BB-Align: A Lightweight Pose Recovery Framework for Vehicle-to-Vehicle Cooperative Perception

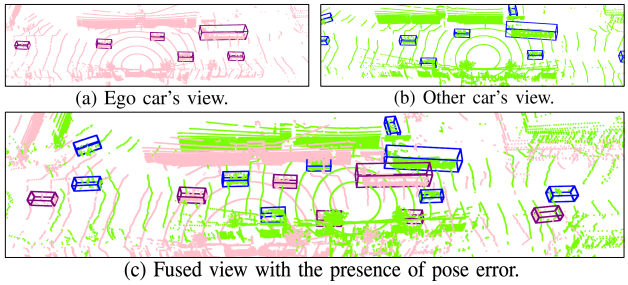
Lixing Song†, William Valentine†, Qing Yang‡, Honggang Wang*, Hua Fang , Ye Liu°
†Rose-Hulman Institute of Technology, ‡University of North Texas, *Yeshiva University,
University of Massachusetts Dartmouth °Macau University of Science and Technology
song3, valentwa @rose-hulman.edu, Qing.Yang@unt.edu, Honggang.Wang@yu.edu, hfang2@umassd.edu, liuye@must.edu.mo

2024 IEEE 44th International Conference on Distributed Computing Systems (ICDCS) | 979-8-3503-8605-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICDCS60910.2024.00098

***Abstract*—Vehicle-to-Vehicle (V2V) cooperative perception has become increasingly popular in the field of autonomous driving, effectively overcoming the inherent limitations of single-vehicle perception systems, such as limited range and susceptibility to occlusions. In a V2V system, vehicles in close proximity can share perception data. To fuse this data, which is collected from different viewpoints by each vehicle, accurate pose information (including position and heading direction) is essential to transform the received data to the receiving vehicle's viewpoint. However, pose errors, often caused by measurement noise or sensor failures, can lead to severe misalignment during data fusion, resulting in incorrect object detections and potentially hazardous decisions in autonomous driving systems. To address this challenge, we present *BB-Align*, a lightweight pose recovery framework that utilizes Lidar Bird's-eye View (BV) images and object bounding Boxes for relative pose estimation. Designed as a plug-and-play solution, the proposed method requires no additional model training, enabling effortless integration into existing V2V systems. Our approach uses Lidar-derived BV images with a Log-Gabor filter-based feature map for effective image matching despite image sparsity. To reduce errors from self-motion distortion, we also integrate object bounding boxes for finer alignment. The proposed method is rigorously evaluated on the V2V4Real dataset—currently the only real-world V2V dataset. Our approach demonstrates high pose estimation accuracy, outperforming an existing graph-matching method. It achieves translation and rotation errors of less than and °, respectively, in 80% of cases within a range between vehicles. Furthermore, by integrating the proposed framework into cooperative object detection models under serious pose error, the result shows up to a 2x increase in Average Precision (AP) compared to those without pose recovery, with more pronounced improvements in the short range.**

I. Introduction

Autonomous driving, an increasingly popular field, critically relies on exceptional perception capabilities. However, the perception system of a single car often falls short in capturing the complexities of busy and dynamic traffic environments, plagued by limitations such as short range and occlusions. Vehicle-to-Vehicle (V2V) cooperative perception, where vehicles in proximity share perception data, has emerged as a significant solution to mitigate these limitations. Recent V2V works [1, 2] have demonstrated remarkable performance in tasks like 3D object detection and tracking, significantly surpassing the performance of single-car systems. Nevertheless, this data-sharing approach presents challenges, including communication bandwidth limitation and latency in data delivery, which can adversely affect cooperative perception performance.



(a) Ego car's view. (b) Other car's view.

(c) Fused view with the presence of pose error.

Fig. 1: Illustration of the impact of pose error on data fusion in V2V cooperative perception. The points with different colors represent Lidar scans from different cars, while the bounding boxes indicate objects detected by the corresponding cars.

One of the most critical challenges in cooperative perception is the error in positioning information (including GPS coordinates and heading direction), commonly referred to as pose error. Since perception data from different vehicles is captured from different viewpoints, pose information is required to transform the viewpoint of the shared data to align with the view of the ego car (the car receiving the data) for data fusion. However, this information can be corrupted due to various reasons, such as GPS device failure, measurement noise, and transmission errors. The corrupted pose can significantly diminish the perception accuracy. As illustrated in Fig. 1, when the ego car employs erroneous pose information to fuse Lidar data received from another car, the resulting misalignment causes the ego car to incorrectly detect objects at inaccurate locations, potentially impairing the vehicle's driving strategy.

To address the challenge of pose error in V2V cooperative perception, existing studies [2, 3, 4] have explored developing robust neural network models to cope with such errors. A more recent work [5] integrates agent-object pose graph optimization for error correction. However, these approaches all involve extensive neural network training and often struggle to maintain their efficacy in scenarios that they have not been explicitly trained for. On the other hand, rigid registration, as a traditional, non-neural-network-based approach, is commonly employed in the field of robotics. The method matches two overlapping datasets (like Lidar points or images) to determine the relative pose. However, these methods typically require

2575-8411/24/$31.00 ©2024 IEEE
DOI 10.1109/ICDCS60910.2024.00098

similar sensor configurations. In the context of V2V systems, where vehicles may be equipped with different Lidar systems, the heterogeneous sensor setup poses a significant hurdle, frequently preventing traditional registration methods from delivering consistent results. Furthermore, applying 3-D registration in the V2V system requires transmitting the entire Lidar point cloud between vehicles, which can lead to significant communication bandwidth consumption.

To provide a dataset-independent and bandwidth-conserving solution, in this paper, we present *BB-Align*, a two-stage Lidar Bird's-eye view image and object bounding Box-based pose recovery framework for V2V cooperative perception. To estimate the relative pose between two cars, our proposed method first employs an image-matching approach on the Lidar Bird's-eye View (BV) images for a high-level alignment. Subsequently, by aligning object bounding boxes detected by both cars, the method can achieve finer alignment. Designed as a plug-and-play module, our method requires no additional model training and can be easily integrated into existing V2V systems. Building on an image-matching technique, our method operates independently of prior pose information and is capable of recovering pose errors at any severity. Overall, the main contributions of this paper are summarized as follows:

- We design a two-stage pose recovery framework for V2V cooperative perception. The initial stage utilizes the Lidar BV images obtained from different vehicles to perform image matching. To address the sparsity of BV images, we use a Log-Gabor filter-based feature map, the Maximum Index Map (MIM) [6], to identify subtle features as keypoints for accurate image matching.
- To further achieve finer alignment, our framework employs a second-stage refinement process to mitigate errors caused by self-motion distortion. This stage utilizes the bounding boxes from object detection. By aligning the overlapped but initially unaligned boxes, we can further enhance the accuracy of pose estimation.
- We conduct a rigorous performance evaluation of the proposed method on the V2V4Real dataset [7], currently the only available real-world V2V dataset. The results demonstrate remarkable accuracy in pose estimation. Furthermore, when applied to cooperative object detection tasks, our method significantly improves Average Precision (AP) compared to scenarios without the pose recovery framework.

II. Related Work and Background

Cooperative Perception. Due to challenges such as limited range and occlusion in single-car perception, cooperative perception—where cars in proximity share sensory data—has garnered significant attention, as highlighted in surveys like [8, 9]. Most research in this field has been focused on exploring ways of fusing different perception data, such as maps [10], sensory data [11], or processed features [12, 2, 1]. Fig. 2 illustrates the different fusion methods. The early fusion that simply combines raw Lidar points was attempted in pioneer works like Cooper [11]. Despite its potential,

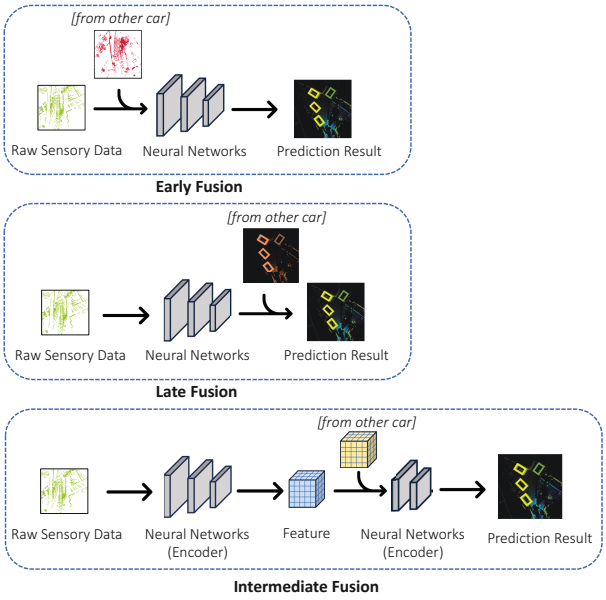


Fig. 2: Different fusion methods for cooperative perception.

early fusion raises concerns regarding high communication bandwidth cost for transmitting raw data (like Lidar points). To conserve bandwidth while sharing important observations, intermediate fusion, which shares neural network-generated intermediate feature maps, has become increasingly popular. Various types of feature maps and network structures have been investigated [12, 13, 2, 1, 14, 15]. For instance, F-Cooper [12] employs a simple *maxout* operation to process feature maps from two cars; V2VNet [13] uses a spatially aware graph neural network (GNN); AttFuse [15] integrates a self-attention mechanism. Recently, with the advent of the powerful Transformer architecture, works like V2X-ViT [2] and coBEVT [1] have adapted it to address noise issues and enhance feature extraction. In contrast, late fusion, which though requires the least bandwidth cost by only sharing detection results, has been shown to underperform in experiments, as evaluated in benchmarks like [7, 16].

Pose Calibration Solutions for V2V. When fusing data, erroneous pose information can critically compromise cooperative perception performance, potentially jeopardizing driving strategies in autonomous driving. To address this issue, previous works [2, 3, 4] have focused on developing robust neural networks incorporating pose error correction components. For instance, [3] designed a module specifically to predict the relative pose error between cars, while [2] adopted a multi-agent approach to capture spatial relationships between agents, thereby reducing localization noise. More recently, CoAlign [5] has utilized agent-object pose graph optimization to correct pose errors, demonstrating resilience to a certain degree of pose inaccuracy. However, these neural network-based methods, being dataset-specific, often struggle in untrained scenarios and require model re-training for different application domains, which may not be practical or efficient. In contrast,

our proposed work is designed as a stand-alone plug-and-play module that can effortlessly integrate with any fusion model or method. It operates independently of prior pose information, offering enhanced resilience to pose errors at any degree.

3-D Rigid Registration. For recovering the relative pose information, rigid registration is a commonly used solution in robotics, particularly in SLAM (Simultaneous Localization and Mapping), In the domain of 3D point-set registration, a classical approach for estimating the relative pose between two Lidar scans[1] is Iterative Closest Point (ICP) [17]. ICP aligns two scans iteratively by minimizing the distance between point clouds. Building upon this, global registration techniques [18, 19] have developed more effective methods for point cloud alignment. With the advent of neural networks, recent approaches [20, 21, 22] have demonstrated exceptional performance in registration for Lidar scans. However, since they are training-based, they often exhibit reduced effectiveness in scenarios that differ from their training environments. Furthermore, in the V2V context, employing Lidar points for pose estimation involves transmitting entire point clouds, encountering the same bandwidth concern as early fusion methods. Therefore, in our pursuit to develop a dataset-independent method and also to minimize bandwidth consumption, 3-D registration is not an ideal choice for V2V pose recovery.

2-D Image Matching. Leveraging Bird's-eye View (BV) images from point clouds offers a bandwidth-efficient alternative to 3-D point-set registration that can estimate the relative pose on the ground plane. Image matching has been well-established over decades [23], with traditional methods like SIFT (Scale Invariant Feature Transform) [24] focusing on detecting keypoints and matching them. However, the sparsity of Lidar-generated BV images presents significant challenges for these methods, often failing to detect meaningful features. The use of Log-Gabor filters has shown promise in feature extraction for both optical satellite and Lidar-generated images [25, 26], with recent studies adapting this technique for BV image-based applications [27]. Nevertheless, the dynamic nature of V2V scenarios, with varying sensor configurations and vehicle movements, poses unique challenges to these techniques. To overcome these challenges, this paper introduces a novel two-stage design integrating BV images and object bounding boxes for precise alignment, effectively addressing the limitations of traditional image matching methods.

Graph Matching. Graph matching techniques construct graphs with detected objects as nodes and the distances between them as edges, enabling the estimation of relative pose transformations between observations from different vehicles. Notably, VIPS [28] employs a spectral-based matching approach tailored for a Vehicle-to-Infrastructure (V2I) setup. Another recent study [29] leverages intra-agent geometrical context to enhance feature descriptiveness for matching. However, these methods depend heavily on the dense spatial patterns formed by surrounding traffic and struggle in light traffic conditions. Our experimental results, detailed later, demonstrate that our proposed method consistently outperforms graph matching across various traffic scenarios.

[1]A scan refers to a set of cloud points collected by a lidar sensor during one single scan of the surroundings.

III. The Setup of Pose Recovery for V2V

Problem Description. In the context of V2V cooperative perception, we assume that two vehicles, each equipped with lidar sensors, are capable of exchanging sensory data along with their pose information. When the ego car receives perception data from another vehicle, it utilizes the informed pose to appropriately transform the received data to its own point of view. This transformed data is then integrated with the ego car's own data. However, a corrupted pose can critically compromise this data fusion process. Our proposed work aims to recover the pose by estimating the relative pose for data fusion while minimizing the need for additional data transmission to conserve bandwidth.

The relative pose between vehicles comprises two components: translation and rotation. Translation is represented by a 3-D Cartesian coordinate , indicating the positional shift from the other car to the ego car. Rotation is described as a tuple of three degrees , corresponding to *yaw*, *roll*, and *pitch*, which denote the angular differences in orientation between the two cars. However, for ground vehicles like cars, *roll*, *pitch*, and the coordinate typically remain constant. The movement of cars is predominantly on the plane (ground), which primarily involves changes in , , and *yaw*. Based on this intuition, our work proposes a bird's-eye view (BV) approach, effectively transforming the challenge of 3-D pose recovery into a 2-D problem.

Pose Recovery. In the context of two-car cooperative perception, for the application of our pose recovery method, the other car needs to transmit its BV image and object bounding boxes from its objection detection model to the ego car. Due to the highly compressed nature of BV images, the communication cost associated with transmitting this information is significantly lower compared to transmitting raw Lidar data or even processed feature maps. Upon receipt of this information, the ego car can estimate the relative pose from the other car to the ego car. This output is a 3-degree of freedom transformation denoted as , where represents rotation and denote translation. Given this 2-D transformation, we can further construct the corresponding 3-D transformation. This transformation can be represented as a 3-D homogeneous transformation matrix. As aforementioned, given the application scenarios of ground vehicles, we assume *pitch* ( ), *roll* ( ), and the translation on the -axis ( ) mostly remain constant. By combining all these parameters, we can recover the transformation matrix which is defined as

(1)

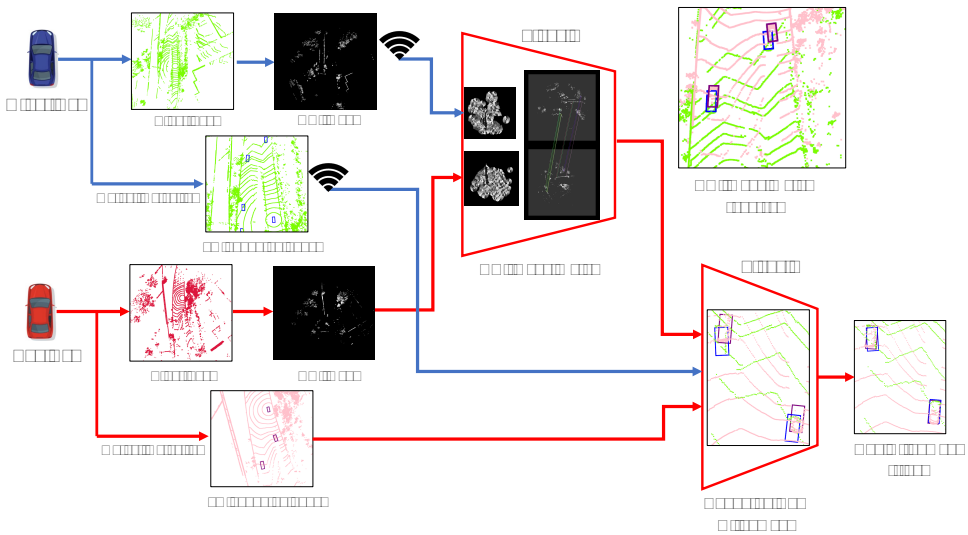


Fig. 3: Overview of the proposed two-stage pose recovery framework. Colored arrows and trapezoids represent data paths and modules in different cars, with red for the ego car and blue for the other car. The WiFi icons denote the information transmission from the other car to the ego car. The first stage applies image matching on BV images generated from both cars. It utilizes a Log-Gabor filter-based feature map (small images inside of the upper trapezoid) to enhance image matching. Although the initial matching results may contain minor errors, the second stage employs object bounding boxes to further refine and correct any remaining misalignments.

where can be expressed as

(2)

Note that the bold variables are estimated from our proposed work while the non-bold variables are pre-defined constant values. With this estimated transformation matrix, we can transform the received perception data to the viewpoint of the ego car. For example, assuming a point is received from the other car, we can compute its transformed position in the view of the ego car as

(3)

where the operator means extracting the first di-mension. In the next section, we will explore in detail the technical aspects of employing our method to estimate the transformation matrix .

IV. The Proposed 2-Stage Framework

In this section, we elaborate on the design details of our proposed *BB-Align* framework. The structure of the framework is visualized in Fig. 3. As depicted, our method employs a two-stage approach. The first stage is based on BV image matching, indicated by the upper trapezoid in Fig. 3, which will be discussed in detail in Section IV-A. Followed by Section IV-B, we will introduce the second-stage object bounding box alignment, as illustrated by the lower trapezoid in the diagram. As a summary, we will conclude with an overview of the algorithm in Section IV-C.

A. BV Image Matching

Generating BV Image. Given a set of 3-D points $\mathcal{P}$ where $\mathcal{R}$ and is the number of points, there exist several approaches for generating BV image [30, 31, 27]. For our specific objective of pose recovery, we want to utilize tall objects such as buildings and trees as salient landmarks for matching. Therefore, we adopt the height map approach [30]. To generate a BV image from $\mathcal{P}$, we first partition the points into 2-D cells on the ground plane ( ) with a resolution parameter (cell size). Assuming the Lidar range is , we define a cell set within the region as $\mathcal{C}$, where —. A BV image $\mathcal{B}$ with dimensions is generated using the maximum height in each cell as the pixel intensity. The intensity of a pixel in this BV image is defined as

(4)

Compared to the density map approach [31], which uses point density for pixel intensity, the height map approach not only enables the use of stationary high objects as reliable landmarks but also inherently filters out ground-hitting points. These ground points typically provide no meaningful information and can be detrimental to effective image matching.

Creating Feature Map. Given a pair of 2-D images, the standard process for image matching usually includes four steps: 1) detecting keypoints, 2) computing feature descriptors,

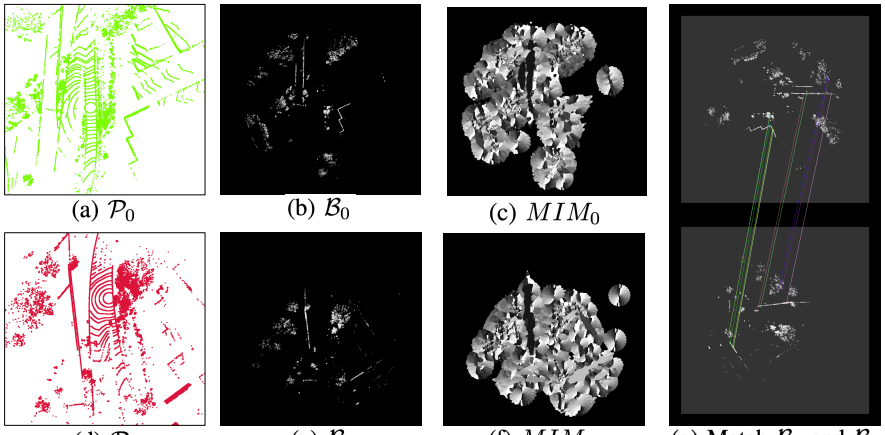


Fig. 4: Step-by-step illustration of BV image matching, plotted with real-world data from two cars driving 45 meters apart. Subfigures (a), (b), and (c) depict the point cloud, BV image, and feature map MIM generated from one car, respectively. Correspondingly, subfigures (d), (e), and (f) represent the same components from the other car. Subfigure (g) displays the matching results, with lines connecting the matched keypoints.

3) matching keypoints based on the descriptors, and 4) estimating the transformation between matched keypoints. However, the extreme sparsity of Lidar BV images poses significant challenges, particularly in detecting keypoints and computing effective descriptors. To address this challenge, inspired by the previous work [25, 27], we adopt a Log-Gabor filter-based approach to generate feature maps for BV images.

The Log-Gabor filter is an effective tool in texture analysis and image representation, primarily due to its capability in capturing fine texture features at different orientations. This characteristic is crucial for identifying key features such as the edges of buildings and tree tops in Lidar BV images. Given a BV image $\mathcal{B}$, we can create a feature map called Maximum Index Map (MIM) based on [6, 25]. To do that, we first convert the image into the polar coordinates with

(5)

In the spatial domain, the response of a 2-D Log-Gabor filter is given by:

(6)

where and are parameters of the filter, which represents the scale and the preferred orientation respectively. are bandwidth hyperparameters. Applying such a filter to an image facilitates the extraction of features at the specific scale, as determined by , and at a particular orientation, as indicated by . This method is particularly effective in isolating image characteristics that are significant at certain spatial frequencies and orientations. To capture features across different scales and orientations, we can apply a bank of filters with different parameters. Let be an array of orientations and an array of scales[2], where and are the number of scales and orientations, respectively. To capture image characteristics in a variety of scales and orientations, we can pass the image through all filters in the filter bank. In this filter bank, by specifying indices and , a single filter can be selected, and its response, redefined by substituting in Equation 6, is given by

(7)

By applying this filter to a BV image , the amplitude of under this specific filter can be expressed as

(8)

where denotes the convolution operation. Then, the amplitude for a certain orientation over all scales can be obtained by summing up the amplitudes at all scales with that same orientation, as

(9)

Finally, a feature map, Maximum Index Map (MIM), is computed by identifying the index of the orientation that yields the maximum amplitude, which can be expressed as

(10)

Inherently, the value of the Maximum Index Map (MIM) at a specific location in the image indicates the direction/angle of the dominant features observed at that point. This attribute is crucial in identifying subtle features in sparse images, such as recognizing disconnected lines as edges, isolated blobs as

[2]The setting of can be found in [32].

tree tops, and similar elements.

Detecting Keypoints & Computing Descriptors. By Utilizing keypoint detectors such as FAST [33], we can identify keypoints, including edges and corners. For each detected keypoint, we compute a descriptor (a vector) using its surrounding pixels in the feature map. This process starts with defining a square area around the keypoint, known as a *patch*, covering an area of $J \times J$ pixels. This patch then is subdivided into $l \times l$ grids. Within each grid, a histogram $h(o)$ is calculated, representing the orientation frequency on the MIM. As a result, a feature vector with dimensions $l \times l \times N_o$ can be obtained.

It is important to note that while the MIM provides scale-invariant features, it does not inherently offer rotation invariance; rotations change orientations, consequently affecting the MIM values. To address this limitation, we integrate the Bird's-eye View Feature Transform (BVFT) descriptors designed in [27]. This design adopts a strategy akin to that used in the classic computer vision method ORB [34], which computes the dominant orientation within a patch and then rotates the patch to align its dominant orientation with a fixed direction like $0°$. As a result, the pixel values in MIM can remain constant regardless of rotation.

Matching Keypoints & Estimating Transformation. Once the descriptors for keypoints from two BV images are computed, we match these keypoints based on the similarity of their descriptors. This similarity is measured by the Euclidean distance between the descriptors. With the matched keypoints in pairs and their corresponding positions, we employ the RANdom SAmple Consensus (RANSAC) algorithm to estimate the relative pose between the two images. RANSAC typically returns a 2-D homogeneous transformation matrix and the number of inliers—keypoints that align within a predefined error threshold under this transformation. In practical applications, the count of the inliers can serve as an indicator of the success of the match. Overall, this image-matching procedure is illustrated in Fig. 4. Notably, despite the extreme sparsity of the BV images (as illustrated in Fig. 4(b) and Fig. 4(e)), the MIM-based approach can effectively capture the thin lines as keypoints which are the edges of the buildings.

B. Fine Alignment using Object Bounding Box

Self-Motion Distortion. While BV image matching is effective in identifying and aligning prominent landmarks, its accuracy is compromised by minor errors, primarily from self-motion distortion in Lidar point clouds. Self-motion distortion occurs due to the movement of the Lidar sensor during the data acquisition process. To elaborate, capturing a complete scan of Lidar points is not instantaneous but requires a certain amount of time. During this period, if the sensor moves, its viewpoint changes accordingly. As a result, the points captured at different moments during the scan correspond to slightly different viewpoints. This variance in viewpoints leads to discrepancies in the data, known as self-motion distortion. In V2V scenarios, where cars often travel at different speeds and in varying directions, this issue becomes more pronounced. As a result, while the successful alignment of prominent large objects such as buildings and trees can be achieved, there remains a discrepancy when attempting to align smaller objects, like cars. This challenge is illustrated in Fig. 5 using real-world data.

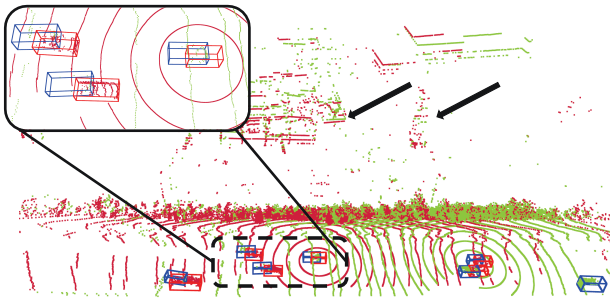
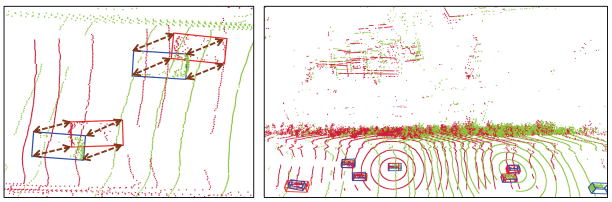


Fig. 5: Misalignment caused by self-motion distortion. The point clouds from the two cars, colored in green and red, are aligned using the BV image match (stage 1) alone. The 3-D bounding boxes, indicated in blue and red, highlight objects (cars) detected by different cars. The arrows emphasize the well-aligned building edges. The dashed box, with a zoom-in view inside the solid box, reveals minor misalignments in the cars' positions.

Traditional solutions to mitigate self-motion distortion involve applying an odometry algorithm to estimate the movement of the sensor and then using it to recover non-distorted data. However, these methods require continuous computation on each Lidar scan collected, which is computationally intensive. Another approach is using the Iterative Closest Point (ICP) algorithm for point-to-point matching to correct misalignments. Nevertheless, this method encounters challenges when the same object is observed from different viewpoints. For instance, if two cars in the V2V system observe another target car from the front and the rear, merging these points directly on a point-to-point basis would be erroneous.



(a) (b)

Fig. 6: (a) shows the object bounding box matching process from the bird's-eye view. The red and green boxes represent objects detected from different cars. The double-headed arrows indicate the pairing of corners on the overlapped boxes. (b) shows the aligned result.

Fortunately, in the cooperative perception setup, the availability of object detection results provides valuable information about the positions and approximate dimensions of detected objects (vehicles), even when they are only partially observed. The access to object detection data enables us to

Algorithm 1: 2-Stage Pose Recovery

Both Cars: Generate BV images $\mathcal{B}_{\text{other}}$ and $\mathcal{B}_{\text{ego}}$ using Equation (4).
Both Cars: Perform object detection to obtain prediction results and project the 3-D object bounding boxes as 2-D BV boxes $_{\text{other}}$ and $_{\text{ego}}$.
Other Car: Send $\mathcal{B}_{\text{other}}$ and $_{\text{other}}$ to the ego car
Ego Car:
for *'ego' 'other'* **do**
    Compute from $\mathcal{B}$ using Equations (5) to (10).
    Detect keypoints and their positions from $\mathcal{B}$.
    Compute the descriptors from for .
Find similar descriptors between $_{\text{other}}$ and $_{\text{ego}}$ and record the corresponding matching indices $_{\text{other}}$ $_{\text{ego}}$.
Define source points $_{\text{src}}$ $_{\text{other}}$ $_{\text{other}}$ and destination points $_{\text{dest}}$ $_{\text{ego}}$ $_{\text{ego}}$.
Estimate 2-D transformation matrix from $_{\text{src}}$ to $_{\text{dest}}$.
Apply to transform $_{\text{other}}$ to $_{\text{other-trans}}$ under the view of the ego car.
Identify overlapping bounding boxes between $_{\text{other-trans}}$ and $_{\text{ego}}$, extracting corresponding corners $_{\text{other-trans}}$ and $_{\text{ego}}$.
Estimate 2-D transformation from $_{\text{other-trans}}$ to $_{\text{ego}}$.
Calculate the combined transformation $_2$.
Extract rotation , and translations $_x$ $_y$ from $_2$.
Construct the 3-D transformation $_3$ using Equation (1).

utilize these objects as anchors for performing more precise alignment. After applying the transformation result from the initial BV image match, the discrepancy is typically reduced to just 2 or 3 meters. In the majority of cases, the object bounding boxes representing the same targets, as detected by both cars, typically already exhibit significant overlap. Then, by using the corresponding corners of these overlapped boxes as matched keypoints, we can apply RANSAC to estimate another transformation to further align the boxes. In practical implementation, the corners of bounding boxes are stored as a sequence of points, consistently ordered in accordance with the 3-D Cartesian world coordinate system. This consistent ordering ensures that there is no confusion in matching the corresponding corners of the bounding boxes. Additionally, since our alignment is concentrated on the plane, we can simplify this task by projecting these bounding boxes as the bird's-eye view 2-D rectangles. The corner paring process and the aligning result are illustrated in Fig. 6.

C. Algorithm Overview

Algorithm 1 delineates the complete procedure of the proposed method. Note that the functions required to execute this algorithm, such as estimating the transformation given source and destination points, are standard geometric operations. Their implementations are readily available in computer vision libraries like `OpenCV`. Overall, through this two-stage algorithm, we can effectively recover the relative pose from the other car to the ego car. This is achieved without the need for prior pose knowledge and can recover pose error at any severity.

V. Performance Evaluation

In this section, we conduct a comprehensive performance evaluation of our proposed framework using a real-world V2V dataset. The experimental study aims to quantify the accuracy of the proposed method and explore the various factors that influence this accuracy. Additionally, we integrate our framework into existing V2V cooperative perception models to assess how the recovered pose information enhances object detection when faced with corrupted pose data. Finally, given our method's two-stage design, we conduct an ablation study to verify the individual contributions of different components to the overall system's efficacy.

Dataset. Most existing datasets for V2V cooperative perception, such as OPV2V [15] and V2Vset [2], are generated from virtual environments. While these simulations can provide realistic-looking road scenes, they often fall short in replicating the noisy, low-quality, or other undesirable conditions commonly encountered in real-world settings. Therefore, we have chosen to use the only real-world V2V dataset, V2V4Real [7], for evaluating our method. This dataset comprises 20K frames of Lidar scans from two vehicles, collected over 19 hours of driving. However, not all frames in this dataset are applicable for evaluating pose recovery. Instances where two cars exhibit minimal or no overlap in perception are not conducive for pose recovery. This technical limitation arises due to factors such as significant distance between the vehicles, occlusions, or divergent headings, which prevent sharing common observational data. These cases are therefore excluded from our study. We selected 12K frames out of the total 20K, focusing on those where at least *two* common cars are observed by both vehicles. This selection assures a sufficient overlap in the observed views of both cars.

Model Setup. Our implementation of BV image matching is based on the source code of previous work [27] and then integrated into the codebase of V2V4Real [7]. We configure the Log-Gabor filter for generating the MIM feature map with scales and orientations. Feature descriptors are computed using a patch size of and a grid size of . For bounding box prediction in object detection, we evaluate two models: the PointPillar-based F-Cooper [12] and the self-attention-enhanced coBEVT [1]. Though these models were originally designed as V2V fusion methods to combine feature maps from two cars, in our case, they are utilized as single-car object detection models, omitting the feature fusion aspect. Unless specified otherwise, coBEVT is used as the default object detection model. All experiments were conducted on a desktop equipped with an Intel i9-13900K CPU and an Nvidia Quadro A6000 GPU.

A. Pose Recovery Accuracy Study

Experiment Setup. In our experiments, for each pair of data from two cars comprising Lidar scans and object detection bounding boxes, we apply our proposed framework to compute the transformation matrix *from* the other car *to* the ego car. To calculate the accuracy, we compare our estimated pose parameters $_x$ $_y$, and (in Equation 1) with the ground

truth provided in the dataset. Furthermore, the accuracy is quantified using two metrics: **Translation Error**, which is the absolute difference on positional shift x y, and **Rotation Error**, which measures the absolute angular difference .

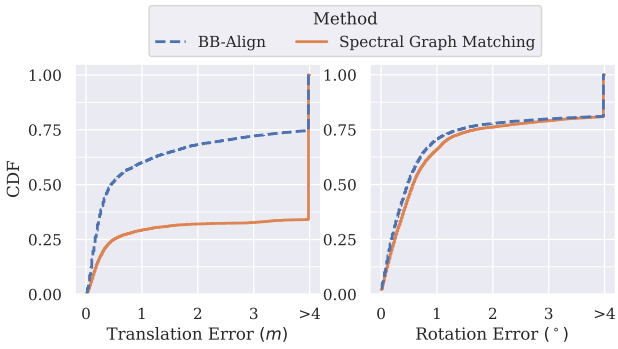


Fig. 7: Pose recovery accuracy comparison.

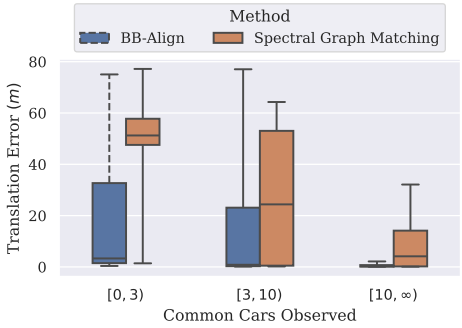
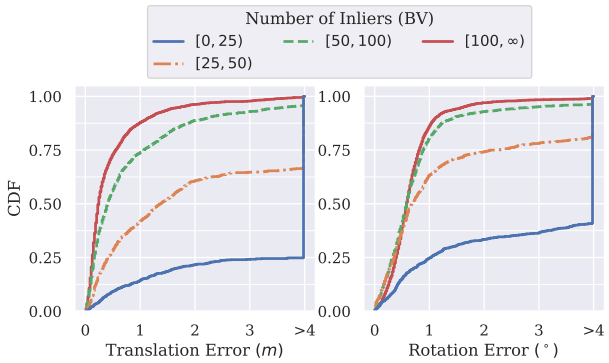
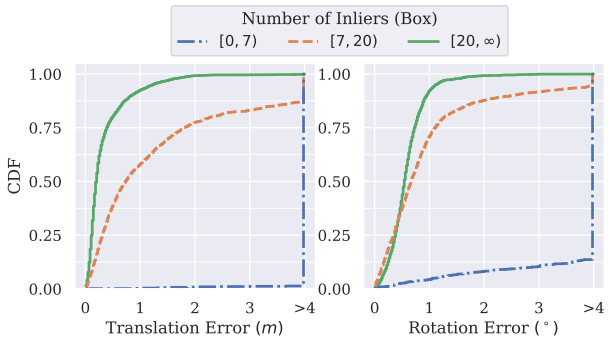


Fig. 8: Pose recovery accuracy w.r.t. commonly observed cars. Each box plot with whiskers represents the 10th, 25th, 50th, 75th, and 90th percentiles of the data.

Comparison. Most existing work in pose recovery for V2V relies on neural network training, whereas our approach employs a non-learning-based, plug-and-play methodology. Regarding image matching, we experimented with traditional methods like SIFT [24] and ORB [34]. However, these methods proved to be ineffective, failing to produce meaningful results. We selected to compare our work against another non-learning-based method, VIPS [28], which employs a graph matching approach to establish one-on-one correspondences between sets of objects detected by two different cars and then estimate the relative pose between the pairs. As shown in Fig. 7, our method outperforms the graph matching method, VIPS, particularly in terms of translation error. Approximately 60% of our pose estimations exhibit errors of less than 1 meter, compared to only about 30% for the graph matching method. The graph-matching method significantly depends on the unique spatial distributions formed by many surrounding vehicles. To validate this dependency, we illustrate the translation error comparison under varying counts of commonly observed cars in Fig. 8. In scenarios with sparse traffic (e.g., fewer than three surrounding vehicles), the graph-matching method struggles to identify effective matching features. As



(a) w.r.t. $\text{Inliers}_{\text{bv}}$.



(b) w.r.t. $\text{Inliers}_{\text{box}}$.

Fig. 9: Pose recovery accuracy w.r.t. Number of Inliers.

the number of surrounding vehicles increases, there is a noticeable decrease in translation error. However, due to the numerical instability associated with eigendecomposition in spectral graph matching, the overall accuracy of this method remains inferior to that of our proposed method. Note that the rotation error from both methods is comparable, as each method effectively captures the orientational features, which are prominently manifested by the traffic pattern and landscape along the road direction.

Success Rate. First, we examine the chances of successful pose recovery using our method. To determine the success of a recovery attempt, we rely on the number of inliers output by the RANSAC algorithm, which counts the keypoints that are aligned within a predefined error threshold. We denote the inlier count from BV image matching as $\text{Inliers}_{\text{bv}}$, and the count from bounding box alignment as $\text{Inliers}_{\text{box}}$. These inlier counts serve as indicators to assess the confidence of the pose recovery result. Fig. 9 shows the Cumulative Distribution Function (CDF) of translation and rotation errors for varying inlier counts in both matching processes, with Fig. 9(a) for BV image matching and Fig. 9(b) for box alignment. As observed, the accuracy improves with an increase in inliers. Specifically, where $\text{Inliers}_{\text{bv}}$ or $\text{Inliers}_{\text{box}}$, over 90% of cases have translation and rotation errors less than and . On the contrary, with low inlier counts, like $\text{Inliers}_{\text{box}}$ or $\text{Inliers}_{\text{box}}$, the result shows significantly lower accuracy. Based on this observation, we set an empirical

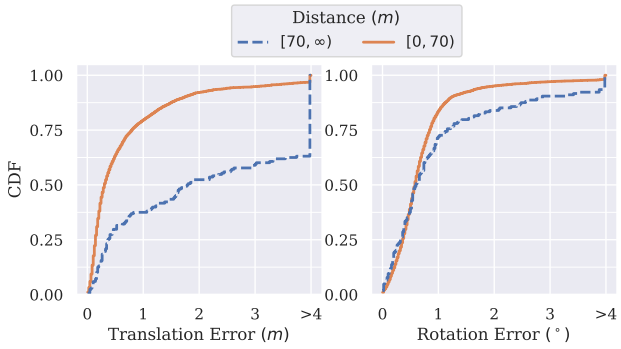


Fig. 10: Pose recovery accuracy w.r.t. distance ( ).

threshold: $\text{Inliers}_{\text{bv}}$ and $\text{Inliers}_{\text{box}}$ to define a successful recovery. With this criterion, out of a total 6,145 data pairs (12,290 frames), our method can successfully recover pose information for 80% (4,915 pairs) of data. Unsuccessful pose recoveries typically occur in scenarios where there are insufficient landmarks available for keypoint detection, such as in vast open areas without prominent objects.

In the subsequent analysis, we focus on examining the accuracy on the successful pose recoveries and investigate various factors that may influence this accuracy.

Distance. Fig. 10 illustrates the accuracy variation under different the distance between two cars. Notably, when the distance is within , approximately 80% of the data exhibits an estimated pose error of less than and . However, as the distance exceeds , while the translation accuracy decreases, the rotation error remains around for about 70% of data, This observation aligns with our expectations, as increased distances lead to sparser overlapped observations between the two cars, complicating the image matching process. In typical V2V scenarios, close-range cooperation between cars is often more critical, as it can influence immediate driving actions. The noteworthy performance of our method within a range is particularly significant, despite the less impressive results at longer distances.

B. Stage-by-Stage Investigation

Given the two-stage design of our system, we undertake on a stage-by-stage investigation of the two pivotal components: BV image matching and box alignment. This investigation is centered on examining the impact of the determining factors on each stage.

Impact of Distance on BV Image Matching. As indicated in the previous result in Fig. 10, our method shows sensitivity to longer distances. In Fig. 11, we present a more granular analysis of the accuracy of BV image matching *alone* across various distance categories. As expected, shorter distances correlate with higher accuracy. However, an observant reader might note that even in the most favorable scenario (distances less than), the accuracy is not better than the overall performance of the case depicted in Fig. 10. This highlights the necessity of the second-stage alignment. More

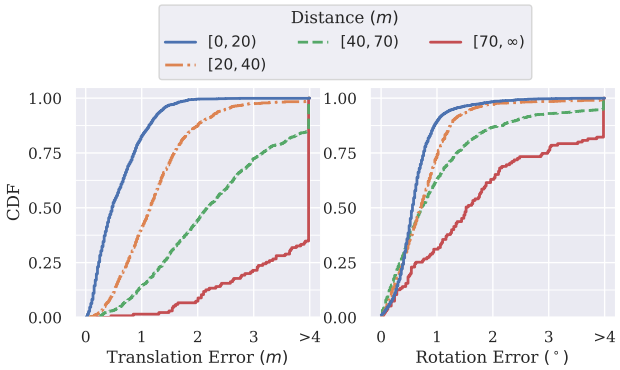


Fig. 11: Accuracy of BV image matching w.r.t. distance ( ).

detailed findings on this aspect will be discussed in our ablation study later in Section V-D.

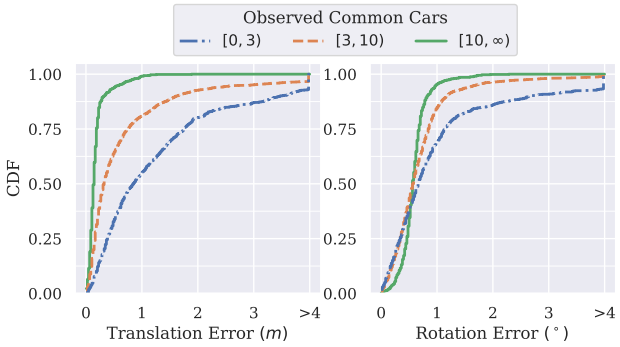


Fig. 12: Accuracy of box alignment (upon BV image matching) w.r.t. the number of commonly observed cars between the two vehicles.

Impact of Number of Common Cars on Box Alignment. For the second-stage alignment in our method, we leverage the cars commonly detected by both vehicles to achieve a finer alignment. Therefore, it is important to investigate how the number of commonly observed cars affects the box alignment accuracy. Fig. 12 presents the accuracy across different counts of commonly observed cars. As anticipated, a higher number of common cars, providing more bounding boxes for alignment, correlates with increased accuracy. While the accuracy deteriorates quickly with less than 3 cars observed, there are still 50% of cases exhibiting less than error. We argue that higher accuracy is particularly crucial for safety in busier traffic conditions—a scenario where our method excels in. Remarkably, when more than 10 cars are observed, over 90% of the cases have an error under and .

Impact of Object Detection Model on Box Alignment. Given that the second-stage box alignment relies on bounding boxes produced by the object detection model, we also investigate the impact of the choice of this model. Fig. 13 contrasts the results obtained using coBEVT and F-Cooper as the detection models. The findings indicate that the choice of model plays a minor role in the overall performance of our

Method	AP@IoU=0.5/0.7							
	t	θ °			Pose Recovered			
	Overall	0-30m	30-50m	50-100m	Overall	0-30m	30-50m	50-100m
Early Fusion	21.2/8.9	34.4/14.8	19.6/9.9	3.5/0.9	39.6/18.0	67.1/36.5	30.5/13.0	7.1/1.3
Late Fusion	18.7/9.3	33.1/18.9	16.8/7.9	2.5/0.6	33.9/12.9	63.0/28.3	27.0/9.2	4.7/0.7
F-Cooper	26.5/14.3	43.0/25.0	23.5/12.3	3.6/1.3	40.8/18.1	70.6/35.7	29.6/11.8	7.1/1.1
coBEVT	31.1/17.8	52.6/32.0	27.2/15.6	4.7/1.9	38.9/14.7	71.5/29.4	28.6/11.4	5.2/0.9

TABLE I: Comparison of object detection results under corrupted pose, with and without our pose recovery framework.

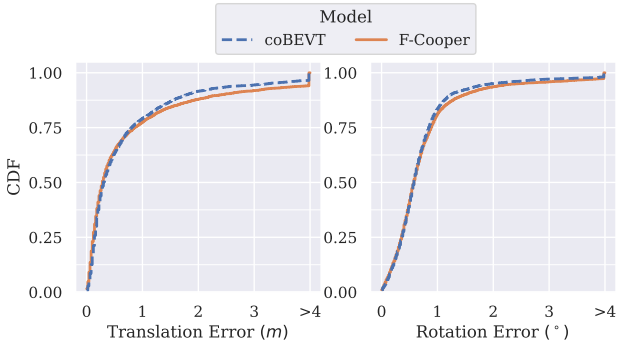


Fig. 13: Pose recovery accuracy w.r.t. distance ( ).

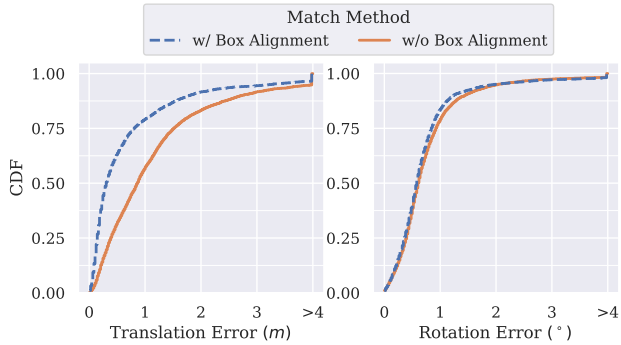


Fig. 14: Accuracy w/ and w/o the box alignment.

system. This suggests that our method is relatively independent of the object detection models, demonstrating its compatibility and potential for integration with various V2V systems.

C. Objection Detection Improvement

Experiment Setup. In this section, we apply our proposed framework to the task of object detection within the context of V2V cooperative perception. In this setup, the ego car uses the informed pose information from the other car to transform the perception data that is fed into the detection models for cooperative prediction. We examine various fusion methods, including early fusion (combining raw Lidar data), late fusion (merging predicted bounding boxes), and intermediate fusion (integrating feature maps from neural networks). The implementations for these methods are sourced from the V2V4Real repository. For intermediate fusion, we use F-Cooper [12] as an example of earlier work and coBEVT [1] as a representative of recent advancements in this field.

To evaluate our framework, we introduce errors into the pose information used by all fusion models. This is achieved by adding a zero-mean Gaussian noise with the standard deviations of for translation and for rotation. Then the models are tested both with and without the pose recovery framework. Table I shows the Average Precision (AP) with different Intersection Over Union (IoU) across various scenarios. For a more comprehensive analysis, we further categorize and break down the results based on the distances between the two cars.

As observed, the introduced noise significantly impairs the performance of all methods. With an advanced neural network design, methods like F-Cooper and coBEVT exhibit a degree of resilience against this noise, albeit still demonstrating unacceptable performance, with none method maintaining an Average Precision (AP) above 35.0/20.0 for IoU=0.5/0.7. However, after integrating our pose recovery framework, it leads to a dramatic improvement across all methods and distances. This enhancement is especially pronounced in early and late fusion methods, nearly doubling their AP scores at IoU=0.5. Notably, the improvement in the close-range scenarios (0-30m) is even more exciting, with AP@IoU=0.5 scores across all methods exceeding 60.0, and some reaching above 70.0. While the enhancements at mid and long ranges are less substantial, this trend aligns with our earlier analysis regarding the impact of distance on pose recovery accuracy. Nevertheless, the significant boost in AP, particularly in short-range scenarios, is of practical importance, as these are the conditions where high detection accuracy is demanded for safe and effective driving decision-making.

D. Ablation Study

To evaluate the effectiveness of the second-stage box alignment, Fig. 14 presents a comparison of pose recovery accuracy with and without the second-stage alignment. Notably, the exclusion of box alignment results in a marked increase in translation error; the 75th percentile of translation error escalates from to . Interestingly, the rotation error remains relatively stable, even in the absence of box alignment. This observation underscores that the box alignment predominantly contributes to correcting translation errors, which are often introduced by self-motion distortion.

VI. Acknowledge

The work is supported by the National Science Foundation grants CNS-2231519, OAC-2017564, and ECCS-2010332.

VII. Conclusion

In this paper, we introduce *BB-Align*, a lightweight, two-stage pose recovery framework tailored for V2V cooperative perception. Utilizing Bird's-eye View (BV) images and object bounding boxes, the framework accurately estimates the relative pose between two cars while minimizing communication costs. Designed as a non-training-based, plug-and-play module, *BB-Align* integrates seamlessly with existing V2V systems. The method combines Log-Gabor filter-based BV image matching with subsequent object bounding box alignment. Our evaluation on the real-world V2V dataset showed that *BB-Align* outperforms an existing graph matching method [28] and achieves pose errors under and in 80% of cases within a range. Integration into cooperative object detection systems results in a doubling of Average Precision (AP) in severe pose error scenarios. Future work will focus on enhancing the time efficiency of BV image matching.

References

[1] R. Xu, Z. Tu, H. Xiang, W. Shao, B. Zhou, and J. Ma, "Cobevt: Cooperative bird's eye view semantic segmentation with sparse transformers," *arXiv preprint arXiv:2207.02202*, 2022.

[2] R. Xu, H. Xiang, Z. Tu, X. Xia, M.-H. Yang, and J. Ma, "V2x-vit: Vehicle-to-everything cooperative perception with vision transformer," in *European conference on computer vision*, pp. 107–124, Springer, 2022.

[3] N. Vadivelu, M. Ren, J. Tu, J. Wang, and R. Urtasun, "Learning to communicate and correct pose errors," in *Conference on Robot Learning*, pp. 1195–1210, PMLR, 2021.

[4] Y. Yuan, H. Cheng, and M. Sester, "Keypoints-based deep feature fusion for cooperative vehicle detection of autonomous driving," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3054–3061, 2022.

[5] Y. Lu, Q. Li, B. Liu, M. Dianati, C. Feng, S. Chen, and Y. Wang, "Robust collaborative 3d object detection in presence of pose errors," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*.

[6] S. Fischer, F. Šroubek, L. U. Perrinet, R. Redondo, and G. Cristóbal, "Self-invertible 2d log-gabor wavelets," *International Journal of Computer Vision*, vol. 75, no. 2, pp. 231–246, 2007.

[7] R. Xu, X. Xia, J. Li, H. Li, S. Zhang, Z. Tu, Z. Meng, H. Xiang, X. Dong, R. Song, *et al.*, "V2v4real: A real-world large-scale dataset for vehicle-to-vehicle cooperative perception," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13712–13722, 2023.

[8] Y. Han, H. Zhang, H. Li, Y. Jin, C. Lang, and Y. Li, "Collaborative perception in autonomous driving: Methods, datasets and challenges," *arXiv preprint arXiv:2301.06262*, 2023.

[9] Z. Bai, G. Wu, X. Qi, Y. Liu, K. Oguchi, and M. J. Barth, "Infrastructure-based object detection and tracking for cooperative driving automation: A survey," in *2022 IEEE Intelligent Vehicles Symposium (IV)*.

[10] Z. Zhang, S. Wang, Y. Hong, L. Zhou, and Q. Hao, "Distributed dynamic map fusion via federated learning for intelligent networked vehicles," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, May 2021.

[11] Q. Chen, S. Tang, Q. Yang, and S. Fu, "Cooper: Cooperative perception for connected autonomous vehicles based on 3d point clouds," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, (Los Alamitos, CA, USA).

[12] Q. Chen, X. Ma, S. Tang, J. Guo, Q. Yang, and S. Fu, "F-cooper: Feature based cooperative perception for autonomous vehicle edge computing system using 3d point clouds," in *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*.

[13] T.-H. Wang, S. Manivasagam, M. Liang, B. Yang, W. Zeng, and R. Urtasun, "V2vnet: Vehicle-to-vehicle communication for joint perception and prediction," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK*, Springer.

[14] Z. Meng, X. Xia, R. Xu, W. Liu, and J. Ma, "Hydro-3d: Hybrid object detection and tracking for cooperative perception using 3d lidar," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 8, pp. 4069–4080, 2023.

[15] R. Xu, H. Xiang, X. Xia, X. Han, J. Li, and J. Ma, "Opv2v: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 2583–2589, IEEE, 2022.

[16] H. Yu, Y. Luo, M. Shu, Y. Huo, Z. Yang, Y. Shi, Z. Guo, H. Li, X. Hu, J. Yuan, *et al.*, "Dair-v2x: A large-scale dataset for vehicle-infrastructure cooperative 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[17] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 5, pp. 698–700, 1987.

[18] Q.-Y. Zhou, J. Park, and V. Koltun, "Fast global registration," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, Springer.

[19] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp.," in *Robotics: science and systems*, vol. 2, p. 435, Seattle, WA, 2009.

[20] C. Choy, J. Park, and V. Koltun, "Fully convolutional geometric features," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 8958–8966, 2019.

[21] C. Choy, W. Dong, and V. Koltun, "Deep global registration," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2514–2523, 2020.

[22] Z. Qin, H. Yu, C. Wang, Y. Guo, Y. Peng, S. Ilic, D. Hu, and K. Xu, "Geotransformer: Fast and robust point cloud registration with geometric transformer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[23] J. Ma, X. Jiang, A. Fan, J. Jiang, and J. Yan, "Image matching from handcrafted to deep features: A survey," *International Journal of Computer Vision*, vol. 129, pp. 23–79, 2021.

[24] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, 2004.

[25] J. Li, Q. Hu, and M. Ai, "Rift: Multi-modal image matching based on radiation-variation insensitive feature transform," *IEEE Transactions on Image Processing*, 2020.

[26] F. Cao, F. Yan, S. Wang, Y. Zhuang, and W. Wang, "Season-invariant and viewpoint-tolerant lidar place recognition in gps-denied environments," *IEEE Transactions on Industrial Electronics*, 2020.

[27] L. Luo, S. Cao, B. Han, H.-L. Shen, and J. Li, "Bvmatch: Lidar-based place recognition using bird's-eye view images," *IEEE Robotics and Automation Letters*, vol. 6, pp. 6076–6083, 2021.

[28] S. Shi, J. Cui, Z. Jiang, Z. Yan, G. Xing, J. Niu, and Z. Ouyang, "Vips: real-time perception fusion for infrastructure-assisted autonomous driving," in *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, MobiCom '22.

[29] Z. Song, T. Xie, H. Zhang, J. Liu, F. Wen, and J. Li, "A spatial calibration method for robust cooperative perception," *IEEE Robotics and Automation Letters*, 2023.

[30] G. Kim and A. Kim, "Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

[31] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017.

[32] P. Kovesi, "What are log-gabor filters and why are they good?." https://www.peterkovesi.com/matlabfns/PhaseCongruency/Docs/convexpl.html.

[33] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria*, Springer.

[34] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International Conference on Computer Vision*.