2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

Addressing Data Annotation Challenges in Multiple Sensors: A Solution for Scania Collected Datasets

Ajinkya Khoche[1,2], Aron Asefaw[1], Alejandro González[2], Bogdan Timus[2], Sina Sharif Mansouri[2] and Patric Jensfelt[1]

***Abstract*— Data annotation in autonomous vehicles is a critical step in the development of Deep Neural Network (DNN) based models or the performance evaluation of the perception system. This often takes the form of adding 3D bounding boxes on time-sequential and registered series of point-sets captured from active sensors like Light Detection and Ranging (LiDAR) and Radio Detection and Ranging (RADAR). When annotating multiple active sensors, there is a need to motion compensate and translate the points to a consistent coordinate frame and timestamp respectively. However, highly dynamic objects pose a unique challenge, as they can appear at different timestamps in each sensor's data. Without knowing the speed of the objects, their position appears to be different in different sensor outputs. Thus, even after motion compensation, highly dynamic objects are not matched from multiple sensors in the same frame, and human annotators struggle to add unique bounding boxes that capture all objects. This article focuses on addressing this challenge, primarily within the context of Scania-collected datasets. The proposed solution takes a track of an annotated object as input and uses the Moving Horizon Estimation (MHE) to robustly estimate its speed. The estimated speed profile is utilized to correct the position of the annotated box and add boxes to object clusters missed by the original annotation.**

I. Introduction

The pursuit of autonomous vehicles, particularly in heavy vehicle manufacturing, has gained momentum across various industries. At its core, the essential element driving the progress is ground truth data, essential for evaluating the Autonomous Vehicles (AV) software stack. To obtain this invaluable data, several approaches have emerged. One method involves equipping the ego and multiple non-ego vehicles with Global Positioning System (GPS) sensors and orchestrating staged scenarios, for instance, overtaking, U-turns, roundabouts etc. While the GPS enables comprehensive state-awareness of the environment and provides valuable insights, this approach can't be extended to real-world driving. As such, highly controlled scenarios also fall short in emulating the complexities of real-world driving, leaving gaps in the generalization ability of the system.

Deep Neural Networks (DNNs) have recently emerged as the backbone of autonomous driving systems, with various approaches proposed throughout the AV stack, including for perception [1], [2], localization [3], motion prediction and situational awareness [4], control and path planning [5], vehichle-to-vehicle communication [6] as well as end-to-end driving systems [7]. A majority of DNN approaches are trained using supervised learning and require annotations. Simulation systems offer another avenue for generating training data at scale. However, the challenge lies in bridging the gap between simulated and real-world sensory output [8]. The successful integration of synthetic data into practical experiments remains a persistent question in the journey toward autonomous vehicles. The third approach, albeit time-consuming and expensive, involves annotating datasets collected from vehicles on the road. This method necessitates rigorous quality checks.

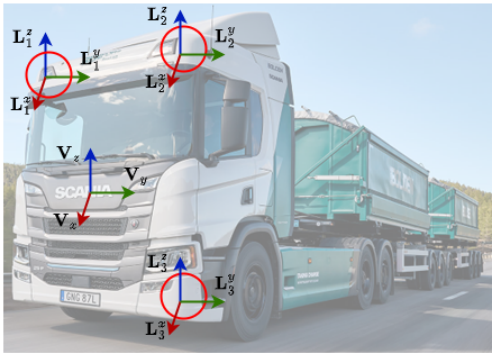

Fig. 1: Illustration of the Scania truck with different sensor placement highlighted with red circles, while the i^{th} sensor and the vehicle coordinate frame are shown as $\mathbf{L_i}$ and $\mathbf{V}$ respectively.

The field of autonomous driving has seen an abundance of datasets capturing challenging real driving scenarios. The KITTI dataset, proposed by Geiger et. al. [9], was a pioneering work in this regards, providing annotations for 3D object detection, stereo matching and optical flow, as well as high quality position labels to enable research in Simulataneous Localization and Mapping (SLAM). The NuScenes [10], Waymo [11] and Argoverse [12] datasets are notable for their large scale and diversity, including night-time driving and adverse weather scenarios. Many datasets also provide access to High Definition (HD) maps to enable advanced processing [12], [10]. The recently proposed Argoverse2 [13], aiMotive [14] and Zenseact [15] datasets provide annotations focusing on long range perception.

However most of the existing datasets are captured onboard passenger cars, and their characteristics differ significantly from the data captured onboard trucks. Due to

*This work was supported by the research grant PROSENSE (2020-02963) funded by VINNOVA.

[1]KTH Royal Institute of Technology, Stockholm 10044, Sweden. Corresponding author's e-mail: khoche@kth.se

[2]Autonomous Transport Solutions Lab, Scania Group, Södertälje, SE-15139, Sweden

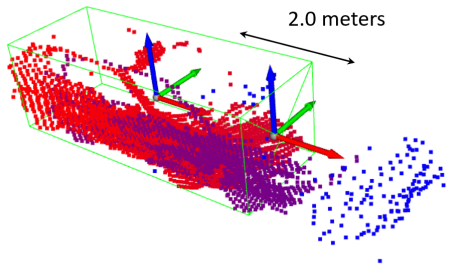

Fig. 2: Point cloud from three sensors after compensating for ego motion, (red, violet and blue points are from LiDARs L_1, L_2 and L_3 respectively). The dynamic object in the scene is observed by different sensors in different time stances, with displacement of 2 m within 100 ms. The manually annotated 3D bounding box is shown in green.

their larger size compared to passenger cars, trucks require a greater number of sensors with increased spacing between them to provide comprehensive surround views, as shown in Figure 1. This extended displacement between the sensors leads to them capturing different, often non-overlapping views of the same object. Moreover, the available datasets mostly consist of suburban scenarios with low driving speeds, whereas long haulage trucks operate on highways, with objects moving at high speeds. These dynamic objects are captured at multiple positions by various sensors, even after ego-motion compensation. To the best of our knowledge, none of the existing datasets capture this phenomenon.

The aforementioned issues pose notable challenges for human annotators when attempting to define accurate 3D bounding boxes around objects. Firstly, the human-labelled bounding boxes may not encompass the entirety of the point clouds, leading to scenarios where portions of the objects, as illustrated in Figure 2, remain outside the bounding box's scope. Secondly, the annotators might label different views of an object at various time instances, leading to inaccurate speed estimation of the vehicle. These annotations, if not refined may lead to an incorrect evaluation of perception algorithms, or be a source of error during training of DNN models.

In this work, we address this problem by modelling the annotated boxes as noisy measurements of the object state. Consequently, state estimation algorithms can be used to infer the object's true state. Given an annotated object track from a multi-LiDAR dataset as noisy positional inputs, this article proposes using Moving Horizon Estimation (MHE) as a state estimator to predict the position and speed of non-ego objects. The estimates are subsequently used to refine the positioning of bounding box annotations to cover all the views of the object.

A. Background & Motivation

Kalman Filter (KF) [16] is widely used in applications where the system dynamics and measurement models are both linear and the noise is Gaussian, while Extended Kalman Filter (EKF) can handle non-linear system and measurement models. KF based estimators may have slow convergence for rapid changes in state, and only consider one measurement for each estimation iteration. Nonlinear Moving Horizon Estimation (NMHE) methods are also getting more attention [17], [18], [19] for their ability to estimate complex nonlinear dynamic models, while they can handle inequality constraints. MHE method uses a moving time window to iteratively estimate the states of a nonlinear dynamic system, providing real-time updates as new measurements become available. The MHE, driven by its optimization-based framework and its ability to utilize a set of measurements, is a preferred choice for accurately estimating the speed of non-ego vehicles in various scenarios, with the added advantage of being able to handle constraints such as bounds on vehicle speed, making it a versatile solution.

B. Contributions

With the abovementioned state-of-the-art as the context, the key contributions of the article are provided in this section. The first and foremost contribution is to highlight the problem of annotation for multiple active sensors in heavy vehicles, which to the best of authors' knowledge, has not been discussed before. As the second contribution, a MHE estimator with kinematic model is formulated to estimate the non-ego vehicle speeds, which is used to further rectify the bounding box annotations. The third contribution stems from the experimental evaluation of the method on a LiDAR dataset captured onboard trucks and buses containing diverse scenarios of non-ego object motion.

C. Outline

The rest of the article is organized as follows: Section II outlines the problem. The MHE formulation is described in Section III. Section IV sets up the experiments and discusses results on data captured onboard Scania platform. Lastly Section V concludes the article by summarizing the findings and discussing directions for future work.

II. Problem Statement

Let $\mathbf{W}$, $\mathbf{V_t}$ and $\mathbf{L_t^i}$ represent the world frame, the vehicle frame at time t, and the i^{th} sensor frame at time t respectively. ${}^{W}\mathbf{T}_{V_t}$ denotes the 4×4 homogeneous transformation matrix representing the vehicle's pose in the world frame at time t. Let ${}^{L_t^i}\mathbf{P}_i(t)$ denote a point cloud acquired by the i^{th} LiDAR sensor during the time period $[t, t+\Delta t]$. The j^{th} point in this point cloud is represented by a 3D position $p_{i,j} = (x_{i,j}, y_{i,j}, z_{i,j})^T$ in the sensor frame and timestamp $\tau_{i,j}$. Since the ego-vehicle is moving continuously while the rotating LiDAR captures data, a transformation needs to be applied to each point of the point cloud, to compensate for the ego-motion to simplify processing by deskewing the point cloud. Let t^* denote the time to which the motion is compensated. In our work $t^* = t + \Delta t/2$, i.e. in the center of the scan period of the point cloud. We denote by ${}^{V_{t^*}}\mathbf{P}_i^*$ the motion compensated point cloud where the position, $p_{i,j}$,

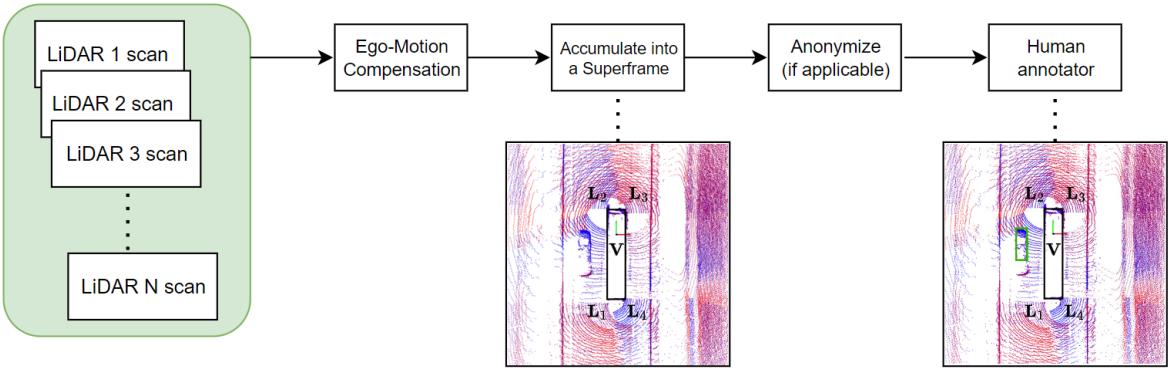

Fig. 3: The annotation process for Scania collected dataset. Scans from multiple LiDARs are motion compensated and accumulated in a *superframe*. Thereafter a time-series of superframes are post processed and sent for manual annotation. A snapshot of a superframe before and after annotation is shown. The ego vehicle is marked in the center with vehicle frame $\mathbf{V}$, and sensor frames $\mathbf{L}_1$ to $\mathbf{L}_4$. The point clouds from multiple LiDARs are colored according to their offset from motion compensated timestamp (chosen to be the middle of the superframe). Red and blue indicate beginning and end of the superframe respectively. The annotated vehicle is shown with a green box.

of each point is transformed to the vehicle frame at time t^*. The motion compensated points are denoted ${}^{V_{t^*}}p^*_{i,j}$, which for brevity will be written as $p^*_{i,j}$ and are calculated as:

$$p^*_{i,j} = {}^{V_{t^*}}\mathbf{T}_{V_{\tau_{i,j}}} \, {}^{V_{\tau_{i,j}}}\mathbf{T}_{L^i_{\tau_{i,j}}} p_{i,j} \tag{1a}$$

$${}^{V_{t^*}}\mathbf{T}_{V_{\tau_{i,j}}} = ({}^{W}\mathbf{T}_{V_{t^*}})^{-1} \, {}^{W}\mathbf{T}_{V_{\tau_{i,j}}} \tag{1b}$$

where ${}^{V_{\tau_{i,j}}}\mathbf{T}_{L^i_{\tau_{i,j}}}$ is given by the extrinsic calibration and is here assumed constant ${}^{V}\mathbf{T}_{L^i}$. A constant linear and angular velocity is assumed in the interval $[t, t + \Delta t]$.

The focus of our work is a multi-LiDAR setup. We define a *superframe* ${}^{V_{t^*}}\mathbf{P}_S$ as a point cloud accumulating all motion compensated points from M LiDARs within a time interval $[t, t + \Delta t]$,

$${}^{V_{t^*}}\mathbf{P}_S = \bigcup_{i=1}^{M} {}^{V_{t^*}}\mathbf{P}^*_i. \tag{2}$$

A time-series of motion compensated superframes is post-processed and sent for annotation of 3D boxes, as shown in Figure 3. Notably, Eq. (1) follows a static world assumption, which does not hold in real-world driving applications. For example, for $\Delta t = 100$ ms, a non-ego vehicle driving at 30 m/sec on a highway could have a worst case displacement of up to 3 meters captured by different sensors within the superframe. A single bounding box is inadequate in capturing this motion, as shown in Fig. 3. The problem then involves modeling the motion of the non-ego object given noisy measurements of the state provided by a time series (or a track) of annotated 3D bounding boxes.

An agent's 3D motion can be generally described by 12 states: $\boldsymbol{x} = [x, y, z, \dot{x}, \dot{y}, \dot{z}, \phi, \gamma, \theta, \dot{\phi}, \dot{\gamma}, \dot{\theta}]^T$ where the first six states denote the positions and linear velocities, and the last six states denote the roll, pitch and yaw angles and their rates respectively. Particularly for driving scenarios, a planar motion with holonomic constraints is considered. This assumption removes the need to estimate the z position, the roll and pitch angles, as well as their rates, reducing the state space to six. Furthermore, assuming the measured heading of an object remains constant within the superframe interval Δt, and the heading is error-free, the problem can be further simplified to a one-dimensional estimation containing two states $\boldsymbol{x} = [d, s]^T$, where d and s are the distance and speed along a specified trajectory.

III. Methodology

In this section, we will present our MHE approach. We will begin by outlining the mathematical formulation that underlies our method, offering an explanation of the equations and principles. Subsequently, we will explain the estimation of non-ego vehicle speeds in MHE. Finally, we will present the architecture that rectifies annotations, providing a holistic view of our approach's practical implementation.

A. Kinematic Model

Given the state $\boldsymbol{x} = [d, s]^T$ as described in the previous section, the distance d is specified along a trajectory specified by a set of error-free headings $\boldsymbol{\Theta} = \{\theta_i : i = 1, ..., n_l\}$, where n_l is the length of an annotated track. The measurements here come from human annotations. The speed s is not measured. The state of a non-ego vehicle at time t is assumed to be given by the constant acceleration model (3).

$$d(t) = d_0 + s(t)t + \frac{1}{2}at^2, \tag{3a}$$

$$s(t) = s_0 + at. \tag{3b}$$

In this work we study short trajectories. A more general solution would need to adopt a more complex motion model.

B. Moving Horizon Estimation

The main objective of the MHE [20] is to obtain the state estimate at time t, given a set of measurements collected over a moving horizon of past time steps. The state and measurement are modelled in discrete form as:

$$\bar{\boldsymbol{x}}_{k+1} = \mathcal{F}(\bar{\boldsymbol{x}}_k, \boldsymbol{u}_k) + \boldsymbol{w}_k, \tag{4a}$$

$$\boldsymbol{y}_k = \mathcal{H}(\bar{\boldsymbol{x}}_k) + \boldsymbol{\Lambda}_k, \tag{4b}$$

where, $\bar{\boldsymbol{x}}_{k+1}$ is the state estimate at time step $k+1$, $\mathcal{F} : \mathbb{R}^{n_s} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_s}$ is the state transition function and $\boldsymbol{u}_k$ is the control input at time step k. $\boldsymbol{u}_k$ is the same as the acceleration a. $\boldsymbol{y_k}$ is the modelled measurement and $\mathcal{H} : \mathbb{R}^{n_s} \rightarrow \mathbb{R}^{n_m}$ is the measurement function. Moreover n_s, n_u, and n_m are the number of states, inputs and measurements respectively, $\boldsymbol{\Lambda}_k \in \mathbb{R}^{n_m}$ and $\boldsymbol{w}_k \in \mathbb{R}^{n_s}$ represent the measurement noise and the process noise correspondingly. The initial state estimate $\bar{\boldsymbol{x}}_0$ is known. Furthermore, $\bar{\boldsymbol{x}}_{k-j|k}$ and $\boldsymbol{y}_{k-j|k}$ are the previous $k-j$ state and measurements as observed from the current time k.

The process noise $\boldsymbol{w}_k$, measurement noise $\boldsymbol{\Lambda}_k$ and the initial state estimate noise are unknown and assumed to follow a random distribution, characterized by the Gaussian Probability Density Function (PDF) with the covariance $\boldsymbol{Q} \in \mathbb{R}^{n_s \times n_s}$, $\boldsymbol{\Omega} \in \mathbb{R}^{n_m \times n_m}$, and $\boldsymbol{\Psi} \in \mathbb{R}^{n_s \times n_s}$ respectively [21].

Given a set of noisy measurements $\boldsymbol{Y} = \{\boldsymbol{y}_j : j = k - N_e, ..., k-1\}$ in a fixed horizon window N_e , the estimated states of the system $\bar{\boldsymbol{X}} = \{\bar{\boldsymbol{x}}_j : j = k - N_e, ..., k-1\}$ are obtained by solving the optimization problem:

$$\min_{\bar{\boldsymbol{x}}_{(k-N_e|k)}, \boldsymbol{W}_{(k-N_e|k)}^{(k-1|k)}} J(k) \tag{5a}$$

$$\text{s.t. } \bar{\boldsymbol{x}}_{i+1|k} = \mathcal{F}(\bar{\boldsymbol{x}}_{i|k}, \boldsymbol{u}_{i|k}) + \boldsymbol{w}_{i|k} \tag{5b}$$

$$\boldsymbol{y}_{i|k} = \mathcal{H}(\bar{\boldsymbol{x}}_{i|k}) + \boldsymbol{\Lambda_{i|k}} \quad i = \{k - N_e, \dots k-1\} \tag{5c}$$

$$\boldsymbol{w}_k \in \mathbb{W}_k, \quad \boldsymbol{\Lambda}_k \in \mathbb{\Lambda}_k, \quad \bar{\boldsymbol{x}}_k \in \mathbb{X}_k \tag{5d}$$

where,

$$J(k) = \underbrace{(\bar{\boldsymbol{x}}_{k-N_e|k} - \tilde{\boldsymbol{x}}_{k-N_e|k})^T \boldsymbol{\Psi}^{-1} (\bar{\boldsymbol{x}}_{k-N_e|k} - \tilde{\boldsymbol{x}}_{k-N_e|k})}_{\text{arrival cost}}$$
$$+ \sum_{i=k-N_e}^{i=k} \underbrace{(\boldsymbol{y}_{i|k} - \mathcal{H}(\bar{\boldsymbol{x}}_{i|k}))^T \boldsymbol{\Omega}^{-1} (\boldsymbol{y}_{i|k} - \mathcal{H}(\bar{\boldsymbol{x}}_{i|k}))}_{\text{stage cost}} + \sum_{i=k-N_e}^{i=k-1}$$
$$\underbrace{(\bar{\boldsymbol{x}}_{i+1|k} - \mathcal{F}(\bar{\boldsymbol{x}}_{i|k}, \boldsymbol{u}_{i|k}))^T \boldsymbol{Q}^{-1} (\bar{\boldsymbol{x}}_{i+1|k} - \mathcal{F}(\bar{\boldsymbol{x}}_{i|k}, \boldsymbol{u}_{i|k}))}_{\text{stage cost}} \tag{6}$$

In (5) $\boldsymbol{W}_{(k-N_e)}^{(k-1)} = col(\boldsymbol{w}_{(k-N_e)}, \dots, \boldsymbol{w}_{(k-1)})$ is the estimated process disturbance from time $k - N_e$ up to $k-1$, estimated at the time k. The first term of the objective J is referred to as the arrival cost. It measures the squared difference between the current and the prior state estimate at the beginning of the horizon window. In effect, the arrival cost is a mechanism for incorporating historical state information into the current estimation problem, ensuring a smooth transition from past estimates to current estimates. The remaining terms are denoted as stage cost or incremental cost. They compute the sum of the squared difference between the predicted and modelled measurement, and the predicted and modelled state respectively. The predictions come from the measurement function and the state transition function. Additionally, the terms are weighted by covariances of the inverse of initial state estimate, measurement and the process noise respectively. A smaller covariance indicates higher confidence in the previous estimate, leading to a larger penalty for deviations. A finite-horizon optimal problem with horizon window of N_e is solved at every time step k, to obtain the corresponding state estimates $\bar{\boldsymbol{x}}^{\star}_{k-N_e|k}, \dots \bar{\boldsymbol{x}}^{\star}_{N_e|k}$.

C. Refining Multi-LiDAR Annotations

The goal of our work is to refine the manual annotations for each non-ego vehicle. We do this by generating boxes corresponding to each individual LiDAR, given a single annotation per non-ego object for each superframe. Solving for Equation (5) traversing along the horizon window N_e provides the optimal state estimate for the entire track, denoted as $\bar{\boldsymbol{X}}^{\star}_{1:n_m}$. These estimates, in conjunction with the human-annotated bounding boxes, are utilized as the input for box refinement, as shown in Figure 4. The first step in this approach involves clustering along the box heading to get G different views of the object captured by various sensors within a superframe. As the second step, the estimated speed $s^{\star}_k$ is used alongside the box heading θ_k to compensate all points classified as being part of the non-ego vehicle for its speed. Concretely, for all LiDAR's i and points j that correspond to the non-ego vehicle we calculate the change in position as:

$$\Delta p_{i,j} = \begin{bmatrix} \Delta x_{i,j} \\ \Delta y_{i,j} \\ \Delta z_{i,j} \end{bmatrix} = \begin{bmatrix} (\tau_{i,j} - t^{\star}) s^{\star}_k \cos\theta_k \\ (\tau_{i,j} - t^{\star}) s^{\star}_k \sin\theta_k \\ 0 \end{bmatrix} \tag{7a}$$

(7b)

Where $\tau_{i,j}$ is the timestamp of point $p_{i,j}$, and t^* is the motion compensation time. $\Delta p_{i,j}$ is added to $p_{i,j}$ to obtain speed-compensated points for the non-ego vehicle. Thereafter the front or rear of the vehicle are inferred using the highest density region [22] along the heading. If the high density region lies behind the euclidean mean, the region represents the back of the object, else the front. This knowledge allows the algorithm to position the bounding box by anchoring the edges to the extreme points of the vehicle along the direction of travel. Next, G copies of the bounding box are produced, one for each cluster. These are denoted as *pseudo bounding boxes*. Lastly, these copies are each shifted back by the inverse of $\Delta p_{i,j}$, leading to the pseudo boxes fitting precisely with the original clusters.

IV. Experiments

In this section, we present our experimental setup, as well as discuss results of the state estimation using MHE and the annotation refinement. In this article, we focus on highly dynamic vehicles as the problem is more pronounced for

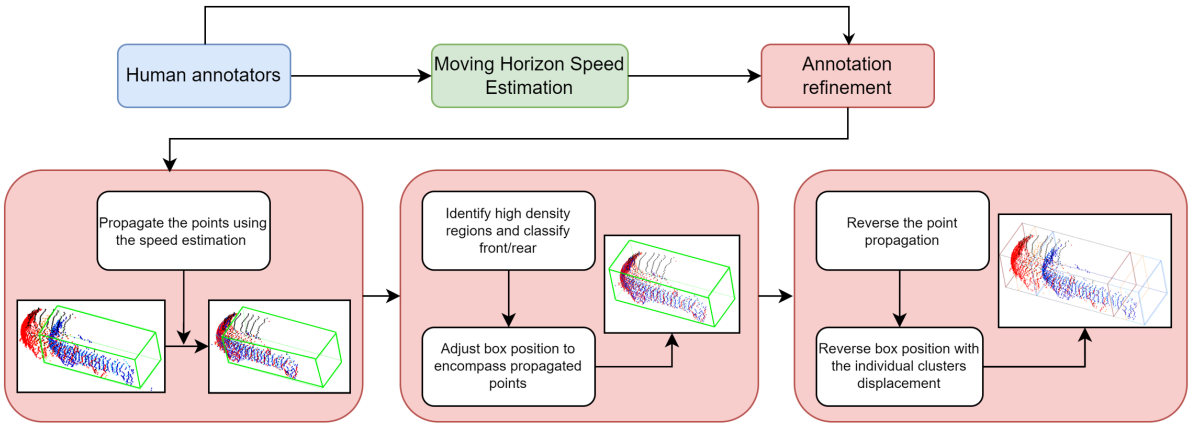

Fig. 4: Proposed approach for refining multi-LiDAR annotations. The input is time-sequential track of human labelled annotation (shown with green box) alongside it's speed estimate obtained using MHE. Clustering along heading captures red points, which represent a view of the vehicle missed by the human annotated box. Next, the MHE speed estimate is used to shift the points according to the difference between $\tau_{i,j}$ and t^*. The red points move forward, whereas the blue points move back. Thirdly, the annotated box is moved to align with the shifted points. Lastly, the green box is duplicated for each cluster, and both the points and pseudo bounding boxes are shifted back according to the reverse cluster displacement. The pseudo bounding boxes are colored according to their best fitting clusters.

such cases. But our approach could be extended to cover other classes eg. articulated vehicles, pedestrians etc. as well.

A. Experimental Setup

Sequences of multi-LiDAR data was collected and annotated on Scania platforms consisting of trucks and buses. The dataset encompasses a wide range of scenarios, including urban and highway driving, as well as challenging adverse weather conditions. The annotated sequences have a duration of 10 s and include motion-compensated LiDAR points captured at 10 FPS, 3D bounding boxes, class labels, and tracking IDs for each object. The annotators utilized keyframes to extract essential parameters, such as class and size. Keyframes are selected based on the time sequence, specifically focusing on the frames that capture the highest quality representation of the objects in that sequence. It's important to note that this selection may vary for different objects. It's worth mentioning that, for the sake of generality, we did not account for this variability in our work.

For state estimation using MHE, the kinematic model described in Section III-A is utilized as the state transition model $\mathcal{F}$. Since the problem at hand is solved offline, we chose the horizon window N_e to be the same as entire length of measurements, i.e. length of an annotated track. The state estimates of the entire track are thus optimized in a single iteration. The MHE parameters are indicated as $n_s = 2$ and $n_m = 1$, as only the distance is measured. $\mathbf{\Psi}^{-1} = \mathbf{I}_{2\times2}$, $\boldsymbol{Q} = \mathbf{I}_{2\times2}$, and $\mathbf{\Omega} = \mathbf{I}_{1\times1}$ where $\mathbf{I}$ is identity matrix.

B. Results

A comparison between MHE, a Kalman Filter [16] based estimation, and a basic speed estimation approach is depicted in Figure 5. Four non-ego vehicle tracks are sampled at random across various logs. For convenience, the tracks are chosen such that they are visible for the major duration of the sequence. The KF estimate, denoted by dotted black, maintains the same state space and the state transition model as MHE. The basic, or naive speed estimate shown in blue, is obtained by simply dividing the distances and times between the annotation intervals. The MHE estimate, shown in red, follows a smooth trajectory due to the stage cost in Eq. (6) being constrained by the kinematic model. On the other hand, the naive estimate follows an irregular speed curve, which is due to the human annotator labelling differing and inconsistent views of the object at different time instances. MHE also helps in removing outliers in naive speed estimate in extreme cases (Figs. 5c and 5d). The recusrive KF estimate is observed to be less smooth compared to MHE. Although a detailed comparison is challenging due to lack of precise ground truth, the smoother motion produced by MHE estimates makes it a more appropriate method.

The results of annotation refinement for the selected vehicles, as mentioned in the previous paragraph, are depicted in Figure 6. The human-annotated box, colored in green, clearly misses various views, as indicated by the presence of red and black points in rows one to three and orange and violet points in row four. These discrepancies are effectively addressed by the refined annotations or pseudo bounding boxes, which are color-coded based on the point clusters they encompass, allowing them to accurately represent these perspectives. Of particular interest, rows 1-3 depict partially observed vehicles. The process of identifying high-density regions plays a pivotal role in classifying the rear side of the vehicle, facilitating the precise fitting of bounding boxes on speed-compensated clusters. This step was observed to be crucial for obtaining accurate pseudo bounding boxes.

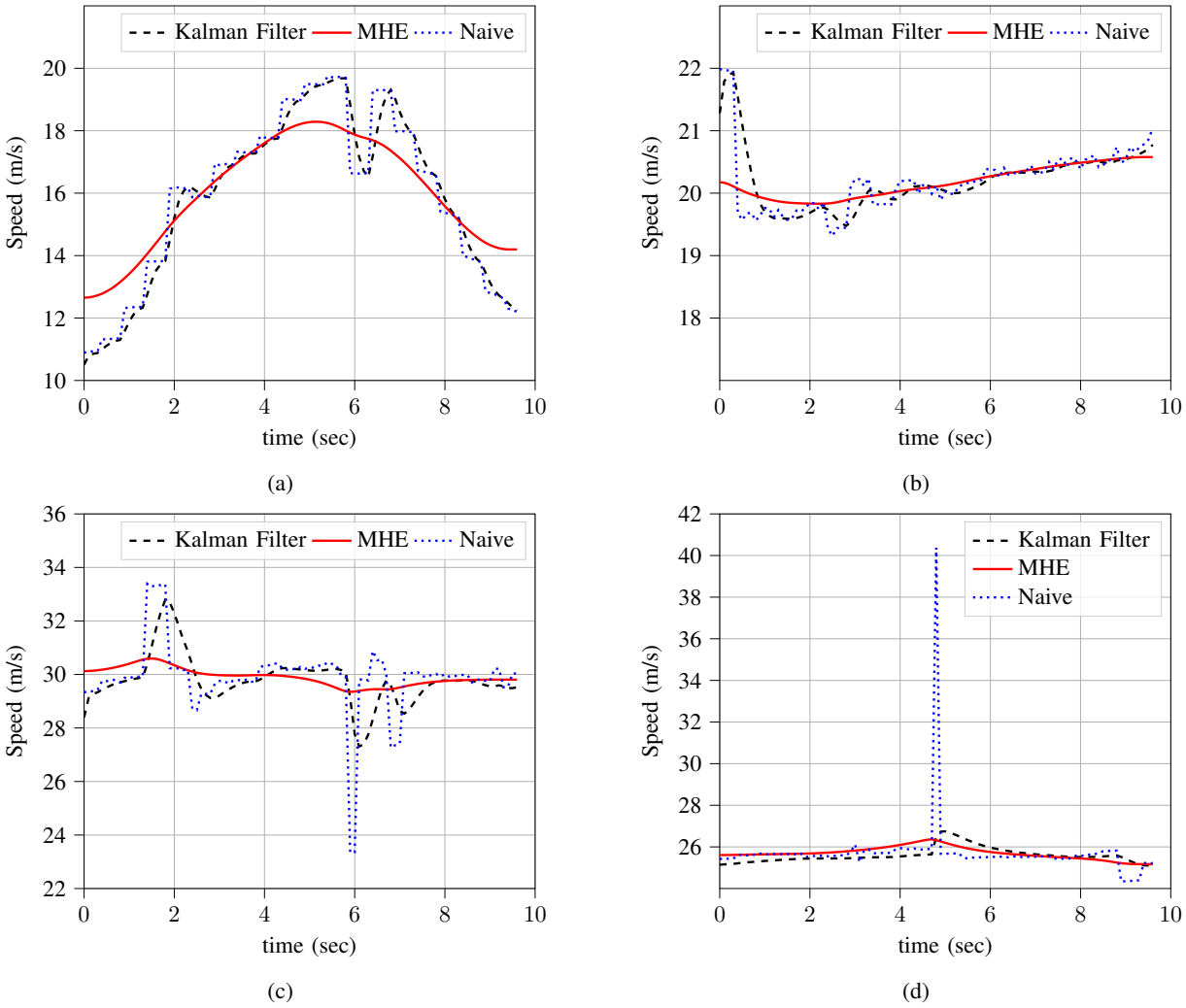

Fig. 5: Estimation of speed for four different non-ego vehicles across various logs. The blue plot represents the MHE estimates, whereas the orange plot denotes the naive speed estimate, obtained by dividing the distances and times in between the annotation intervals.

V. Conclusions and Future Work

This article has presented a solution to the data annotation challenges associated with heavy vehicles equipped with multiple active sensors. We have utilized MHE estimators to estimate the speed of non-ego objects and rectify bounding boxes. The effectiveness of the proposed solution is demonstrated through an evaluation using real-life data gathered and annotated by Scania. Looking ahead, there are several avenues for further research and development. One such direction involves tailoring the modeling approach based on the specific class of objects, such as bicycle models, articulated vehicles, pedestrians, and so on. This customization can potentially enhance the accuracy of MHE speed estimation. Moreover, the current framework operates on 10 s frames. Future research can focus on extending its application to longer time sequences without annotations, providing a cost-effective means to expand annotated datasets. A prior step for this would involve using the refined annotations to train DNN algorithms for object detection, tracking etc. In summary, the proposed solution represents a significant step forward in addressing data annotation challenges in heavy vehicle sensor systems. With ongoing research and innovation, we can anticipate further advancements in this field, contributing to the evolution of safer and more efficient transportation technologies and addressing a common bottleneck in developing machine learning models for autonomous vehicles and other applications.

References

[1] A. Khoche, M. K. Wozniak, D. Duberg, and P. Jensfelt, "Semantic 3d grid maps for autonomous driving," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2022, pp. 2681–2688.

[2] A. Khoche, L. P. Sánchez, N. Batool, S. S. Mansouri, and P. Jensfelt, "Fully sparse long range 3d object detection using range experts and multimodal virtual points," *arXiv preprint arXiv:2310.04800*, 2023.

[3] C. Chen, B. Wang, C. X. Lu, N. Trigoni, and A. Markham, "Deep learning for visual localization and mapping: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

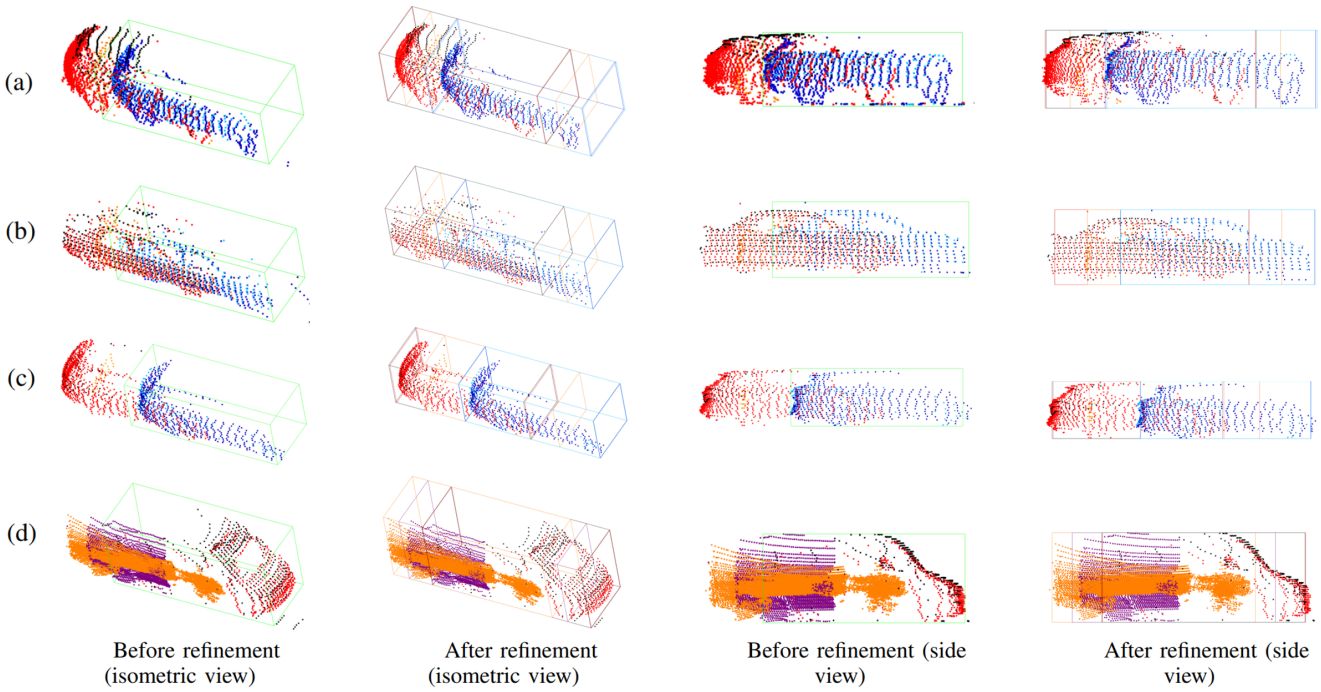

Fig. 6: Comparison of annotations before and after refinement proposed in Section III-C. Rows 1-4 represent objects whose speed plots are given in Fig. 5 (a)-(d) respectively. The green box represents human annotated bounding box. The point colors distinguish points captured by different LiDAR sensors. Following refinement, the pseudo bounding boxes are color-coded based on the point clusters that they are aligned with. The pseudo bounding boxes successfully capture missing views of the object.

[4] A. Seff, B. Cera, D. Chen, M. Ng, A. Zhou, N. Nayakanti, K. S. Refaat, R. Al-Rfou, and B. Sapp, "Motionlm: Multi-agent motion forecasting as language modeling," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 8579–8590.

[5] S. Teng, X. Hu, P. Deng, B. Li, Y. Li, Y. Ai, D. Yang, L. Li, Z. Xuanyuan, F. Zhu *et al.*, "Motion planning for autonomous driving: The state of the art and future perspectives," *IEEE Transactions on Intelligent Vehicles*, 2023.

[6] Y. Li, D. Ma, Z. An, Z. Wang, Y. Zhong, S. Chen, and C. Feng, "V2x-sim: Multi-agent collaborative perception dataset and benchmark for autonomous driving," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 914–10 921, 2022.

[7] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang *et al.*, "Planning-oriented autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 853–17 862.

[8] X. Hu, S. Li, T. Huang, B. Tang, R. Huai, and L. Chen, "How simulation helps autonomous driving: A survey of sim2real, digital twins, and parallel intelligence," *IEEE Transactions on Intelligent Vehicles*, 2023.

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

[10] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.

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

[12] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan *et al.*, "Argoverse: 3d tracking and forecasting with rich maps," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 8748–8757.

[13] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J. K. Pontes *et al.*, "Argoverse 2: Next generation datasets for self-driving perception and forecasting," *arXiv preprint arXiv:2301.00493*, 2023.

[14] T. Matuszka, I. Barton, Á. Butykai, P. Hajas, D. Kiss, D. Kovács, S. Kunsági-Máté, P. Lengyel, G. Németh, L. Pető *et al.*, "aimotive dataset: A multimodal dataset for robust autonomous driving with long-range perception," *arXiv preprint arXiv:2211.09445*, 2022.

[15] M. Alibeigi, W. Ljungbergh, A. Tonderski, G. Hess, A. Lilja, C. Lindstrom, D. Motorniuk, J. Fu, J. Widahl, and C. Petersson, "Zenseact open dataset: A large-scale and diverse multimodal dataset for autonomous driving," *arXiv preprint arXiv:2305.02008*, 2023.

[16] R. E. Kalman, "A new approach to linear filtering and prediction problems," 1960.

[17] E. L. Haseltine and J. B. Rawlings, "Critical evaluation of extended kalman filtering and moving-horizon estimation," *Industrial & engineering chemistry research*, vol. 44, no. 8, pp. 2451–2460, 2005.

[18] A. Papadimitriou, H. Jafari, S. S. Mansouri, and G. Nikolakopoulos, "External force estimation and disturbance rejection for micro aerial vehicles," *Expert Systems with Applications*, vol. 200, p. 116883, 2022.

[19] S. S. Mansouri, H. Jafari, and G. Nikolakopoulos, "External force estimation based on nonlinear moving horizon estimation for mav navigation," in *2020 European Control Conference (ECC)*. IEEE, 2020, pp. 1312–1317.

[20] C. V. Rao, J. B. Rawlings, and D. Q. Mayne, "Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations," *IEEE transactions on automatic control*, vol. 48, no. 2, pp. 246–258, 2003.

[21] S. Ungarala, "Computing arrival cost parameters in moving horizon estimation using sampling based filters," *Journal of Process Control*, vol. 19, no. 9, pp. 1576–1588, 2009.

[22] E. Parzen, "On estimation of a probability density function and mode," *The annals of mathematical statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.