Received 1 March 2022; accepted 29 March 2022. Date of publication 8 April 2022;
date of current version 12 May 2022. The review of this article was coordinated by Editor Sumei Sun.

Digital Object Identifier 10.1109/OJVT.2022.3165930

Bridging the Gap Between Point Cloud Registration and Connected Vehicles

HONGYU LI [1], HANSI LIU [1] (Student Member, IEEE), HONGSHENG LU [2], BIN CHENG [2], MARCO GRUTESER [1], AND TAKAYUKI SHIMIZU [2] (Member, IEEE)

[1] Wireless Information Network Laboratory(WINLAB), Rutgers University, North Brunswick Township, NJ 08902 USA
[2] Toyota Motor North America, Inc., R&D InfoTech Labs, Mountain View, CA 94043 USA

CORRESPONDING AUTHOR: HONGYU LI (e-mail: hongyuli@winlab.rutgers.edu)

This work of Hongyu Li was supported at Rutgers by a grant-in-aid from Toyota Motor North America, InfoTech Lab.

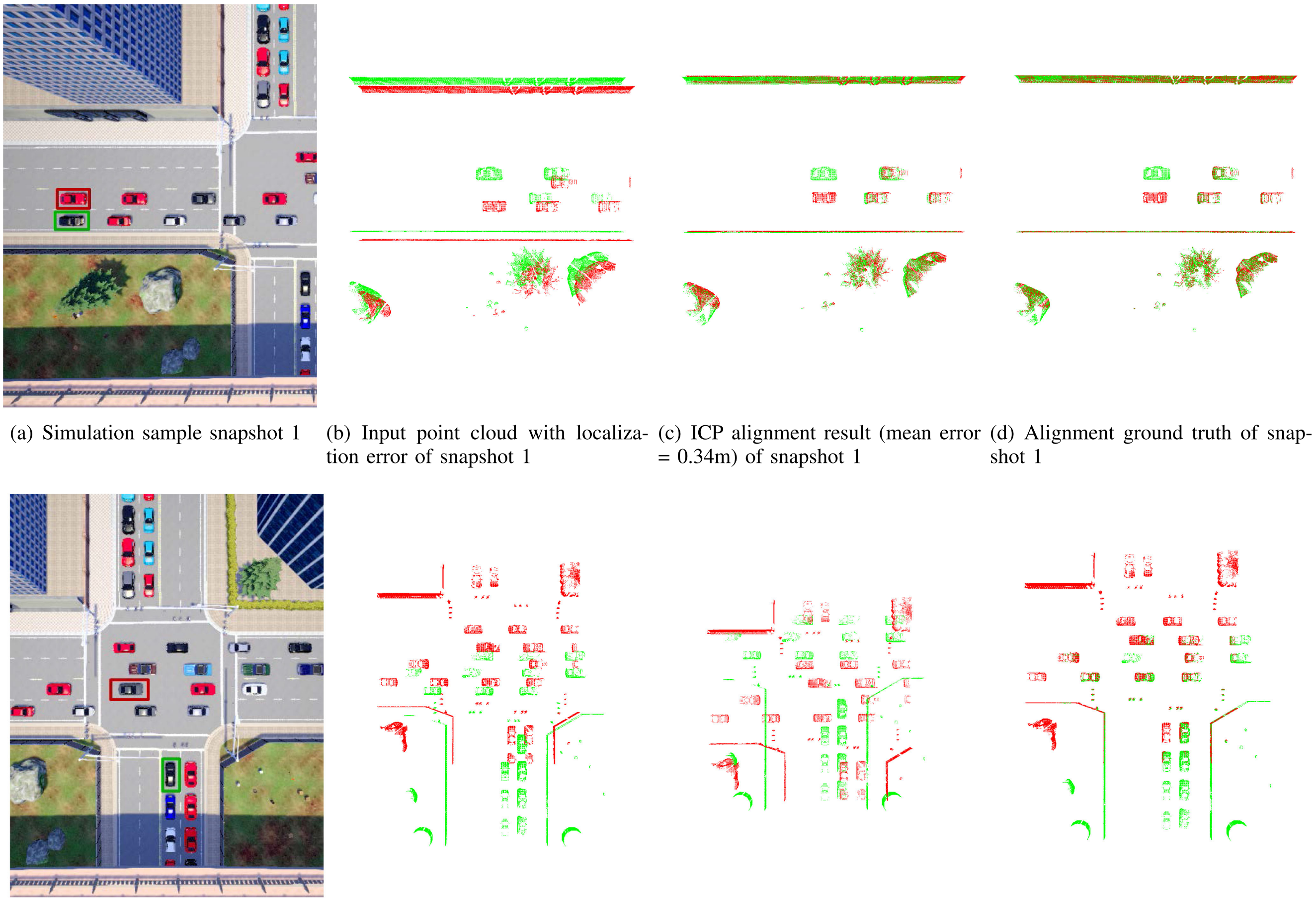
ABSTRACT Connected vehicles can benefit from sharing and merging their observations to develop a more complete understanding of the traffic scene and track traffic participants behind obstructions. Although vehicle-to-vehicle(V2V) communications provide a channel for point cloud data sharing, it is challenging to align point clouds from two vehicles with state-of-the-art techniques due to localization errors, visual obstructions, and differences in perspective. Therefore, we propose a two-phase point cloud registration mechanism to fuse point clouds which focuses on key objects in the scene where the point clouds are most similar and infer the transformation from those. Our system first identifies co-visible objects between vehicle views based on hyper-graph matching using multiple similarity metrics, and then refines the overlap region between co-visible objects across the views for point cloud registration. The system is evaluated based on both experimental and simulation data, which shows tremendous performance improvement when combing with state-of-art baselines.

INDEX TERMS Cooperative perception, Object Detection, Point Cloud Registration, V2X communication.

I. INTRODUCTION

As driving is becoming increasingly automated, vehicles rely on multiple sensors (e.g., ultrasound, cameras, RADAR, or LiDAR) to maintain comprehensive awareness of the surrounding traffic environment. While much progress has been made, it remains challenging to ensure the dependability over the long tail of events and traffic situations that vehicles can encounter. In particular, vehicles must contend with: (i) physical occlusions, in which objects are blocked by others and are only partially observable or unobservable; (ii) sensing limitations, including field of view, resolving power, or lighting conditions that may limit the sensing range and quality. Connected vehicles have the potential to overcome such limitations by sharing observations across a wireless network and merging them across different vehicles, since such physical occlusions and sensing limitations from one perspective can often be easily addressed when viewing the scene from a different perspective.

In this paper, we specifically focus on the fusion of 3D point clouds from different vehicles, which are usually generated by stereo cameras or LiDARs, and broadly used for on-vehicle applications such as object detection, object tracking, etc. The previous work [5] has shown that the object detection accuracy can be improved about 10% for the detection within 20 meters and 30% for longer distances by fusing the point clouds from other viewpoints. In order to benefit from such point cloud fusion in real world, one main challenge is to align point clouds captured by different vehicles. Since the non-negligible vehicle localization error in real system will make the simply merged point clouds even more noisy, point clouds from different vehicles should be well aligned before feeding to applications. Although there has been extensive research on point cloud alignment/registration, the state-of-the-art methods cannot be directly applied to align the pairwise point clouds from vehicles, as they require large overlapping ratio between point cloud pairs [2], [7], [19], [25]. (Note that the scope of this paper is to align point cloud pairs without high definition maps, since they are expensive to create and maintain, and only available for limited areas). Due to occlusions from surrounding objects or vehicles perceiving



(a) Simulation sample snapshot 1 (b) Input point cloud with localization error of snapshot 1 (c) ICP alignment result (mean error = 0.34m) of snapshot 1 (d) Alignment ground truth of snapshot 1

(e) Simulation sample snapshot 2 (f) Input point cloud with localization error of snapshot 2 (g) ICP alignment result (mean error = 13.21m) of snapshot 2 (h) Alignment ground truth of snapshot 2

FIGURE 1. Illustration of ICP point cloud registration performance in bird's eye view.

the scene from different directions, the observations from different vehicles usually have little overlap ratio and fail to be aligned by the state-of-the-art point cloud registration algorithms. Considering the most widely used point cloud registration algorithm, Iterative Closet Point (ICP) [2], as an example, the alignment results are shown in Fig. 1. When vehicles are close to each other and driving towards the same direction as shown in Fig. 1(a) marked with red and green rectangles, ICP can align the point clouds from these two vehicles when decimeter level localization error is introduced, as (i) the inputs combined with localization error could still be considered as a good initialization and (ii) there are large enough overlapping ratio. However, as for the scene in Fig. 1(e), the overlapping ratio will become much lower since vehicles are driving towards different direction and there are objects in between. Thus, ICP fails to fuse the point clouds accurately as shown in Fig. 1(g). The requirement of the overlapping ratio for the state-of-the-art point cloud registration methods largely restricts the potential peer vehicles which can benefit from the vehicle networks based point cloud sharing.

To overcome such limitations, we design a two-phase point cloud alignment system that can fuse point cloud accurately even when vehicles have large viewpoint difference. Our intuition is to detect the overlapping region between two views and only align point clouds based on that, so that the overlap ratio of input point clouds could be largely increased. Specifically, the system first identifies and matches co-visible objects, that is objects visible from both perspectives, using hyper-graph matching based on the extracted location and label information. It then estimates the co-visible region for each pair of co-visible objects and cropped out the larger overlap region. The selected co-visible areas will act as anchor regions and their point cloud will be used to estimate the transformation between two vehicles. The estimated transformation will then be applied to the entire point cloud from the same viewpoint. We evaluate the accuracy of point cloud registration and co-visible matching based on both real-world KITTI [16], [17] dataset and synthetic CARLA [11] datasets. Our contribution can be summarized as followed:

- We propose the first system which can accurately align point cloud pairs under complex traffic conditions, such as scenes with various occlusions and large view angle difference[1] (e.g., 90°, 180°).

- We introduce a technique to identify co-visible objects by combining multiple similarity metrics obtained in 3D object detection results to distinguish co-visible objects from single-visible objects.
- We show that fusion accuracy is improved when point cloud registration is focused on the co-visible object with the overlapping area among the views.
- We evaluate our system based on both synthetic scenes and real-world experimental data across highway and intersection scenarios and show that it can improve point cloud registration algorithms with a significant margin.

Note that this paper is not proposing new point cloud registration algorithms, but enable existing ones to be applicable in complex traffic scenes, which further activates the potential of vehicle network based data sharing.

II. RELATED WORK

As our work lies at the intersection of point cloud registration and vehicle information fusion, the related work in these two areas is summarized in this section.

The term "point cloud" refers to a set of data points in 3D space which are usually used to represent objects or scenes. Since point clouds are generally produced by depth-capable sensors such as LiDAR [45] or RGBD cameras [32] with a partial view of a scene, two or more partially overlapping point clouds are often combined to represent the full 3D geometry of the sensing region. This process of finding a translation and rotation transformation of one point cloud so that the overlapping portion matches that of another point cloud is called point cloud registration or alignment. Note that, the terminology **point cloud registration or alignment** in this paper refers to the specific algorithms to match the point clouds, and **point cloud fusion** refers the complete pipeline of combing point clouds from a system perspective, including prepossessing, sharing, and registration or alignment.

A. POINT CLOUD REGISTRATION

1) PAIRWISE POINT CLOUD REGISTRATION

Generally, there are two popular paradigms for point cloud registration: *correspondence*-based methods and *correspondence-free* methods, depending on whether correspondences between point clouds are extracted explicitly.

Correspondence-based methods first detect and match 3D keypoints across point clouds and then infer the transformation from these putative correspondences. Since it is too inaccurate to match points based on position alone, the matching process is based on features the describe the shape surrounding a point. Traditional hand-crafted features commonly summarize pairwise or higher-order relationships in histograms such as Fast Point Feature Histograms [36], Viewpoint Feature Histograms [37], or Clustered Viewpoint Feature Histograms [1]. With the development of deep learning, a number of neural network based feature descriptors have been proposed, such as PointNet [34], 3DMatch [48], PPFNet [10], 3DSmoothNet [19], Multi-view Descriptor [25], FCGF [8] etc. Although the robustness of the learned 3D descriptors is improved compared to the hand-crafted features, their registration pipelines still rely on the same process of matching geometric features across point clouds. All these methods require significant overlap between two point clouds to accurately combine them. For example, the evaluation of these methods, such as [10], [19], [25], require the input point cloud pair to overlap by at least 30%, which is not necessarily the case when vehicles approach an intersection from different directions as in Fig. 1.

Iterative Closest Point (ICP) [2] and its variants, e.g., point-to-plane ICP [27], point-to-line ICP [4] and Generalized-ICP [39], are the most commonly used correspondence-free methods. These algorithms perform optimization by iteratively refining a point correspondence and the associated rotation from an assumed starting correspondence, but they are not robust against outliers and converge to a global optimum only when starting with a reasonable initial alignment. To overcome this, correspondence-based methods can be used before ICP to provide the coarse alignment. To remove the need for initial alignment, recent work either integrates such two stage registrations into an end-to-end learning algorithm [7], [28], [44] or proposes non-training global registration pipelines [3], [47], [49]. Moreover, NDT [29] represents point clouds by a combination of normal distributions to apply standard numerical optimization, and TEASER [46] reformulate the registration problem using a truncated least squares to yield a fast computation and provides readily checkable conditions to verify if the returned solution is optimal. Although the applicability of pairwise registration methods are extended, they still cannot work for vehicle point cloud registration directly due to the high outlier ratio caused by distinct vehicle views and occlusions.

2) SCENE-BASED OPTIMIZATION

The aforementioned methods can register point clouds in a pairwise manner, but the ambiguous cases that arise in pairwise matching can be mitigated by incorporating cues from multiple views. Projects such as [18], [40], [41] posed the task of finding a global alignment as picking the best candidates from a set of putative pairwise registrations, such that they satisfy the loop constraints. However, such approaches are less desirable for vehicle point cloud generation since they require the presence of a larger numbers of neighbouring vehicles that share their point cloud in order to perform single registration. Similar to our setting, the most relevant work to ours is arguably [12], which matches and aligns point clouds in different LiDAR scans. It can recover the correct alignment over larger vehicle displacements, when vehicles are traveling

[1] Defined as the bearing difference between two vehicles.

in the same direction, but not the intersection scenario we consider. Finally, although [15], [31] aim to register vehicle point clouds at the object level, these methods assume a complete point cloud of the surroundings from a high-resolution 3D map as inputs. Given the substantial overhead of maintaining such maps, we aim for a map-free solution.

Baselines: In order to illustrate our contribution, 4 baseline algorithms are chosen as the benchmark for point cloud performance comparison, including (1) the most widely used correspondence-free registration algorithm ICP [2], (2) the Generalized-ICP [39], which is more robust for incorrect correspondences compared with ICP, (3) a deep learning based feature descriptor FCGF [8] which is one of the most recent progress on correspondence based registration algorithm, and (4) the SSM [12], which is the closest work to ours in terms of processing pipeline. Note that, we also try to consider TEASER [46] as the baseline. But its estimated inliers are usually incorrect, or the number of inliers is less than specified lower bound, which results matching failure for most cases in our datasets.

B. VEHICLE-TO-VEHICLE INFORMATION FUSION

Other pioneering work shows the potential benefit of vehicle information fusion, but did not consider the various vehicle viewpoint differences and localization errors at intersection scenarios as we do. [42] proposed to fuse vehicle information for perception, but focused on fusing compressed LiDAR features. Although [23] studies the full stack of multi-vehicle cooperative perception and driving, the design and implementation are limited to when vehicles are following each other. [35] proposed a system to share vehicle's view through extracted features using SLAM [14]. However, the focus of their proposed system is on visualizing and reconstructing the shared camera view. Both [5] and [43] explored the benefits of point cloud fusion, but no localization error was involved in the pipeline, which always leads to perfect point cloud fusion.

III. SYSTEM OVERVIEW

Based on vehicular application scenarios and foregoing review of the state-of-the-art in point cloud registration algorithms we identify the following challenges for point cloud fusion across vehicles:

- Aligning point clouds in complex traffic situations. Complex traffic scenes challenge the point clouds registration between vehicles in two aspects: (i) the presence of multiple traffic participants leads to participants experiencing different occlusions in their observations. This significantly decreases the amount of overlap between point clouds from different vehicle; (ii) the same object can be observed by vehicles from very different view angles, for example when approaching from different legs of an intersection. The resultant observations can thus contain the same object observed from different sides, which again leads to relatively distinct point clouds with little overlap.
- Limit bandwidth consumption. The system should be able to exchange any required data over a wireless network between vehicles. While this information does not necessarily have to be exchanged over very bandwidth-limited technologies such as Dedicated Short Range Communications (DSRC), the system should be able to exchange the data over emerging technologies such as millimeter-wave (mmWave) communications.
- Tolerate vehicle localization errors. Vehicle localization errors will affect the transformation from vehicle sensor coordinates to world coordinates, which is based on vehicle locations. Although increasing sensor capabilities and improved localization lead to more accurate vehicle localization, the system should still be able to handle localization errors at least at the decimeter level.

In order to merge vehicle point clouds and meet the design objectives, we propose a two-phase point cloud fusion system which first identifies objects that are co-visible from each vehicle's perspective and then refine the point clouds based on co-visible region of these objects to align the point clouds. The system is designed based on the outputs of 3D object detection, since it is usually available when the vehicle has depth sensing capability and the robust 3D object detection results provide reliable hints during point cloud fusion. As shown in Fig. 2, the input of the system based on the 3D object detection results include the labels, centers, and point clouds of detected objects. The coordinates of inputs are transformed into world coordinates based on each vehicle's own localization. The *Co-visible Object Detection* module extracts multiple similarity metrics based on the detected object labels and centers to distinguish co-visible objects from those that are visible only from a single perspective. Even though co-visible objects can be observed from two vehicles' viewpoints, they may not have enough overlapping visible area to yield an accurate point cloud registration. Thus, *Co-visible Region Refinement* further trim the point cloud to keep the overlapping area of the co-visible objects for transformation estimation between two views.

The resulting aligned point clouds can be combined to create a more complete representation of the traffic scene, which better supports advanced driving assistance applications. Note that not all pairs of vehicle point clouds can be fused in our system, only the ones include co-visible objects and co-visible areas can be fused by *Co-visible Object Detection* and *Co-visible Region Refinement*, respectively. If fusion is not possible, vehicles can fall back on their individual perception. The detailed fusion requirements of each module will be discussed in Sections IV and V.

Note that our system design limits bandwidth consumption since it only needs abstract information to determine whether point cloud pairs can be potentially aligned, and then requests the raw point cloud to estimate the transformation. Specifically, the input data volume of our fusion system to determine the eligibility of point cloud alignment is in the order of kilobytes, which includes the label, center and visible region of each detected objects. Such information can be shared

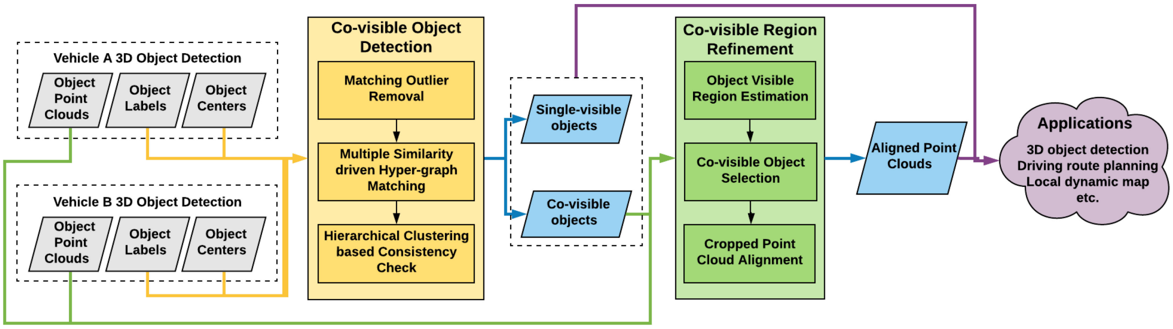

Vehicle A 3D Object Detection
Object Point Clouds
Object Labels
Object Centers
Vehicle B 3D Object Detection
Object Point Clouds
Object Labels
Object Centers
Co-visible Object Detection
Matching Outlier Removal
Multiple Similarity driven Hyper-graph Matching
Hierarchical Clustering based Consistency Check
Single-visible objects
Co-visible objects
Co-visible Region Refinement
Object Visible Region Estimation
Co-visible Object Selection
Cropped Point Cloud Alignment
Aligned Point Clouds
Applications
3D object detection
Driving route planning
Local dynamic map
etc.

FIGURE 2. System Design.

through messages transmitted using periodic V2V communications, e.g., Cooperative Perception Messaging transmitted via DSRC. With varied number of objects detected in the vehicle's view, the raw point cloud of detected objects could be as large as in the order of a few megabytes, which could be transmitted via large-bandwidth mmWave communications when needed.

IV. CO-VISIBLE OBJECTS DETECTION

Co-visible objects detection module is designed to identify co-visible objects. Specifically, we formulate the problem of detecting co-visible objects as a hyper-graph matching problem [24], which is broadly used in computer vision for key points correspondence determination. Since the single-visible objects are considered outliers in graph matching, we first coarsely filter out the single-visible objects using a threshold-based filtering. Based on the remaining objects in two views, different similarity metrics will be extracted for hyper-graph matching to take advantage of the label and location information obtained from object detection. As hyper-graph matching can pair objects but not distinguish single visible objects, we design a distance consistency check based on hierarchical clustering to identify the correctly matched co-visible objects. The illustration of co-visible objects detection is shown in Fig. 3. After outlier removal, the detected objects are plotted, where different color represents objects detected from different view and different shape indicates different object labels. Hyper-graph matching is able to generate the matching between objects, and the consistency check will further extract the correct matching based on the matched pairs.

A. PRELIMINARY

Hyper-graph matching was originally proposed to match correspondences between images and can be solved efficiently through reweighted random walk [24]. A hyper-graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ consists of nodes $v \in \mathcal{V}$, hyper-edges $e \in \mathcal{E}$ as well as the attributes $a \in \mathcal{A}$ associated with the hyper-edges. A hyper-edge e encloses a subset of nodes with size $\delta(e)$ from $\mathcal{V}$, where $\delta(e)$ denotes the order of each hyper-edge. The goal of hyper graph matching is to establish the mapping between nodes of two graphs $\mathcal{G}^P = (\mathcal{V}^P, \mathcal{E}^P, \mathcal{A}^P)$ and $\mathcal{G}^Q = (\mathcal{V}^Q, \mathcal{E}^Q, \mathcal{A}^Q)$.

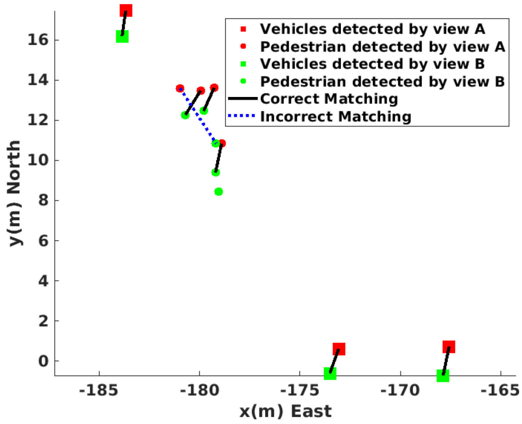


FIGURE 3. Object matching illustration.

Suppose a set of all possible node correspondences $\mathcal{C} = \mathcal{V}^P \times \mathcal{V}^Q$, and k tuples $c_{\omega_1} = (v^P_{p_1}, v^Q_{q_1}), \ldots, c_{\omega_k} = (v^P_{p_k}, v^Q_{q_k}) \in \mathcal{C}$ among them. For kth order hyper graph matching, the similarities of the k-tuples can be measured by comparing attributes of two kth order hyper-edge $e^P_{p_1,\ldots,p_k}$ and $e^Q_{q_1,\ldots,q_k}$, which means the hyper-edges connecting $v^P_{p_1,\ldots,p_k}$ and $v^Q_{q_1,\ldots,q_k}$ respectively. Denoting the kth order similarity function by Ω, the kth order similarity of the k-tuple can be measured by $\Omega(a^P_{p_1,\ldots,p_k}, a^Q_{q_1,\ldots,q_k})$. Therefore, the affinity tensor including kth order similarities can be generalized in a recursive manner as follows:

$$\mathbf{H}^{(k)}_{\omega_1,\ldots,\omega_k} = \boldsymbol{\Omega}_k(a^P_{p_1,\ldots,p_k}, a^Q_{q_1,\ldots,q_k}) + \lambda^{(k-1)}\sum_{l=1}^{k}\mathbf{H}^{(k-1)}_{\omega_1,\ldots,\omega_k\omega_l}$$

$$\mathbf{H}^{(1)}_{\omega_i} = \boldsymbol{\Omega}(a^P_{p_i}, a^Q_{q_i}) \tag{1}$$

where $\lambda^{(k)}$ represents the weighting factor of kth order similarity value and the superscript on $\mathbf{H}$ denotes the dimension of a tensor. Therefore, the object function of the hyper-graph matching can be formulated to equation 2, where $\mathbf{X}$ is a binary assignment matrix, m^P and n^Q denote the number of nodes in $\mathcal{G}^P$ and $\mathcal{G}^Q$, $\mathbf{1}^P_m$ and $\mathbf{1}^Q_n$ represent all-ones vector with size m and n respectively. By maximizing the matching score of objective function under the one-to-one constraints, the hyper-graph matching problem can be solved by the Hungarian method [30] to find the assignment matrix $\mathbf{X}^*$.

$$\mathbf{X}^* = \text{argmax}_{\mathbf{X}} \mathbf{H}(k) \otimes \mathbf{X}$$
$$\text{s.t.}\ \mathbf{X}\mathbf{1}_{n^{\mathcal{Q}}\times 1} \leq \mathbf{1}_{m^P\times 1},\ \mathbf{X}^{\mathbf{T}}\mathbf{1}_{m^P\times 1} \leq \mathbf{1}_{n^{\mathcal{Q}}\times 1} \qquad (2)$$

B. MATCHING OUTLIER REMOVAL

To improve the object matching accuracy, our proposed system first removes matching outliers. In the hyper-graph matching task, matching outliers refer to the nodes which only consist in one graph and can not be matched. In our vehicle view matching context, matching outliers are the single-visible objects. As the increasing of overlapping region between vehicle's view, there will be more single-visible objects involved each view. Therefore, removing such single-visible objects will reduce the number of outliers and increase object matching accuracy. At current stage, the single-visible can be coarsely excluded based on nearest neighbour search. As all the objects from two views are transformed into the same world coordinates, if a object does not have any neighbours in the other view within a threshold distance, the object can be classified as single-visible object and removed before matching. The distance threshold can be determined based on the localization accuracy, such as the accuracy value provided by Android Location API [20], which indicates horizontal accuracy in meters as the radius of 68% confidence.

C. HYPER-GRAPH MATCHING WITH MULTIPLE SIMILARITY MEASURES

Inspired by the existing work [6], [21], [24], our work extends the hyper-graph matching by combing multiple similarity measures, including attribute-based similarity measures, geometry-based similarity measures, etc., in matching.

Each vehicle can first build a hyper-graph based on its locally detected objects, and additionally, it can also build a hyper-graph for a remote vehicle based on the shared information from that vehicle. In the built hyper-graphs, nodes denote detected objects by the observing vehicle and edges represent the spatial relationship between detected objects. The attributes of each node, such as the category the object belongs to, the size of the object, etc., can be utilized to distinct one node from others. In our work, we focus on exploiting the category information of objects since it is invariant under different viewpoints.

$$\mathbf{H}^{(1)}_{\omega_1,\omega_2,\omega_3} = exp\left[-\frac{1}{\sigma_{s^1}}\sum_{k=1}^{3} |\sin(\theta^P_{\omega_k}) - \sin(\theta^Q_{\omega_k})|\right] \qquad (3)$$

$$\mathbf{H}^{(2)}_{\omega_1,\omega_2,\omega_3} = exp\left[-\frac{1}{\sigma_{s^2}}\sum_{\substack{i,j\in\{1,2,3\}\\ i\neq j}} |d^P_{\omega_i\omega_j} - d^Q_{\omega_i\omega_j}|^2\right] \qquad (4)$$

$$\mathbf{H}^{(3)}_{\omega_1,\omega_2,\omega_3} = \frac{1}{3}\sum_{k=1}^{3} \text{diff}(l^P_{\omega_k} - l^Q_{\omega_k})^{\sigma_{s^3}} \qquad (5)$$

In order to qualify the hyper graph similarity, we specifically extract the angle, distance and label similarities based on hyper edges. The angle similarity [13] is defined in equation 3 based on a pair of 3 rd order hyper-edges, $e^P_{a,b,c} \in \mathcal{E}^P$ and $e^Q_{x,y,z} \in \mathcal{E}^Q$, $(a, b, c \in \mathcal{V}^{\mathcal{P}}\ a \neq b \neq c$ and $x, y, z \in \mathcal{V}^{\mathcal{Q}}\ x \neq y \neq z)$, where $\theta^P_{\omega_k}$ and $\theta^Q_{\omega_k}$ denote the angles in the triangle pairs formed by the correspondence ω_k in $\mathcal{P}$ and $\mathcal{Q}$. The distance similarity [24] is quantified based on equation 4, where $d^P_{\omega_i\omega_j}$ and $d^Q_{\omega_i\omega_j}$ represent the length of edges formed by the nodes within the hyper-edge. σ_{s^1} and σ_{s^2} are scale factors, which are set empirically to 0.5 and 0.15 as in [24].

To take advantage the label information predicted by vehicle object detector, we define the label similarity $\mathbf{H}^{(3)}_{\omega_1,\omega_2,\omega_3}$ in equation 5, where $l^P_{\omega_k}$ and $l^Q_{\omega_k}$ are the labels of the correspondence. diff function is designed to output value ranging from 0 to 1, which 1 indicates labels of the correspondence are completely the same, and 0 means completely different. Depending on the representation of the shared label from vehicles, diff function can be implemented in various ways. If only the final predicted label of each object is available, then diff function can be implemented as piecewise function, where same labels outputs 1 and different label outputs 0. If the predicted confidence vector across all categories are available, then diff function can be computed based on the cross-entropy of the two confidence vectors. The scale factor σ_{s^3} is set to 3 empirically in our implementation. Although the distance and label similarity can be implemented based on 2 rd order and 1st order edge respectively, we define them based on 3 rd order here for easier probability combination.

$$\mathbf{H}_{\omega_1,\omega_2,\omega_3} = \left[\lambda^{(1)}\mathbf{H}^{(1)} + \lambda^{(2)}\mathbf{H}^{(2)}\right]\mathbf{H}^{(3)} \qquad (6)$$

To merge the three similarity metrics, we combine them as defined in equation 6, where the subscript of $\mathbf{H}^{(1)}$ $\mathbf{H}^{(2)}$ $\mathbf{H}^{(3)}$ are omitted as they share the same subscript as defined in equation 3,4,5. Instead of linearly adding all the metrics as generalized in [24], we propose to use the label similarity as a conditional probability. It is because not only the correct correspondence in label similarity can generate higher values, the incorrect correspondence happened to have same labels will also produce higher value. Therefore, linearly combining the label similarity will actually increase the overall similarity score for incorrect matching, which yields lower matching accuracy. Using the label similarity as a conditional probability can benefit the hyper-graph matching because the overall similarity score will only be higher if the correspondence have matched labels. $\lambda^{(1)}$ and $\lambda^{(2)}$ are the weights for linear combining the angle and edge length similarity, and we set both to 0.5 for equal weights assignment in our implementation.

D. HIERARCHICAL CLUSTERING BASED CONSISTENCY CHECK

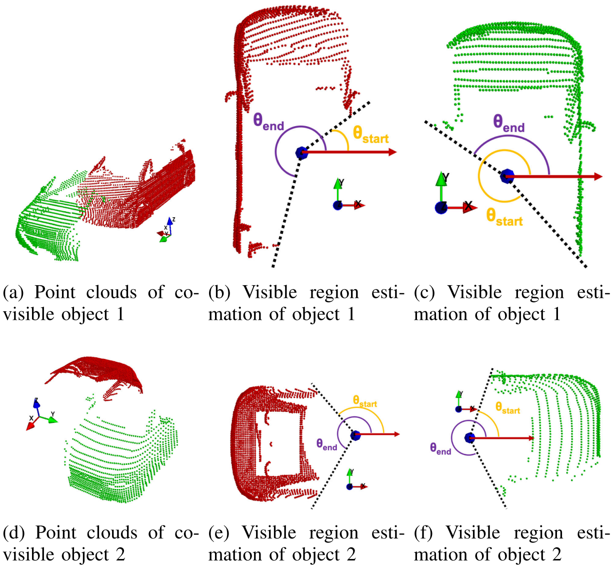
Since hyper-graph matching only assigns matched objects between views but can not distinguish co-visible objects from single-visible objects, we propose a hierarchical clustering based consistency check to extract co-visible objects from hyper-graph matching results.

For detected objects in two observations, the distance between a pair of matched objects can be computed according to the euclidean distance between the centers of objects in the world coordinate system. If the matching is correct, then the distance between the pair of objects consists two parts: (i) the localization error and (ii) the error from inaccurate 3D object detection. As the state-of-the-art algorithms can archive 81.43% accuracy for vehicle 3D detection based on 0.7 IoU threshold [33] but the localization error is still as high as meter-level in the urban area, it is reasonable to infer the localization error is the major component of the distance between matched pairs. Since the objects detected by the same vehicle shares the same localization error, the pairwise distance between the correct matched objects should share such same component in distance, which is the combination of localization error of two vehicles. But the distance between incorrect matched objects maybe largely diverse. Although the direction of matched pairs have similar characteristics, it is not resilient to the object detection error as the distance between matched pairs. Small errors in object center location may cause large direction error of matched co-visible object pairs.

Based on the analysis that the pairwise distance between correct matched objects should be relative consistent, clustering method can be applied on graph matching results to extract the correct matched co-visible objects. Specifically, we perform the hierarchical clustering on the hyper graph matching outputs, and classify the objects cluster with consistent pairwise distance as co-visible objects and the others as single-visible objects. The threshold distance variance in hierarchical clustering to select the cluster can be determined based on the 3D object detection performance, because it mainly comes from the 3D object detection error. In order to increase the precision of co-visible objects detection, the number of co-visible objects, namely the number of objects within the selected cluster, should be at least three to produce the final output. Otherwise, the system will ignore the information and does not perform fusion based on received data. Such design will make sure the system only fuse the information when it has enough confidence to do so.

V. CO-VISIBLE REGION REFINEMENT

Although the module of co-visible objects detection can identify single-visible objects and match co-visible objects, it remains challenging to perform point cloud registration accurately. It is because that the matched co-visible objects may not have enough common seen area due to the large view-difference and occlusions. For example, in Fig. 4, objects are observed by two different viewpoints and the resulting point clouds are colored by red and green, respectively. Although the two point clouds in Fig. 4(d) refer to the same co-visible object, no overlapping area exists between them. On the contrary, the two points in Fig. 4(a) shows a clear overlapping area of the a co-visible object. As discussed in Section II, the overlapping region between the two point clouds is essential for correct point clouds alignment. In order to address the challenge of lack of overlapping area, the Co-Visible Region Refinement is designed to first, among all detected co-visible objects, quantify the overlap region of the co-visible objects from two vehicle's views and then align point cloud based on cropped co-visible point clouds.



(a) Point clouds of co-visible object 1 (b) Visible region estimation of object 1 (c) Visible region estimation of object 1

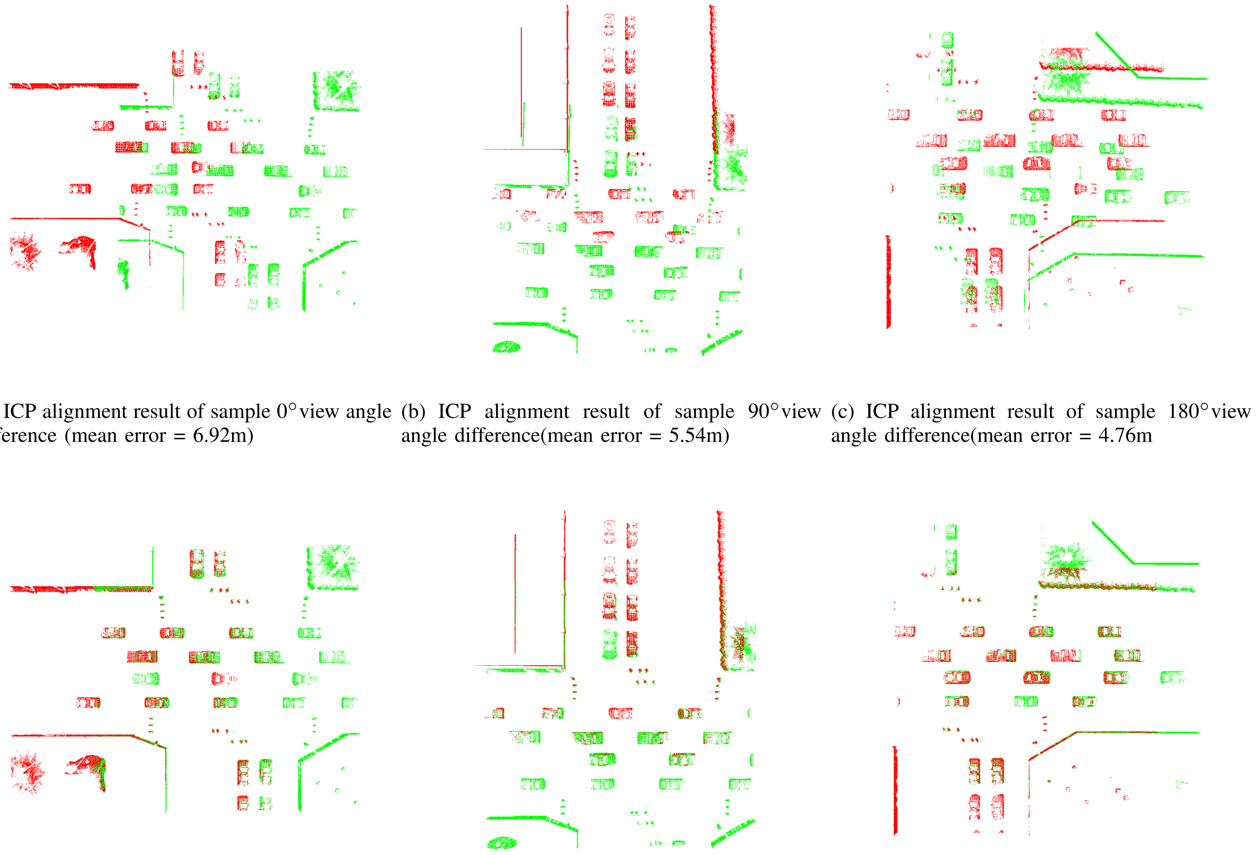
(d) Point clouds of co-visible object 2 (e) Visible region estimation of object 2 (f) Visible region estimation of object 2

FIGURE 4. **Object Visible Region Estimation.**

A. OBJECT VISIBLE REGION ESTIMATION

In order to identify the overlap region of co-visible objects, the visible region of co-visible objects needs to estimate based on each vehicle's viewpoint. To address such challenge, we propose to quantify the object visible region approximately from the bird's eye view based on the relative location between detected object center and corresponding point cloud of the object. As point clouds are generated by sensors with depth sensing capability, e.g. stereo cameras and LIDARs, such depth sensing capability is compliant with the line-of-sight rule, which can not see through objects and only sense line of sight object regions. Therefore, the point cloud generated by these sensors must locates between the detected object's center and the observer vehicle. Inspired by these characteristics, we approximate the object visible region by estimating the angle of the point cloud's coverage region with respect to the object's center. By projecting the point clouds to a bird's eye view, the coverage region of a point cloud can be represented as $\theta^i = [\theta^i_{start}, \theta^i_{end}]$, where θ^i_{start} and θ^i_{end} are the starting and ending angle of the point cloud coverage area for object i. Both θ^i_{start} and θ^i_{end} are computed according to the same axis such as west to east. Thus, the point cloud coverage can be quantified based on the counter-clockwise circular angle difference from θ^i_{start} to θ^i_{end}. The demonstration of visible region estimation are shown in Fig. 4(b), (c), (e), (f).

The points which are closer to object center or reflected by vehicle roof will be noisy to estimate object visible region. Thus, a prepossessing can be applied to improve the robustness by filtering out such as points based on distance threshold or surface detection.



(a) ICP alignment result of sample 0°view angle difference (mean error = 6.92m)

(b) ICP alignment result of sample 90°view angle difference(mean error = 5.54m)

(c) ICP alignment result of sample 180°view angle difference(mean error = 4.76m

(d) Our alignment result of sample 0°view angle difference(mean error = 0.03m)

(e) Our alignment result of sample 90°view angle difference(mean error = 0.02m)

(f) Our alignment result of sample 180°view angle difference(mean error = 0.05m)

FIGURE 5. **Sample alignment results with 0°, 90°, 180° view angle difference in Carla simulation.**

B. CO-VISIBLE OBJECT SELECTION

Although co-visible objects can be identified based on object matching, there is no guarantee that the observation of co-visible objects from each vehicle's view will have overlapping area. Given the fact that, the state-of-the-art point cloud registration methods require the overlapping area between point clouds to estimate the transformation. The co-visible objects which include no or little overlapping area is generally not suitable for point cloud registration.

Based on this observation, we propose to measure the intersection area based on the overlapping of point cloud coverage angle which can be defined as $|\ \text{intersect}(\theta^i, \theta^j)\ |$. In general, the intersection between two point cloud coverage angle indicates the overlapping area between two point clouds. For example, the estimated object visible region shown in Fig. 4(e) and (f) don't have any overlapping area since the intersection of them is zero. However, the visible region shown in Fig. 4(b) and (c) shows the intersection angle between two point clouds is around 120 °, which can be potentially used for point cloud registration.

Generally, larger overlapping area between point clouds will have better registration performance. In order to improve to the point cloud registration accuracy, we propose to examine the visible region of each pair of detected co-visible objects, and only keep the point clouds for the pairs whose visible region is larger than a threshold. If there are more than one of such pair is found, the system will further to crop and align the point clouds. However, the system will reject to align the point cloud pair if the intersection visible region of all co-visible objects is smaller the threshold and only align the point cloud based on minimizing the distance between co-visible object centers. In our implementation, the threshold is set to 30° as it is commonly required for state-of-art point cloud registration algorithms.

C. CROPPED POINT CLOUD ALIGNMENT

Based on the selected co-visible objects, point cloud registration can be applied to align two point clouds. In order to improve the robustness of point cloud registration, we propose to crop the point cloud based on the intersection of visible regions. Specifically, only the points within the intersection region intersect(θ^i, θ^j) will be used for point cloud registration. Such process will remove outliers and increase the inlier ratio for point cloud registration. Eventually, the transformation estimated based on the selected co-visible object will be applied to the whole point cloud captured by the sharing vehicle. General point cloud registration algorithms can be used here for transformation estimation, such as ICP.

VI. EXPERIMENT SETUP AND IMPLEMENTATION

In order to explore the system performance, we construct three datasets, including an experimental dataset extracted from KITTI and two synthetic datasets generated by the CARLA [11] and SUMO [26] simulators. Additionally, we also implemented 3D object detection and three baseline algorithms.

A. KITTI BASED EXPERIMENTAL DATA

The KITTI [16], [17] dataset includes detailed information of a single autonomous vehicle travelling through a wide range of road scenarios. It contains a trove of sensor readings from a variety of sensor modalities such as high resolution color and grayscale stereo cameras, a Velodyne 3D laser scanner and a GPS/IMU inertial navigation system. For this work, however, the KITTI dataset is not directly applicable since we need two sets of point clouds captured at different perspectives of the same scene.

We address this limitation by leveraging Lidar point clouds obtained at different timestamps of the same route from the vehicle, to imitate two cars travelling by following each other. (There are no eligible cases found to imitate cars facing each other.) More specifically, we examined KITTI's 3D object detection dataset [17] and identified a plethora of time-instance pairs where more than three same street object are detected at both scenes and vehicles are traveled more than 4 meters based on GPS. We removed pairs where the transformation between object pairs are not consistent to filter out inaccurate ground truth labeling and moving street objects. This process ended up generating 668 pairs of different time instances that satisfied the requirement of our fusion pipeline, which includes at least 3 pairs of co-visible objects and at least one of co-visible objects has more than 30°overlapping region. As our system built based on the KITTI 3D object detection task, which can only be performed on the front view camera, the input point clouds in this experiment are limited to the LIDAR points in the front view. The evaluated based on this dataset will be referred as **KITTI** in the following sections.

B. CARLA BASED SYNTHETIC DATA

Since there is no labeled large view angle difference dataset available, we use CARLA [11] to render realistic intersection scenes, which provides open digital assets (urban layouts, buildings, vehicles) and supports flexible specification of sensor suites. Based on CARLA's builtin map and navigation APIs, we created a four-legged perpendicular intersection simulation, which will be introduced in section VI-B1 and referred as **CARLA**. In order to further evaluate the system with more realistic urban traffic pattern, a four-legged skewed intersection is co-simulated by CARLA and SUMO, where CARLA is mainly used to render the scene and generate sensor information and SUMO is used for urban traffic mobility simulation. The design of the co-simulation is described in section VI-B2 and referred as **SUMO** in following sections.

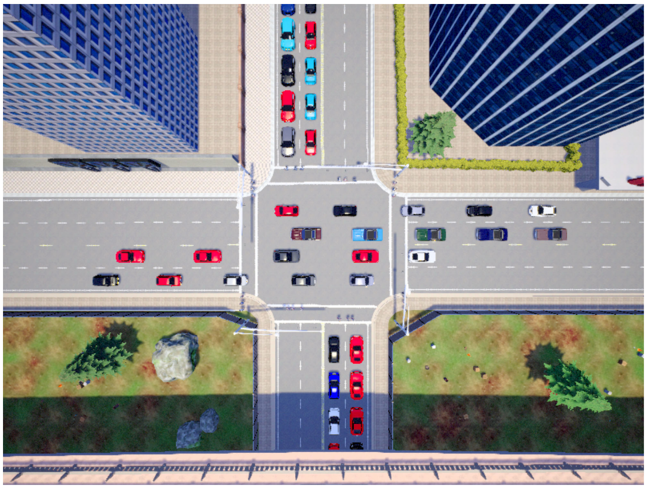


FIGURE 6. Bird's eye view sample snapshot of CARLA intersection simulation.

1) CARLA SOLO SIMULATION

The CARLA solo intersection scenario is rendered in Town 5 of CALRA builtin map, includes 4 directions and each direction with two lanes. A bird's eye view sample snapshot of the simulation is shown in Fig. 6. Each lane has 5 vehicles, which are set to be the same model to avoid rear-ended collisions, since vehicles are controlled based on throttle in CALRA and different models may have different acceleration based on same throttle value. But the outlook color of each set of 5 vehicles are different and randomly picked. In addition, 6 pedestrians are considered on 4 corners of the intersection in groups of three, which are standing in line and trying to cross the intersection if there is not conflicting traffic. Overall, the intersection scenario includes 40 vehicles and 24 pedestrians in total.

For each vehicle, 4 cameras are mounted on the center top of vehicles' roofs with the height as 2m from the ground to cover the full 360°field of view. In order to generate dense point clouds, we use 'depth camera' in CARLA to obtain the pixel depth, which is the ground-truth pixel range perfectly aligned with the corresponding RGB image. Besides, the position and bounding box of vehicles, and the pose transform of cameras are also logged. In order to simulate the unstable GPS reading, we randomly sample localization error from a Gaussian Distribution $X \sim \mathcal{N}(0, 1)$ with zero mean and 1 meter standard deviation. Among the process of vehicles and pedestrians completed cross the intersection (positioned on the other side of intersection), we evenly take 16 snapshots

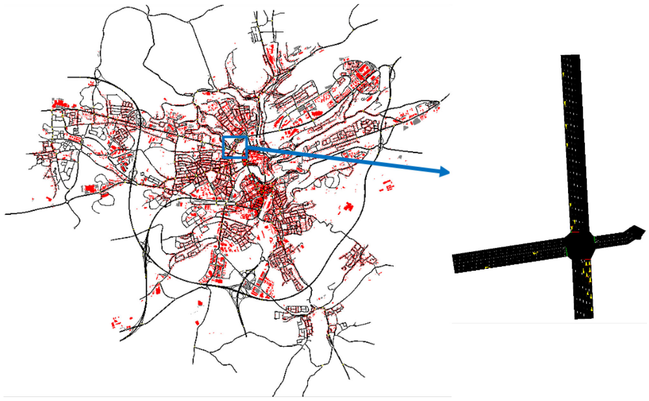


FIGURE 7. CARLA-SUMO co-simulation intersection.

with 1 second interval. Applying the same filtering of KITTI dataset, 3228 pairs of vehicle observations meets our fusion system requirement and are extracted for evaluation.

2) CARLA-SUMO CO-SIMULATION

SUMO [26] is a traffic simulation package designed to handle large road networks, which allows for intermodal simulation and comes with a large set of tools for scenario creation. To generate realistic traffic mobility, we built the simulation based on [9], which simulates the realistic traffic demand and mobility patterns for Luxembourg, a mid-size European city. Limited by the CARLA simulation memory size, we cannot co-simulate the whole Luxembourg traffic pattern with CARLA. Therefore, a four-legged skewed intersection in the downtown area of Luxembourg is cropped out as show in Fig. 7.

In order to render the road structure in CARLA, the net file of SUMO simulation is cropped and exported to OpenDrive file. Then RoadRunner is used to perform coordinates projections and eventually fed into CARLA. Note that, there is no surrounding buildings rendered in this simulation. During the simulation, the mobility of vehicles is controlled by the SUMO simulation based on socket API and there are up to 42 vehicles rendered at the same timestamp. The model of vehicles is randomly picked but comply with the categorical definition in the SUMO simulation, such as sedan, van, bus, etc. Each vehicle is equipped with a LIDAR at 2 meter's height, which has 10Hz rotation frequency and generates around 1 million points per second. The noise along the ray-casting direction of LIDAR reading is added which sampled from a Gaussian distribution with 0.02m standard deviation. 21,902 pairs of observations are found within a 90s duration simulation with 1Hz snapshot rate. The location logging is same as the CARLA solo simulation.

C. 3D OBJECT DETECTION IMPLEMENTATION

To obtain 3D object detection results, we implemented two detectors on both datasets, respectively. For the KITTI dataset, we reproduce the 3D object detection workflow proposed in [33] to obtain the results. As there is no pre-trained 3D object detection model available for Carla synthetic data, we implemented a 2D-driven detector inspired by [33] to intimate the state of art 3D object detection performance based on ground-truth. Specifically, 2D object detection is performed on RGB images, and the detected 2D bounding box will be projected into 3D space based on depth. If a projected 2D bounding box intersects with the ground-truth bounding box, then the corresponding ground-truth bounding box will be used by adding a 3D noise vector which sampled from the uniform distribution within range [-0.2m, 0.2m]. Note that the precision and recall of such 3D object detector still depends on the 2D detection performance, where we use the pre-trained SSD-ResNet50 model provided Tensorflow Object Detection API [22].

D. BASELINE ALGORITHMS IMPLEMENTATION

As introduce in the section 2.1, we implemented 4 baseline algorithms to serve as the benchmark for point cloud performance comparison. Specifically, (1) ICP [2] is implemented based on MATLAB pcregistericp function. (2) Generalized-ICP [39] (referred as GICP in evaluation) takes advantage of the implementation of Point Cloud Library [38] (3) the deep learning feature descriptor FCGF [8] is extracted based on the pre-trained model on KITTI dataset and combines with RANSAC as a state-of-art correspondence-based baseline. (4) The closest work to ours, SSM [12] is also implemented in MATLAB. Note that, SSM [12] uses ICP to align the point clouds after its object matching.

VII. EVALUATION RESULTS

In this section, we evaluate the proposed method in terms of (i) the achievable accuracy of point cloud registration; (ii) the accuracy of object matching in the co-visible objects detection; (iii) the benefit eligibility of our method under different system settings.

A. POINT CLOUD REGISTRATION ACCURACY

Following the evaluation setup in [2], [7], [19], [25], we select accuracy, mean error and error standard deviation (STD), as the primary evaluation metrics. Specifically, the accuracy is computed based on the average distance between the ground-truth alignment and the estimated alignment. If the average distance is less than a pre-defined threshold (0.2 m in our evaluation), the estimated alignment is considered as correct aligned. The accuracy metric is actually same as the 'recall' defined in [2], [7], [19], [25]. The mean error and the STD are then computed for the correct aligned pairs.

In Table 1, we show the performance of four baseline point cloud registration algorithms, *ICP*,*GICP FCGF*, *SSM* as well as three enhanced variants by using our method denoted by *Ours + ICP*, *Ours + GICP* and *Ours + FCGF*. Across all metrics, Ours enhanced variants outperforms baseline solutions in all datasets by a significant margin. Specifically, *Ours + ICP* achieves the highest accuracy in KITTI dataset and *Ours + GICP* performs best in CARLA and SUMO dataset.

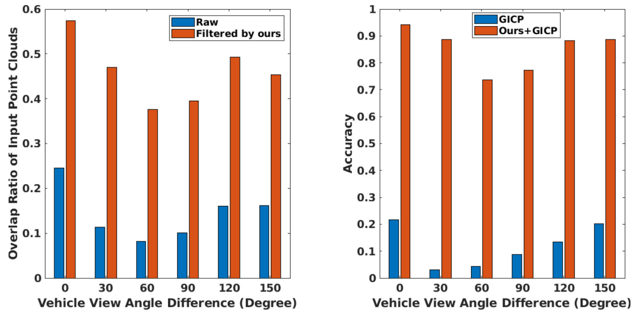
TABLE 1 Point cloud registration accuracy on KITTI, CARLA and SUMO dataset

	KITTI			CARLA			SUMO		
	Accuracy(%)	Mean(cm)	STD(cm)	Accuracy(%)	Mean(cm)	STD(cm)	Accuracy(%)	Mean(cm)	STD(cm)
ICP	8.53	10.22	9.26	5.79	10.56	10.47	15.84	11.74	11.78
GICP	14.82	9.30	9.00	22.49	11.48	11.41	19.10	11.43	11.49
FCGF	72.75	8.76	8.34	33.58	9.57	8.88	19.53	11.08	10.75
SSM	9.88	9.72	8.87	35.29	8.79	8.03	26.16	10.27	9.82
Ours+ICP	**86.68**	**7.84**	**7.20**	75.71	7.66	6.40	81.61	8.57	7.60
Ours+GICP	82.04	9.36	8.69	**92.81**	**5.92**	**4.53**	**90.55**	**6.68**	**5.44**
Ours+FCGF	84.73	9.14	8.75	65.49	10.12	9.54	67.82	10.51	10.09

Comparing to the vanilla ICP, GICP and FCGF, our method can boost the alignment accuracy to 86.68%, 92.81% and 90.55% for KITTI, CARLA and SUMO dataset respectively. Such a large gain is due to our method can extract co-visible regions between the two input point clouds and thus largely reduce the ambiguity in the point cloud registration.

Note that, the performance of method Ours+FCGF can be potentially further improved if the feature extractor of FCGF is re-trained on the extracted point clouds using our method. SSM does not work well on both datasets, as it only crops point cloud based on the object matching but not further refine co-visible regions. As KITTI dataset involves large localization errors, the object matching in SSM, which largely relies on pairwise distance between objects, fails. Our method, on the other hand, is much more robust to the large localization errors, as our method takes advantage of multiple similarity measures between observed objects and these similarity measures are resilient to the localization errors.

We also evaluate the performance of our method with respect to view-angle differences in the SUMO dataset. Since the SUMO dataset is generated based on skewed intersection and vehicles can make left and right turns from each intersection leg, the view-angle difference between any two vehicles varies from 0°to 180°. To quantify the input point clouds overlapping region over three view-angle differences, we define the overlap ratio as the number of overlapped voxels over all voxels when downsampling point clouds with 0.1m. Figure 8 shows the overlap ratio of input point clouds and the accuracy of point cloud registration for different view-angle difference, where each bar the covers the range of [x, x+30°]. Compared to the raw inputs which are the points of within all the detected object bounding boxes, the filtered point clouds using our method yield a much higher overlap ratio for different view-angle difference. The results indicate that the performance of the methods highly correlates with the overlap ratio of the input point clouds. In particular, the performance of GICP and Ours+GICP yield relative lower accuracy under smaller overlap ratio input point clouds, such as around 30°and 60°, and show relative higher accuracy for larger overlap ratio cases, like around 0°and 150°. In summary, the results show that our method can increase overlap ratio in the input point clouds and thus improves point cloud registration accuracy significantly compared with baselines.



(a) Point cloud overlap ratio across different vehicle view angle difference

(b) Point cloud registration accuracy across different vehicle view angle difference

FIGURE 8. Point cloud registration performance across different vehicle view angle difference.

Additionally, we also qualitatively compare the point cloud registration results in sample test cases across different view angle difference between ICP and Ours+ICP methods in Fig. 5. Figure 5(a) and (d) show the comparison when two vehicles are closely following each other with the same view angle. Even though such case pairs include overlapping area, there are still large portion of single-visible objects involved, which makes the ICP fails to align the point cloud correctly. Figure 5(b) and (c) demonstrate the results of ICP in 90°and 180°vehicle view angle difference respectively, where the alignment is not performed accurately due to low overlapping. However, our two phase point cloud registration method can can align the point clouds accurately as show in Fig. 5(e) and (f).

B. *OBJECT MATCHING ACCURACY*

Since the two phase design of our system takes the output of object matching to perform co-visible region refinement, we evaluate the co-visible objects detection individually in terms of precision, recall and accuracy. Note that the metrics are only calculated when our co-visible objects detection can produce a result, i.e., when there are at least three pairs of objects are kept after consistency check. Specifically, the precision and recall are defined as the number of correct co-visible matching over the number of all detected co-visible objects and the ground-truth number of co-visible objects, respectively. But the correct detection in the accuracy evaluation requires to not only match the co-visible objects, but

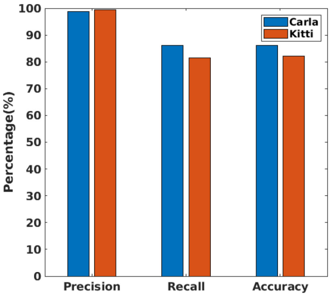


FIGURE 9. Object matching precision, recall and accuracy.

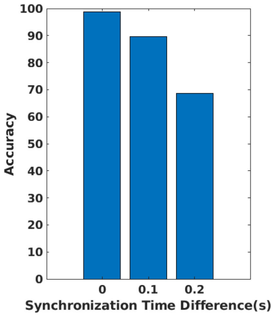


FIGURE 10. Accuracy across different synchronization time difference.

also identify the single-visible objects correctly. As shown in Fig. 9, our method can achieve 98.76% and 99.43% precision, 86.14% and 81.46% recall, 86.22% and 82.21% accuracy for the KITTI and the CARLA dataset, respectively. The high precision of our system guarantees that co-visible objects can be identified and matched accurately and thus guarantees the correctness of the input to the following co-visible region refinement and the final point cloud registration. The high recall of our system makes sure that the most co-visible objects are extracted for next phase.

C. BENEFIT ELIGIBILITY

In order to push our system to real deployment, we study the system benefit eligibility in this section, including: (1) how likely does a vehicle can find a peer to share and align point cloud based on our system (2) what is network requirement and performance trade-off for our system between different vehicle settings (3) how does the point cloud synchronization affect the system performance.

Since our system requires to discover at least three pairs of co-visible objects to fuse the point cloud, we explored how likely these cases will occur in our CARLA and SUMO intersection simulations. If a vehicle can fuse the data from at least one neighboring vehicle, it can potentially benefit from our system. In order to quantify such benefits, we define **Benefit Ratio**, which is the ratio of the number of vehicles which can gain benefit at current timestamp over the number of all vehicles. The benefit ratio is calculated for each of snapshots in our simulations. As the point cloud in CARLA dataset is generated based on four set of RGB-D cameras, we perform the experiments based on two different field of views(FoV), i.e., 360°FoV and 90°FoV by using all 4 set of cameras and only front view cameras respectively. Figure 11 shows the empirical cumulative distribution of benefit ratios. Since a larger field of view can increase the overlapping sensing area between vehicles, the benefit ratio of 360°FoV are generally higher than that of 90°for both co-visible object detection and co-visible region refinement. Even though some cases include more than 3 pairs of co-visible objects, the overlapping area of the co-visible objects are still too small to perform the co-visible region refinement. Therefore, the co-visible region refinement of both FoVs are lower than the co-visible object detection. Compared with CARLA dataset, SUMO dataset has lower benefit ratio. It is because the SUMO dataset has more diverse traffic distribution which will be harder for meet the system requirement. The variation of the benefit ratios within a setup depends on the location distribution of the objects. The benefit ratio of the co-visible region refinement with 90°FoV in CARLA dataset is greater than 0.7 for some snapshots. It is because the crossing vehicles of these snapshots are close to the center of the intersection and make themselves to be identified as co-visible objects for others. Among the cases where vehicles can perform the co-visible region refinement, we also check the number of neighboring vehicles whose information can be potentially fused. We observe that the median number of the candidate neighbours for the co-visible region refinement is 2 and 14 for the 90°FoV case and the 360°FoV case in CARLA dataset, respectively. Therefore, it is confident to believe the proposed system can benefit fair amount of vehicles when they are driving around busy intersections.

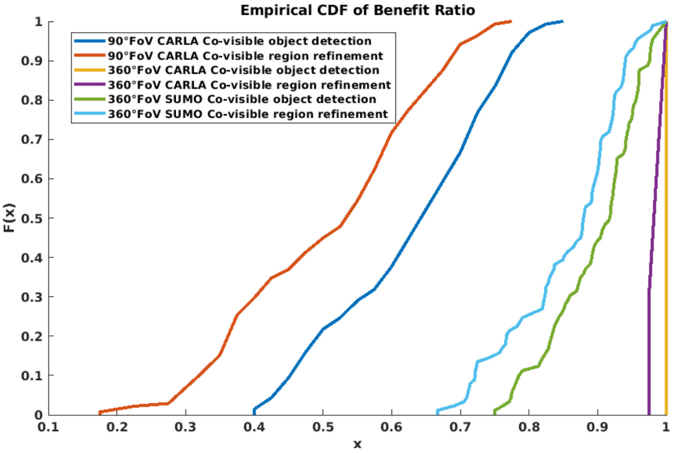


FIGURE 11. Cumulative distribution of benefit ratio.

We explore the network requirements and system performance across different field of views, as shown in Table 2. The volume of data to be shared is calculated based on the size of required information by each vehicle to perform the fusion. The accuracy in this subsection is evaluated based on

TABLE 2 Data sharing volume and system performance across different field of view

Field of View	Data sharing volume Per Vehicle		Median Point Fusion Benefit Ratio	Accuracy
	Abstract Info	Point Cloud		
90°	1.69KB	0.42MB	51.88%	85.16%
360°	4.98KB	1.53MB	99.23%	98.81%

the same definition as in section 7.1, but with a loose threshold 1m. Specifically, the volume of data required by the system to determine whether it is eligible to perform the fusion is the size of abtract information including detected object center label and point cloud visible region, and the point cloud data sharing volume is considered as the size of whole point cloud which is arguably to enable various applications. 90°FoV vehicle to vehicle information fusion requires to share 1.69KB for the abstract information and 0.42MB for the whole point cloud. 360°FoV needs more shared information, with 4.98KB and 1.53MB for the two steps respectively, but also consequently increases the benefit ratio and decreases the mean errors of point cloud registration.

Additionally, we also evaluate the system performance with different synchronization time difference. Although the point cloud can be shared with a timestamp, the synchronization between point cloud pairs may not be perfect due to hardware clock bias, sensor sample frequency, etc. Thus, the point cloud registration accuracy is evaluated according to different synchronization error by selecting input point clouds with different timestamps. As shown in Fig. 10, the point cloud registration accuracy decreases as more synchronization difference introduced. Given the relatively large synchronization error in 100ms and 200ms, the system can still align the point cloud with 89.66% and 68.55% accuracy respectively.

VIII. CONCLUSION AND DISCUSSION

In order to overcome the limitations of state-of-art point cloud registration algorithms and enable the point cloud fusion across connected vehicles, a two-phase point cloud fusion system is proposed. The system first identifies and matches co-visible objects using hyper-graph matching based on the extracted location and label information. It then estimates the co-visible region for each of them and crops out the larger overlap region. The selected co-visible area acts as an anchor point and its point cloud will be used to estimate the transformation. We evaluate the accuracy of point cloud registration and co-visible matching based on both real-world KITTI [16], [17] dataset and synthetic CARLA [11] datasets, and shows it can achieve 86.68% and 92.81% accuracy with a 0.2 m mean point-wise error threshold.

We believe that the system performance can be further improved if more objects are detected and considered during object matching. Our existing implementation focuses on detecting moving objects in the scene but static objects such as traffic lights and light poles could increase the probability of discovering 3 co-visible objects across two vehicle views and thereby improve the benefit ratio of surrounding vehicles.

REFERENCES

[1] A. Aldoma, F. Tombari, R. B. Rusu, and M. Vincze, “Our-CVFH-oriented, unique and repeatable clustered viewpoint feature histogram for object recognition and 6dof pose estimation,” in *Proc. Joint DAGM (German Assoc. Pattern Recognition) OAGM Symp.*. Springer, 2012, pp. 113–122.

[2] P. J. Besl and N. D. McKay, “Method for registration of 3D shapes,” in *Proc. Sensor Fusion IV: Control Paradigms Data Structures*, *International Society for Optics and Photonics*, 1992, vol. 1611, pp. 586–606.

[3] D. Campbell and L. Petersson, “Gogma: Globally-optimal gaussian mixture alignment,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5685–5694.

[4] A. Censi, “An ICP variant using a point-to-line metric,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2008, pp. 19–25.

[5] 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 3 D point clouds,” in *Proc. 4th ACM/IEEE Symp. Edge Comput.*, 2019, pp. 88–100.

[6] M. Cho, J. Lee, and K. M. Lee, “Reweighted random walks for graph matching,” in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2010, pp. 492–505.

[7] C. Choy, W. Dong, and V. Koltun, “Deep global registration,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2514–2523.

[8] C. Choy, J. Park, and V. Koltun, “Fully convolutional geometric features,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 8958–8966.

[9] L. Codecá, R. Frank, S. Faye, and T. Engel, “Luxembourg sumo traffic (lust) scenario: Traffic demand evaluation,” *IEEE Intell. Transp. Syst. Mag.*, vol. 9, no. 2, pp 52–63, Summer 2017.

[10] H. Deng, T. Birdal, and S. Ilic, “Ppfnet: Global context aware local features for robust 3D point matching,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 195–205.

[11] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “CARLA: An open urban driving simulator,” in *Proc. 1st Annu. Conf. Robot Learn.*, 2017, pp. 1–16.

[12] B. Douillard *et al.*, “Scan segments matching for pairwise 3D alignment,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2012, pp. 3033–3040.

[13] O. Duchenne, F. Bach, I.-S. Kweon, and J. Ponce, “A tensor-based algorithm for high-order graph matching,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2383–2395, Dec. 2011.

[14] H. Durrant-Whyte and T. Bailey, “Simultaneous localization and mapping: Part I,” *IEEE Robot. Automat. Mag.*, vol. 13, no. 2, pp. 99–110, Jun. 2006.

[15] B. Gálai, B. Nagy, and C. Benedek, “Crossmodal point cloud registration in the Hough space for mobile laser scanning data,” in *Proc. 23rd Int. Conf. Pattern Recognit.*, 2016, pp. 3374–3379.

[16] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The kitti dataset,” *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.

[17] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? The kitti vision benchmark suite,” in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2012.

[18] Z. Gojcic, C. Zhou, J. D. Wegner, L. J. Guibas, and T. Birdal, “Learning multiview 3D point cloud registration,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1759–1769.

[19] Z. Gojcic, C. Zhou, J. D. Wegner, and A. Wieser, “The perfect match: 3D point cloud matching with smoothed densities,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5545–5554.

[20] Android Location API, 2021. [Online]. Available: https://developer.android.com/reference/android/location/Location

[21] R. Guo, H. Lu, P. Gao, Z. Zhang, and H. Zhang, “Collaborative localization for occluded objects in connected vehicular platform,” in *Proc. IEEE 90th Veh. Technol. Conf.*, 2019, pp. 1–6.

[22] J. Huang *et al.*, “Speed/accuracy trade-offs for modern convolutional object detectors,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7310–7311.

[23] S.-W. Kim *et al.*, “Multivehicle cooperative driving using cooperative perception: Design and experimental validation,” *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 663–680, Apr. 2015.

[24] J. Lee, M. Cho, and K. M. Lee, "Hyper-graph matching via reweighted random walks," in *Proc. IEEE CVPR*, 2011, pp. 1633–1640.

[25] L. Li, S. Zhu, H. Fu, P. Tan, and C.-L. Tai, "End-to-end learning local multi-view descriptors for 3D point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1919–1928.

[26] P. A. Lopez *et al.*, "Microscopic traffic simulation using sumo," in *Proc. IEEE 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, 2018, pp. 2575–2582.

[27] K.-L. Low, "Linear least-squares optimization for point-to-plane ICP surface registration. Dept. Comput. Sci., Univ. of North Carolina, Chapel Hill, NC, USA, Tech. Rep. TR04-004, 2004.

[28] W. Lu, G. Wan, Y. Zhou, X. Fu, P. Yuan, and S. Song, "Deepvcp: An end-to-end deep neural network for point cloud registration," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 12–21.

[29] M. Magnusson, A. Lilienthal, and T. Duckett, "Scan registration for autonomous mining vehicles using 3D-NDT," *J. Field Robot.*, vol. 24, no. 10, pp. 803–827, 2007.

[30] J. Munkres, "Algorithms for the assignment and transportation problems," *J. Soc. Ind. Appl. Math.*, vol. 5, no. 1, pp. 32–38, 1957.

[31] B. Nagy and C. Benedek, "Real-time point cloud alignment for vehicle localization in a high resolution 3D map," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 226–239.

[32] J. Park, Q.-Y. Zhou, and V. Koltun, "Colored point cloud registration revisited," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 143–152.

[33] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3 D object detection from RGB-D data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 918–927.

[34] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 652–660.

[35] H. Qiu, F. Ahmad, F. Bai, M. Gruteser, and R. Govindan, "Avr: Augmented vehicular reality," in *Proc. 16th Annu. Int. Conf. Mobile Syst., Appl., Serv.*, 2018, pp. 81–95.

[36] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2009, pp. 3212–3217.

[37] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3D recognition and pose using the viewpoint feature histogram," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 2155–2162.

[38] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Shanghai, China, May 2011.

[39] A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," in *Proc. Robot.: Sci. Syst.*, Seattle, WA, 2009, vol. 2, p. 435.

[40] P. W. Theiler, J. D. Wegner, and K. Schindler, "Keypoint-based 4-points congruent sets-automated marker-less registration of laser scans. ISPRS journal of photogrammetry and remote sensing," vol. 96, pp 149–163, 2014.

[41] P. W. Theiler, J. D. Wegner, and K. Schindler, "Globally consistent registration of terrestrial laser scans via graph optimization," *ISPRS J. Photogrammetry Remote Sens.*, vol. 109, pp 126–138, 2015.

[42] T.-H. Wang *et al.*, "V2vnet: Vehicle-to-vehicle communication for joint perception and prediction," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2020, pp. 605–621.

[43] Y. Wang, V. Menkovski, I. -H. Ho, and M. Pechenizkiy, "Vanet meets deep learning: The effect of packet loss on the object detection performance," in *Proc. IEEE 89th Veh. Technol. Conf. (VTC2019-Spring)*, 2019, pp. 1–5.

[44] Y. Wang and J. M. Solomon, "Deep closest point: Learning representations for point cloud registration," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 3523–3532.

[45] B. Wu, A. Wan, X. Yue, and K. Keutzer, "Squeezeseg: Convolutional neural nets with recurrent CRF for real-time road-object segmentation from 3 D LiDAR point cloud," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, 2018, pp. 1887–1893.

[46] H. Yang, J. Shi, and L. Carlone, "Teaser: Fast and certifiable point cloud registration," *IEEE Trans. Robot.*, vol. 37, no. 2, pp. 314–333, 2020.

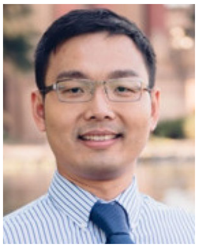
[47] J. Yang, H. Li, D. Campbell, and Y. Jia, "Go-ICP: A globally optimal solution to 3 D ICP point-set registration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 11, pp. 2241–2254, Nov. 2016.

[48] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, "3dmatch: Learning local geometric descriptors from RGB-D reconstructions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1802–1811.

[49] Q.-Y. Zhou, J. Park, and V. Koltun, "Fast global registration," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2016, pp. 766–782.

HONGYU LI received the B.S. degree in computer science from Harbin Engineering University, Harbin, China, in 2011 and the M.S. degree in computer science from the University of Chinese Academy of Sciences, Bejing, China, in 2014. He is currently working toward the Ph.D. degree with the Department of Computer Science, Rutgers University, New Brunswick, NJ, USA. He conducts research in Wireless Information Network Laboratory (WINLAB) and his current research interests is mobile computing with the special focus on driver behavior and vehicle dynamics sensing, cooperated sensing between vehicles.

HANSI LIU (Student Member, IEEE) received the M.S. degree in 2018 from Rutgers University, New Brunswick, NJ, USA, where he is currently working toward the Ph.D. degree with Wireless Information Network Laboratory (WINLAB), Department of Electrical and Computer Engineering. His research interests include mobile computing and computer vision, with a special focus on collaborative perception, multi-modal sensing and localization systems. He is currently exploring cross-modal sensing approaches to improve sensing range, efficiency, tracking and localization for collaborative perception systems via computer vision and wireless communication. Previously, he worked on topics of image retrieval and latent space representation.



HONGSHENG LU received the bachelor's and master's degrees in electrical engineering from Beihang University, Beijing, China, in 2006 and 2009, respectively, and the Ph.D. degree in computer science and engineering from the University of Notre Dame, Notre Dame, IN, USA, in 2015. He is currently a Principal Researcher with Toyota Motor North America Research and Development - InfoTech Laboratories. His research interests include connected and automated vehicle technology, building solutions to enable cooperative perception, and V2X-assisted sensor fusion. He was recognized for his contribution to DSRC congestion control and vehicle-to pedestrian communications. He represents Toyota with Standard Development Organizations and Industry Groups including ETSI and C2C-CC, and is the Vice-Chair of the SAE V2X Core Technical Committee. He is an invited Reviewer to several IEEE journals and many international ITS-related conferences.

BIN CHENG received the Doctoral degree in electrical and computer engineering from Rutgers University, New Brunswick, NJ, USA, in 2019. After that, he joined Toyota InfoTech Laboratories, USA, as a Researcher. His primary research interests include channel congestion control and modeling for vehicular networks, collaborative perception for self-driving and connected vehicles. He was the Technical Program Committee Member for multiple conferences and workshops, including IEEE VTC (Recent Results Track) 2018, 2019, 2022, IEEE ICUWB 2015, and a Reviewer for top-tier conferences and journals, including IEEE TITS, IEEE TMC, IEEE/ACM ToN, ACM TOSN, IEEE INFOCOM, IEEE WCNC, IEEE VTC, IEEE IV. Bin was the recipient of the Best Paper Award of IEEE Vehicular Networking Conference, 2019.

MARCO GRUTESER received the M.S. and Ph.D. degrees from the University of Colorado, Boulder, CO, USA, in 2000 and 2004, respectively. He held research and visiting positions with the IBM T. J. Watson Research Center and Carnegie Mellon University, Pittsburgh, PA, USA. He is currently a Professor of electrical and computer engineering and also computer science (by courtesy) with Wireless Information Network Laboratory (WINLAB), Rutgers University, New Brunswick, NJ, USA . He research interest include mobile computing, is a pioneer in the area of location privacy and recognized for his work on connected vehicles. He was the Program Co-Chair or Vice-Chair for conferences such as ACM MobiSys, ACM WiSec, IEEE VNC and IEEE Percom. He has delivered nine conference and workshop keynotes, was Panel Moderator at ACM MobiCom, and as a Panelist at ACM MobiSys, IEEE Infocom, and IEEE ICC. He was elected Treasurer and Member of the executive committee of ACM SIGMOBILE. He was the recipient of the NSF CAREER Award, a Rutgers Board of Trustees Research Fellowship for Scholarly Excellence, a Rutgers Outstanding Engineering Faculty Award, and also Best Paper Awards at ACM MobiCom 2012, ACM MobiCom 2011 and ACM MobiSys 2010. His work has been regularly featured in the media, including NPR, the New York Times, Fox News TV, and CNN TV. He is an ACM Distinguished Scientist.

TAKAYUKI SHIMIZU received the B.E., M.E., and Ph.D. degrees from Doshisha University, Kyoto, Japan, in 2007, 2009, and 2012, respectively, where he studied physical-layer security exploiting multipath fading randomness in wireless communications. From 2009 to 2010, he was a Visiting Researcher with Stanford University, CA, USA. From 2012 to 2019, he was with TOYOTA InfoTechnology Center, USA, Inc. He is currently a Principal Researcher with Toyota Motor North America, Inc., Research and Development InfoTech Laboratories where he works on the research and standardization of wireless vehicular communications. His research interests include millimeter-wave vehicular communications, vehicular communications for cooperative automated driving, and LTE/5G for vehicular applications. He is a 3GPP delegate for V2X standardization and a Member of several SAE Technical Committees for V2X. He was a Workshop Co-Chair for the 2018 Fall IEEE Vehicular Technology Conference. He is a Member of the IEICE. He was the recipient of the 2010 TELECOM System Technology Award for Student from the Telecommunications Advancement Foundation and the 2020 IEEE Vehicular Networking Conference Best Paper Award.