.50 | 2.55 | 15.94 |
| | Trailer | 2.50 | 2.50 | 6.30 | 39.38 |

4.2 Results

In this subsection we present our results based on the Root Mean Square Errors (RMSE) for position/angle of each unit, and for HAE. Moreover, in relation with these metrics, we analyze our method performance with different LiDAR utilization in the same sensor placement, illustrated in Figure 4.

Position RMSE is obtained as shown in Equation 14. It represents the differences between the detected positions (x_i, y_i) and the ground truth positions $(x_i^{\text{gt}}, y_i^{\text{gt}})$ through the number of data points N .

$$\text{RMSE}_{\text{position}} = \sqrt{\frac{1}{N} \sum_{i=1}^N [(x_i - x_i^{\text{gt}})^2 + (y_i - y_i^{\text{gt}})^2]} \quad (14)$$

Orientation RMSE is obtained using the differences between the detected yaw angles θ_i and the ground truth yaw angles θ_i^{gt} , as in Equation 15.

$$\text{RMSE}_{\text{orientation}} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\theta_i - \theta_i^{\text{gt}})^2} \quad (15)$$

HAE RMSE is calculated in Equation 16. The HAE error at instance i is defined as the difference between the truck angle α_i and the trailer angle β_i of the detections against their ground truth values.

$$\text{RMSE}_{\text{HAE}} = \sqrt{\frac{1}{N} \sum_{i=1}^N ((\beta_i - \alpha_i) - (\beta_i^{\text{gt}} - \alpha_i^{\text{gt}}))^2} \quad (16)$$

We evaluated the performance of our method within these metrics as shown in Table 2, including the inference time of the entire framework: preprocessing + euclidean clustering + truck and trailer classification + tracking + HAE. From the obtained results, we can state that the Truck class shows higher overall error due to lower density of scattered points. This is particularly pronounced in the case of the Renault Twizy, due to its irregular shape compared to the more 'box-like' shapes of the other two vehicles. The Volvo FH presents the best results, as the space between Truck and Trailer is more notable than in the Man TGX, facilitating their classification. The aforementioned results can be seen qualitatively in Figure 5. Here, we choose to represent only the vehicles holding the higher number of frames (Man TGX and Volvo FH), since they are the main target of our research.

In order to analyze how LiDAR usage affects our method, we obtained different results as shown in Tables 3, 4 and 5. We can see through these tables that LiDAR usage is particularly relevant to HAE. This measurement accumulates the most significant error compared to the positions and orientations of both Truck and Trailer classes, making it the most representative metric for evaluation. The data shows a clear trend toward the use of multiple LiDAR sensors to minimize errors. The inclusion of 14 generally provides the lowest error rates

Table 2: Overall Metrics of our I2V Localization Method for Articulated Vehicles

| Platform | Frames | RMSE (Pos [m]/Orient [deg]) | | RMSE (HAE [deg]) | Inference time [ms] |
|---------------|--------|-----------------------------|--------------------|------------------|---------------------|
| | | Truck | Trailer | | |
| Renault Twizy | 518 | 0.19 / 1.68 | 0.16 / 1.65 | 1.96 | 36 |
| Man TGX | 906 | 0.15 / 1.57 | 0.13 / 1.46 | 1.82 | 45 |
| Volvo FH | 713 | 0.14 / 1.30 | 0.12 / 1.26 | 1.77 | 39 |

in the method, except in the case of the Volvo FH, where l3 proves more accurate. When using a combination of two LiDARs, the l2-l4 pairing offers the best performance, as their strategic placement effectively covers the articulated vehicle’s operating field while minimizing occlusions. Adding a third LiDAR, specifically the l2-l3-l4 combination, further lowers the RMSE. In all scenarios, incorporating all four LiDARs holds the most accurate HAE. However, beyond a certain point, the incremental benefits of additional LiDARs may vary, needing a balanced consideration between complexity, accuracy and computational cost.

Table 3: LiDAR setup metrics for Renault Twizy. The "-" implies that no LiDAR was used, whereas the "✓" means a LiDAR was used.

| LiDAR Setup | | | | RMSE (Pos [m]/Orient [deg]) | | RMSE HAE [deg] | Inference time [ms] |
|-------------|----|----|----|-----------------------------|--------------------|----------------|---------------------|
| l1 | l2 | l3 | l4 | Truck | Trailer | | |
| ✓ | - | - | - | 0.46 / 2.75 | 0.31 / 2.42 | 2.91 | 20 |
| - | ✓ | - | - | 0.33 / 2.38 | 0.31 / 2.53 | 2.95 | 20 |
| - | - | ✓ | - | 0.41 / 2.54 | 0.33 / 2.56 | 2.97 | 20 |
| - | - | - | ✓ | 0.31 / 2.64 | 0.34 / 2.45 | 2.80 | 20 |
| ✓ | ✓ | - | - | 0.22 / 2.10 | 0.33 / 2.36 | 2.70 | 25 |
| ✓ | - | ✓ | - | 0.36 / 2.16 | 0.30 / 2.55 | 2.70 | 25 |
| ✓ | - | - | ✓ | 0.29 / 2.11 | 0.32 / 2.79 | 2.69 | 25 |
| - | ✓ | ✓ | - | 0.30 / 2.54 | 0.21 / 2.11 | 2.80 | 25 |
| - | ✓ | - | ✓ | 0.26 / 2.32 | 0.28 / 2.09 | 2.66 | 25 |
| - | - | ✓ | ✓ | 0.26 / 2.35 | 0.27 / 2.27 | 2.84 | 25 |
| ✓ | ✓ | ✓ | - | 0.22 / 1.90 | 0.28 / 2.52 | 2.60 | 30 |
| ✓ | ✓ | - | ✓ | 0.19 / 2.14 | 0.18 / 2.10 | 2.54 | 30 |
| ✓ | - | ✓ | ✓ | 0.31 / 2.27 | 0.23 / 2.19 | 2.50 | 30 |
| - | ✓ | ✓ | ✓ | 0.19 / 2.00 | 0.22 / 2.22 | 2.26 | 30 |
| ✓ | ✓ | ✓ | ✓ | 0.19 / 1.68 | 0.16 / 1.65 | 1.96 | 36 |

5 Conclusions and Future Works

This work has presented our novel I2V Localization method for articulated vehicles. For this purpose, firstly we developed a LiDAR-based euclidean clustering

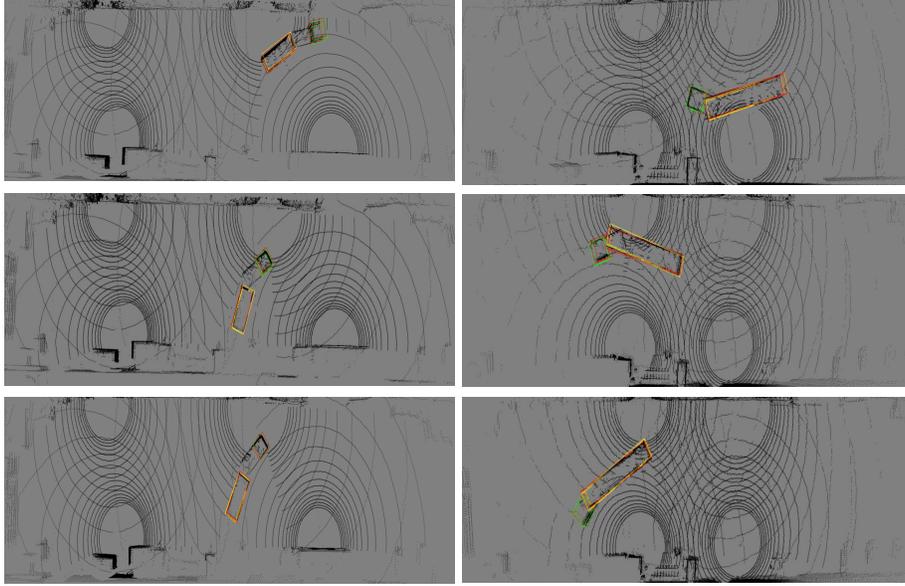


Fig. 5: Qualitative results of different trajectory captures for the Volvo (left) and Man (right) freight vehicles. **Ground Truth** boxes are represented in red. **Trailer** and **Truck** detections are represented in yellow and green, respectively.

Table 4: LiDAR setup metrics for Man TGX

| LiDAR Setup | | | | RMSE (Pos [m]/Orient [deg]) | RMSE | Inference | |
|-------------|----|----|----|-----------------------------|--------------------|-------------|-----------|
| 11 | 12 | 13 | 14 | Truck | Trailer | (HAE [deg]) | time [ms] |
| ✓ | - | - | - | 0.37 / 2.82 | 0.44 / 2.35 | 2.92 | 24 |
| - | ✓ | - | - | 0.47 / 2.55 | 0.34 / 2.72 | 2.89 | 24 |
| - | - | ✓ | - | 0.28 / 2.51 | 0.47 / 2.50 | 2.88 | 24 |
| - | - | - | ✓ | 0.30 / 2.62 | 0.48 / 2.50 | 2.87 | 24 |
| ✓ | ✓ | - | - | 0.31 / 2.05 | 0.26 / 2.30 | 2.73 | 30 |
| ✓ | - | ✓ | - | 0.19 / 2.12 | 0.31 / 2.45 | 2.80 | 30 |
| ✓ | - | - | ✓ | 0.25 / 2.09 | 0.34 / 2.70 | 2.71 | 30 |
| - | ✓ | ✓ | - | 0.29 / 2.60 | 0.23 / 2.25 | 2.80 | 30 |
| - | ✓ | - | ✓ | 0.35 / 2.24 | 0.24 / 2.13 | 2.68 | 30 |
| - | - | ✓ | ✓ | 0.32 / 2.45 | 0.20 / 2.14 | 2.80 | 30 |
| ✓ | ✓ | ✓ | - | 0.27 / 2.05 | 0.27 / 2.64 | 2.57 | 37 |
| ✓ | ✓ | - | ✓ | 0.27 / 2.13 | 0.18 / 1.99 | 2.61 | 37 |
| ✓ | - | ✓ | ✓ | 0.28 / 2.22 | 0.20 / 2.17 | 2.54 | 37 |
| - | ✓ | ✓ | ✓ | 0.21 / 2.12 | 0.13 / 2.26 | 2.20 | 37 |
| ✓ | ✓ | ✓ | ✓ | 0.15 / 1.57 | 0.13 / 1.46 | 1.82 | 45 |

Table 5: LiDAR setup metrics for Volvo FH

| LiDAR Setup | | | | RMSE (Pos [m]/Orient [deg]) | | RMSE | Inference |
|-------------|----|----|----|-----------------------------|--------------------|-------------|-----------|
| l1 | l2 | l3 | l4 | Truck | Trailer | (HAE[deg]) | time [ms] |
| ✓ | - | - | - | 0.32 / 2.75 | 0.49 / 2.31 | 2.87 | 25 |
| - | ✓ | - | - | 0.42 / 2.50 | 0.41 / 2.77 | 2.95 | 25 |
| - | - | ✓ | - | 0.31 / 2.46 | 0.53 / 2.57 | 2.84 | 25 |
| - | - | - | ✓ | 0.23 / 2.57 | 0.44 / 2.46 | 2.91 | 25 |
| ✓ | ✓ | - | - | 0.24 / 2.01 | 0.19 / 2.26 | 2.67 | 31 |
| ✓ | - | ✓ | - | 0.26 / 2.06 | 0.24 / 2.39 | 2.86 | 31 |
| ✓ | - | - | ✓ | 0.20 / 2.04 | 0.27 / 2.64 | 2.64 | 31 |
| - | ✓ | ✓ | - | 0.36 / 2.53 | 0.28 / 2.18 | 2.74 | 31 |
| - | ✓ | - | ✓ | 0.31 / 2.31 | 0.17 / 2.09 | 2.62 | 31 |
| - | - | ✓ | ✓ | 0.27 / 2.39 | 0.16 / 2.19 | 2.75 | 31 |
| ✓ | ✓ | ✓ | - | 0.34 / 2.10 | 0.20 / 2.58 | 2.53 | 35 |
| ✓ | ✓ | - | ✓ | 0.34 / 2.20 | 0.25 / 2.01 | 2.56 | 35 |
| ✓ | - | ✓ | ✓ | 0.22 / 2.27 | 0.22 / 2.23 | 2.48 | 35 |
| - | ✓ | ✓ | ✓ | 0.28 / 1.99 | 0.20 / 2.00 | 2.27 | 35 |
| ✓ | ✓ | ✓ | ✓ | 0.14 / 1.30 | 0.12 / 1.26 | 1.77 | 39 |

detection of freight vehicles. Then, we classified truck and trailer parts, tracking them through SORT algorithm. For angle estimation of each unit, we used PCA and SORT for further smoothing. Eventually, we estimated the hitch angle between both units, which is the most important focus of our research. This statement is proven by our evaluations obtained from real-world scenarios. Our approach proves to be precise and efficient, even without embedding additional sensors into the vehicle, making it a practical and lightweight solution. Moreover, it has been deployed in a real world logistic application. On the other hand, we need to implement an onboard sensing baseline where to evaluate our solution even further. However, compared to onboard solutions mentioned in the state of the art, ours still has some improvement, especially to lower HAE. The fusion of different inputs, such as data from both the infrastructure and the ego-vehicle, could be investigated to improve the overall localization and HAE performance. Furthermore, we would like to integrate deep learning methods to refine the detection and classification processes. Finally, communication systems impact on the performance of the proposed method could provide valuable insights for its application in more complex and dynamic environments.

6 Acknowledgements

This research was part of the CLEANPORTS 5.0 project (grant No 00154791) and the MEDUSA program (grant CER-2023101), funded by MICIN through CDTI under the EU’s MRR. We thank Ruben Rodriguez and Joan Albesa (ID-NEO Technologies S.A.U) and Juan Carlos de Pablo and Angel Martin (FM Logistics Iberica S.L) for their valuable support.

References

1. Arnold, E., Dianati, M., de Temple, R., Fallah, S.: Cooperative perception for 3d object detection in driving scenarios using infrastructure sensors. *IEEE Transactions on Intelligent Transportation Systems* **23**(3), 1852–1864 (Mar 2022). <https://doi.org/10.1109/tits.2020.3028424>, <http://dx.doi.org/10.1109/TITS.2020.3028424> 3
2. Aurora: Aurora tech (2024), <https://aurora.tech/>, accessed: 2024-08-08 2
3. Bai, Z., Wu, G., Qi, X., Liu, Y., Oguchi, K., Barth, M.J.: Infrastructure-based object detection and tracking for cooperative driving automation: A survey. *CoRR abs/2201.11871* (2022), <https://arxiv.org/abs/2201.11871> 2
4. Bai, Z., Wu, G., Qi, X., Liu, Y., Oguchi, K., Barth, M.J.: Infrastructure-based object detection and tracking for cooperative driving automation: A survey (2022), <https://arxiv.org/abs/2201.11871> 3
5. Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B.: Simple online and realtime tracking. In: 2016 IEEE international conference on image processing (ICIP). pp. 3464–3468. IEEE (2016) 2, 8
6. de Borba, T., Vaculin, O., Marzbani, H., Jazar, R.: Increasing safety of automated driving by infrastructure-based sensors. *IEEE Access* **PP**, 1–1 (01 2023). <https://doi.org/10.1109/ACCESS.2023.3311136> 3
7. Chen, Q., Zhang, Q., Niu, X.: Estimate the pitch and heading mounting angles of the imu for land vehicular gnss/ins integrated system. *IEEE Transactions on Intelligent Transportation Systems* **PP**, 1–13 (05 2020). <https://doi.org/10.1109/TITS.2020.2993052> 3
8. Cui, Y., Ge, S.: Autonomous vehicle positioning with gps in urban canyon environments. vol. 2, pp. 1105 – 1110 vol.2 (02 2001). <https://doi.org/10.1109/ROBOT.2001.932759> 3
9. Dahal, A., Hossen, J., Chennupati, S., Sistu, G., Malhan, K., Amasha, M., Yogamani, S.: Deeptrailerassist: Deep learning based trailer detection, tracking and articulation angle estimation on automotive rear-view camera. pp. 2339–2346 (10 2019). <https://doi.org/10.1109/ICCVW.2019.00287> 2, 4
10. DriveBlocks: Driveblocks (2024), <https://www.driveblocks.ai/>, accessed: 2024-08-08 2
11. EuroStat: Freight transport statistics - modal split. aa (2022) 2
12. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **24**(6), 381–395 (jun 1981). <https://doi.org/10.1145/358669.358692>, <https://doi.org/10.1145/358669.358692> 7
13. Fuchs, C., Knopp, B., Zöbel, D., Paulus, D.: Model-based evaluation of practical sensor noise impacts in articulated vehicle driving scenarios. In: 2017 IEEE Intelligent Vehicles Symposium (IV). pp. 577–582 (2017). <https://doi.org/10.1109/IVS.2017.7995780> 2
14. Fuchs, C., Neuhaus, F., Paulus, D.: 3d pose estimation for articulated vehicles using kalman-filter based tracking. *Pattern Recognition and Image Analysis* **26**, 109–113 (07 2016). <https://doi.org/10.1134/S1054661816010077> 2
15. Habibnejad Korayem, A., Khajepour, A., Fidan, B.: A review on vehicle-trailer state and parameter estimation. *IEEE Transactions on Intelligent Transportation Systems* **23**(7), 5993–6010 (2022). <https://doi.org/10.1109/TITS.2021.3074457> 2, 3

16. Habibnejad Korayem, A., Pazooki, A., Durali, L., Khajepour, A., Fidan, B., Ponnuswami, A.V., Khaligh, S.P.: Hitch angle estimation of a towing vehicle with arbitrary configuration. *Trans. Intell. Transport. Sys.* **23**(7), 7535–7546 (jul 2022). <https://doi.org/10.1109/TITS.2021.3071391>, <https://doi.org/10.1109/TITS.2021.3071391> 2, 3, 4
17. Kato, S., Tokunaga, S., Maruyama, Y., Maeda, S., Hirabayashi, M., Kitsukawa, Y., Monrroy, A., Ando, T., Fujii, Y., Azumi, T.: Autoware on board: Enabling autonomous vehicles with embedded systems. In: 2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS). pp. 287–296 (2018). <https://doi.org/10.1109/ICCPS.2018.00035> 6
18. Kuhn, H.W.: The hungarian method for the assignment problem. *Naval Research Logistics Quarterly* **2**(1-2), 83–97 (1955). <https://doi.org/https://doi.org/10.1002/nav.3800020109>, <https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800020109> 8
19. Leng, Z., Wang, Y., Xin, M., Minor, M.A.: The effect of sideslip on jackknife limits during low speed trailer operation. *Robotics* **11**(6) (2022), <https://www.mdpi.com/2218-6581/11/6/133> 2
20. Li, Y., Wu, H.: A clustering method based on k-means algorithm. *Physics Procedia* **25**, 1104–1109 (2012). <https://doi.org/https://doi.org/10.1016/j.phpro.2012.03.206>, <https://www.sciencedirect.com/science/article/pii/S1875389212006220>, international Conference on Solid State Devices and Materials Science, April 1-2, 2012, Macao 2
21. Luo, W., Jiang, C., Zhang, Q., Pan, Z., Heng, L.: Lidar-assisted hitch angle estimation system for self-driving truck. In: 2024 IEEE Intelligent Vehicles Symposium (IV). pp. 2660–2666 (2024). <https://doi.org/10.1109/IV55156.2024.10588838> 4
22. Maćkiewicz, A., Ratajczak, W.: Principal components analysis (pca). *Computers Geosciences* **19**(3), 303–342 (1993). [https://doi.org/https://doi.org/10.1016/0098-3004\(93\)90090-R](https://doi.org/https://doi.org/10.1016/0098-3004(93)90090-R), <https://www.sciencedirect.com/science/article/pii/009830049390090R> 2
23. Olutomilayo, K.T., Bahramgiri, M., Nooshabadi, S., Oh, J., Lakehal-Ayat, M., Rogan, D., Fuhrmann, D.R.: Trailer angle estimation using radar point clouds. *Signal Processing* **188**, 108221 (2021). <https://doi.org/https://doi.org/10.1016/j.sigpro.2021.108221>, <https://www.sciencedirect.com/science/article/pii/S0165168421002589> 2, 4
24. OTLE: Informe anual 2022 del observatorio de transporte y logística de españa. aa (2023) 2
25. Sager, C., Zschech, P., Kühn, N.: labelcloud: A lightweight domain-independent labeling tool for 3d object detection in point clouds. *CoRR* **abs/2103.04970** (2021), <https://arxiv.org/abs/2103.04970> 10
26. Schulz, C., Hanten, R., Zell, A.: Efficient Map Representations for Multi-Dimensional Normal Distributions Transforms. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 2679–2686. Madrid, Spain (October 2018). <https://doi.org/10.1109/IROS.2018.8593602> 3
27. Shepard, D.R.: Dual purpose hitch sensor (August 2016), <https://patents.google.com/patent/US20160236526A1/en>, patent filed on 2015-02-12, published on 2016-08-18. 3
28. Shepard, D.R.: Imu-based hitch angle sensing device (March 2017), <https://patents.google.com/patent/US20170089697A1/en>, patent filed on 2017-03-30, published on 2017-03-30. 3, 4, 9

29. Waabi: Waabi research (2024), <https://waabi.ai/research/>, accessed: 2024-08-08 **2**
30. Wang, F., Zhao, Z.: A survey of iterative closest point algorithm. In: 2017 Chinese Automation Congress (CAC). pp. 4395–4399 (2017). <https://doi.org/10.1109/CAC.2017.8243553> **5**
31. Wurm, K., Hornung, A., Bennewitz, M., Stachniss, C., Burgard, W.: Octomap: A probabilistic, flexible, and compact 3d map representation for robotic systems. vol. 2 (01 2010) **3**
32. Yu, H., Luo, Y., Shu, M., Huo, Y., Yang, Z., Shi, Y., Guo, Z., Li, H., Hu, X., Yuan, J., Nie, Z.: Dair-v2x: A large-scale dataset for vehicle-infrastructure cooperative 3d object detection (2022), <https://arxiv.org/abs/2204.05575> **3**
33. Zimmer, W., Birkner, J., Brucker, M., Tung Nguyen, H., Petrovski, S., Wang, B., Knoll, A.C.: Infradet3d: Multi-modal 3d object detection based on roadside infrastructure camera and lidar sensors. In: 2023 IEEE Intelligent Vehicles Symposium (IV). pp. 1–8 (2023). <https://doi.org/10.1109/IV55152.2023.10186723> **3**
34. Zimmer, W., Creß, C., Nguyen, H.T., Knoll, A.C.: A9 intersection dataset: All you need for urban 3d camera-lidar roadside perception (2023), <https://arxiv.org/abs/2306.09266> **3**
35. Zimmer, W., Wardana, G.A., Sritharan, S., Zhou, X., Song, R., Knoll, A.: Tumtraf v2x cooperative perception dataset. arXiv preprint arXiv:2403.01316 (2024) **3**