2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

Argus: Multi-view LiDAR Point Cloud Fusion for Enhancing Vehicle Detection in Auto Driving

Yifei Tian[1,*], Hongwei Huang[1], Xiangyu Li[1]

[1] Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu, China
tyf@njupt.edu.cn

Abstract. The environmental perception of unmanned ground vehicles (UGVs) directly impacts decisions like path planning and obstacle avoidance, with vehicle detection being critical for autonomous driving. LiDAR provides high-precision point clouds but suffers from sparse density and self-occlusion, often resulting in incomplete vehicle point clouds that hinder detection performance. To address this, we propose ***Argus***, a multiview registration and completion model that fuses multi-frame point clouds of surrounding vehicles. Argus achieves multi-view fusion through a self-attention-based cumulative registration module and a coarse-to-fine residual completion module, refining vehicle point clouds using grid residual layers and a multilayer perceptron. Compared to single-view point clouds, Argus produces denser and more complete vehicle shapes, serving as an independent plug-in to enhance detection methods. Experiments on the KITTI dataset show that Argus improves downstream vehicle detection performance.

Keywords: LiDAR Point Cloud Fusion, Multiple Accumulating Registration Strategy, Coarse to Fine Complete, Vehicle Detection.

1 Introduction

Accurate visual perception is essential for unmanned ground vehicles (UGVs) to interact effectively with their environment. UGVs rely on visual sensors like cameras and LiDAR to gather crucial data for scene understanding and subsequent driving decisions. LiDAR, widely adopted for its long sensing range and precise distance detection [1], often produces sparse and incomplete point clouds due to inherent limitations such as self-occlusion. This compromises UGVs' capabilities in scene analysis, vehicle detection, and downstream tasks. Since vehicle detection is critical for tasks like path planning and obstacle avoidance [2], improving the completeness of vehicle point clouds has become a key focus of research.

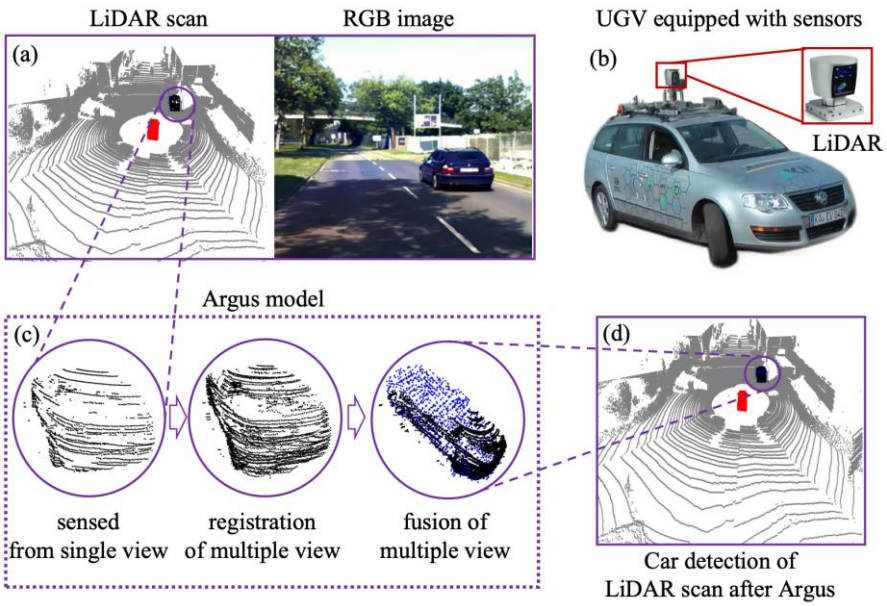

Fig. 1. Overview of the proposed framework. (a) The LiDAR point cloud and RGB image captured by UGV in the KITTI dataset. (b) Data collection platform of KITTI dataset. (c) The overview of the Argus with the input of single view point clouds, and the output as fusion results of multiple view registration and completion. (d) Vehicle detection on fusion results of multiview point clouds by the proposed model.

Point clouds captured by unmanned ground vehicles (UGVs) come from a series of moving perspectives [3], making the completion of moving vehicles a more significant challenge compared to static objects [4]. To improve completeness, researchers commonly use ICP and its variants to align and fuse multi-frame LiDAR point clouds from sequential views [5]. Most existing point cloud completion models focus on single-view data, overlooking the need for multiview completion across diverse perspectives. While some models explore correlations between multiview point clouds, most are designed for single-frame data and rarely address continuous frames or transformations between consecutive moving views [6][7]. As a result, multiview point cloud fusion from consecutive frames captured during autonomous driving remains an important yet underexplored challenge [8].

Deep learning models for point cloud completion can be broadly categorized into three main types: point convolution-based models, multi-view completion-based models, and generative adversarial network (GAN)-based models [9], such as PCN [10], the 3D capsule model [11], 3DGAN [12], and PFNet [13]. Existing training datasets for these models, such as ModelNet [14], ShapeNet [15], and MVP [16], primarily consist of synthetic data from irregular views, making them unsuitable for multiview completion tasks in consecutive moving perspectives. Meanwhile, multiview completion networks perform well on synthetic dense point cloud datasets [17], they struggle to address the challenges of consecutive moving views in UGVs and often show limited

performance on real-world data. This is because point clouds captured by real sensors are typically much sparser than synthetic data, and inherent challenges such as sparse density and self-occlusion [4] continue to hinder object completion performance.

Overall, this issue is further complicated by self-occlusion [18], incomplete surfaces[19], and the lack of ground truth data during training [18], leading to the widespread use of synthetic datasets for training completion models [20]. In response to these challenges, this paper proposes Argus, a fusion model for multiview point clouds captured from moving perspectives. The Argus model consists of two key modules: a multi-view cumulative registration module and a coarse-to-fine completion module. The registration module uses an accumulation strategy to compute rigid transformations between frames, while the completion module fills in missing parts of detected vehicle point clouds, producing dense and complete shapes. To improve vehicle detection in real-world autonomous driving scenarios, the Argus model enhances the quality of point clouds through these modules. A new training dataset, tailored for multiview point cloud fusion tasks, is also introduced, and the model is validated using the KITTI dataset [21]. By integrating Argus with existing vehicle detection methods, vehicle point cloud quality is significantly improved, leading to better detection performance. The main contributions of this paper are summarized as follows:

1. This paper proposes a two-stage fusion model as Argus to achieve the registration and completion of multi-frame point clouds, aimed at enhancing vehicle detection performance during UGV movement.
2. This paper designs a multi-view cumulative registration module to perform rigid registration of local multiple frames through a cumulative registration strategy.
3. The coarse-to-fine residual completion module employed in this paper facilitates the completion of registration results, thereby improving vehicle detection performance on the KITTI dataset.

2 Problem Modeling

2.1 Multi-view splitting

We model the fusion task of vehicle point clouds captured by continuous moving views from UGVs as the registration and completion stages. Due to the lack of the complete point clouds (ground truth) captured from real scenes, we simulate vehicle point cloud sequences in Euclidean space as $X=\{x_i, ..., x_j, ..., x_k\}$, $j \in [i, k]$ from multiple moving virtual views P. The sequence x_j is generated as a part from the original point cloud $\mathcal{X} \in \mathrm{R}^{\mathrm{N}\times 3}$. The virtual viewpoint $p_j \in R^3$ is a coordinate of the sensor center moving along a predetermined trajectory. From each view-point p_j, original points $\mathcal{X}$ divided into seen point $x_j \in R^{n_j\times 3}$ and unseen point $y_j \in R^{m_j\times 3}$ as $\mathcal{X} = \{x_j \cup y_j\}, N = n_j + m_j$ caused by self-occlusion. Inspired by previous work [18], we define the generation of point clouds x_j and virtual view-point p_j through an injective mapping $o_j(\cdot)$, where $o_j: \mathcal{X} \to x_j$ map N full points to n_j occluding points.

The mapping o_j is mainly divided into three steps: first, transform point clouds $\mathcal{X}$ from 3D world reference into the reference frame of view-point p_j; second, compute the seen or unseen points by viewpoint $p_j$according to self-occlusion; third, inverse transform point clouds x_j into world reference. When computing the seen or unseen parts in the second step, we simulate the generation method of LiDAR point clouds perceived from the real world. When a laser beam is emitted from the laser transmitter, reflection occurs when the laser beam encounters an opaque object surface. The receiver calculates the distance based on the reflected time to generate LiDAR point clouds. Thus, we consider viewpoints as the virtual laser emission center and simulate the self-occlusion to generate vehicle point clouds (visible x_j). For partial surfaces that cannot be sensed by the laser due to the self-occlusion problem, we treat them as invisible points $y_j$and remove them from original point clouds $\mathcal{X}$. A sample for fusion model training is defined as input $X = \{x_j | j = i, \dots, k\}$ and target ground truth$\mathcal{X}$.

2.2 Problem definition

To improve the quality of incomplete point clouds sensed from multiple consecutive frames, this paper designs the fusion target into two modules as registration and completion. The first module is to register $k - i$ views of incomplete point clouds x_j into the reference x_k based on their transformation T_{jk}. Then, using the union operation $\cup$ to concatenate transformed point clouds and adopting function δ to merge and refine point clouds as Z. The Θ is defined as the parameters of neural networks in the complete module. Finally, predict complete results $\widehat{\mathcal{X}}$ based on incomplete point clouds Z by using the compete module Θ.

$$\widehat{\mathcal{X}} \equiv \Theta(Z) = \Theta\left(\delta\left(\{T_{ik}x_i \cup \dots T_{jk}x_j \dots \cup x_k\}\right)\right) \tag{1}$$

The homogeneous transformation $T_{jk} \in R^{4\times4}$ between point clouds x_j and x_k, where the T_{jk} contains rotation matrix $R_{jk} \in R^{3\times3}$ and translation vector $t_{jk} \in R^3$. The training target of our model is defined as Eq. (2), including registration result T, registration module θ, and complete module Θ for fusion point clouds into $\hat{X}$. The framework's target is minimizing the loss J among predicted fusion point clouds $\widehat{\mathcal{X}}$ and the ground truth $\mathcal{X}$.

$$T, \theta, \Theta = \arg \quad min_{T_{jk} \in \mathcal{T}, j \in [i,k]} J\left(\widehat{\mathcal{X}}, \mathcal{X}\right) \tag{2}$$

3 Framework design

The framework of our proposed Argus network is designed as Fig. 2, which contains two modules multi-view cumulative registration and coarse-to-fine completion. The input of the Argus is occlusion sequence as $X = \{x_i, \dots x_j, \dots x_k\}$ as $X \in R^{(k-i)\times n_j \times 3}$. The predicted outputs of the registration module are the $k - i$ transformation matrices T_{jk} and the merged point clouds Z. The predicted output and ground truth of the complete module is $\widehat{\mathcal{X}}$ and $\mathcal{X}$.

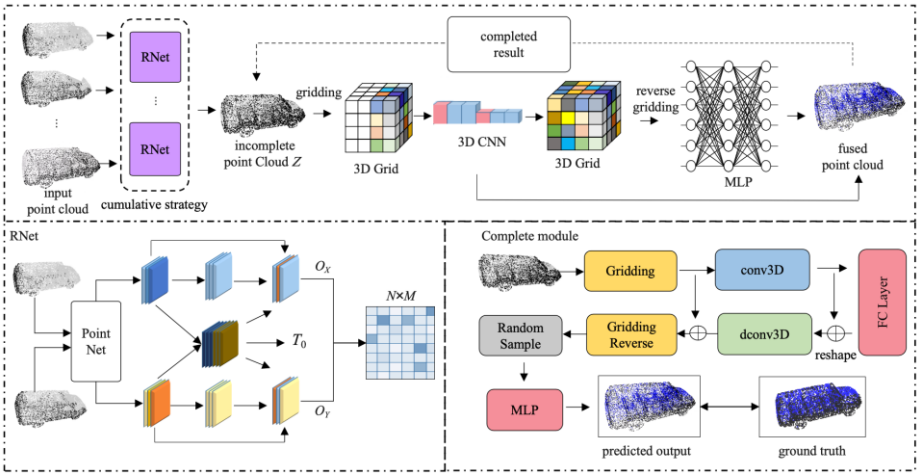

Fig. 2. Framework of the Argus for multiple frames of point cloud registration and completion.

3.1 Registration Module

Feature extraction. To achieve the registration among adjacent frames x_j to x_k, the 6-DoF rigid transformation matrix T_{jk} is computed using registration module [22]. First, we adopt the weight-shared PointNet as a feature encoder to extract the global features of each view of vehicle point clouds. After fusing the global features of point clouds x_j and x_k, we use an MLP to compute the initial transformation T_{jk}^0. Through using the T_{jk}^0, we transform the vehicle point clouds $x_j$into $x_j(T_{jk}^0)$. Next, we use self- and cross-attention in RNet to extract pointwise features f_j and f_k of $x_j(T_{jk}^0)$ and x_k. Using the above extracted features$f_j$and f_k of two frames j and k, we compute the similarity matrix S_{jk} as $S_{jk} = f_j \cdot f_k^T$. To filter the corresponding point pairs with low similarity, we adopt the relaxation constraints of the similarity matrix as $\sum_{l=1}^{n_j} c_{lm} \leq 1$ and $\sum_{m=1}^{n_k} c_{lm} \leq 1$, where the $c_{lm} \in [0,1]$ [22]. The top-$\overline{n_k}$ correspondences are obtained as $C_{jk} = \{(C_{j,k}^j, C_{k,j}^k)\}$ from the similarity matrix S_{jk}. According to the correspondences C_{jk}, RNet employ a weighted SVD algorithm to directly compute the transformation between frames at once, where not use the random sample consensus (RANSAC) method to avoid time-consuming multiple iterations. The transformation matrix T_{jk} between the frames j and k are computed.

Cumulative registration. After computing the transformation matrix T_{jk} between the frames j and k, the point clouds x_j and x_k are registered into the same reference through using RNet. Due to the vehicle point clouds are obtained by successive views in autonomous driving, the transformation matrix T_{jk} is defined as the $k-j+1$ transformation matrix from $T_{j(j+1)}$ to $T_{(k-1)k}$ as shown in Eq.3. Besides, the $T_{j(j+2)}$ also could be defined as the product of transformation $T_{j(j+1)}$ and $T_{(j+1)(j+2)}$. According to

the above principle, the registration of point cloud X_j enables be transformed into frame k through correction by modifying the T_{jk}. The transformed k-j frames of vehicle point clouds are defined as $Z \in R^{(k-j)\times N_{jk}\times 3}$, where$N_{jk} = n_j + \cdots + n_k$. As the output of the registration module, the Z is the input of the next completion module.

$$T_{\mathrm{jk}} = T_{\mathrm{j(j+1)}}T_{\mathrm{(j+1)(j+2)}}\ldots T_{\mathrm{(k-2)(k-1)}}T_{\mathrm{(k-1)k}} = T_{\mathrm{j(j+1)}}T_{\mathrm{(j+1)(j+2)}}\ldots T_{\mathrm{(k-2)k}} = T_{\mathrm{j(j+1)}}T_{\mathrm{(j+1)k}} \quad (3)$$

3.2 Complete Module

Coarse completion. After merging multiple frames of point clouds Z, the point number N_{jk} is a relatively large number so we first adopt a uniform down-sampling method to remove redundant points inspired by [2]. Next, the unstructured point clouds are converted into regular 3D grids $U = \{u_a | a \in \mathbb{R}\hat{}3\}$ in the coarse stage. In order to make 3DCNN gridding layers differentiable on vertex u_a, we defined a neighboring area $\mathcal{N}(u_a) = \{u_q | q \in \mathbb{R}\hat{}3\}$ of grid u_a. The weight w_a of grid $u_a$is initialized as the mean of its neighborhood grid weight $w(u_a, z)$ by using the neighboring points z allocated in the neighboring grid u_q as shown in Eq. 4, where the function ψ is the interpolation function, the $|\mathcal{N}(u_a)|$ is the number of neighboring points in $\mathcal{N}(u_a)$.

$$w_a = \sum_{z\in\mathcal{N}(u_a)} \frac{w(u_a,z)}{|\mathcal{N}(u_a)|} = \sum_{z\in\mathcal{N}(u_a)} \frac{\psi(z)}{|\mathcal{N}(u_a)|} \quad (4)$$

After calculating the weight w, the 3DCNN layers in the coarse complete stage utilize a skip connection to bridge 4 groups of 3D convolution layers and 4 groups of 3D deconvolution layers with 2 groups of fully connected layers. The complete shapes are converted from grids into coarse point clouds $Z^c$by using an inverse gridding step as Eq.5, where the point z^c is defined as the weighted combination of weight w_q and neighboring vertex u_q.

$$z^c = \sum_{u_q\in\mathcal{N}(u_a)} w_q\, u_q / \sum_{u_q\in\mathcal{N}(u_a)} w_q \quad (5)$$

Fine completion. After obtaining the complete results of coarse point clouds Z^c from the coarse stage, all features in 3D grids are aggregated as f^c. Using the grid features f^c, the 4-layer MLPs are designed to complete point clouds with more details, which means the MLPs tend to fit the residual offsets of coordinates between coarse complete point clouds and the ground truth. After obtaining the predicted registration matrix T and fine completion point clouds $\widehat{\mathcal{X}}$, the weight and parameters of the proposed Argus model are optimized by back forward the loss between that of predicted and ground truth.

3.3 Loss Function

We define the loss function of the proposed Argus model into four terms, including registration loss of multiview point clouds L_{reg}, merge loss of point distance among

multiview point cloud L_{pd}, the complete loss of generated grid in the coarse stage L_{grid}, and the complete loss of generate point clouds in the fine stage L_{CD}. The registration loss L_{reg} is defined as the summation of multiple views' transformation differences between predicted and ground truth of rotation matrix R_{jk}, $\widehat{R_{jk}}$ and translate vector t_{jk}, $\widehat{t_{jk}}$ as shown in Eq. 6.

$$L_{reg} = \sum_{j=i}^{k-1} |\widehat{R_{jk}^T} R_{jk} - I_3|_F^2 + |\widehat{t_{jk}} - t_{jk}|_2^2 \tag{6}$$

The merge loss L_{pd} is computed by the summation of the difference of point distance among k frames in Eq. 7.

$$L_{pd} = \sum_{j=i}^{k-1} \quad |\left(\widehat{R_{jk}} X_j + \widehat{t_{jk}}\right) - \left(R_{jk} X_j + t_{jk}\right)|_2^2 \tag{7}$$

The grid loss L_{grid} is defined as the grid difference in the coarse completion stage in Eq. 8, where the U and $\widehat{U}$ are predicted grid and ground truth.

$$L_{grid} = \frac{1}{\mathrm{D}} \sum ||U - \widehat{U}|| \tag{8}$$

The L_{CD} term utilizes the Chamfer distance to estimate the fine complete loss between predicted and ground truth point clouds $\mathcal{X}$ and $\widehat{\mathcal{X}}$ in Eq. (9).

$$L_{CD}(\mathcal{X}, \widehat{\mathcal{X}}) = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \min_{\hat{x} \in \widehat{\mathcal{X}}} ||x - \hat{x}||^2 + \frac{1}{|\widehat{\mathcal{X}}|} \sum_{\hat{x} \in \widehat{\mathcal{X}}} \min_{x \in \mathcal{X}} ||x - \hat{x}||^2 \tag{9}$$

4 Experiments

4.1 Datasets

Training stage. The proposed Argus model is trained using point cloud samples generated from synthetic vehicle objects in the ShapeNet dataset [14][23], where vehicle objects, stored as CAD models, consist of dense points and triangular surfaces. To simulate the characteristics of LiDAR point clouds, including sparse density and self-occlusion, sparse and partially visible point clouds are extracted from virtual observation points, as detailed in Fig. 3. We select car objects (CAD model) from the ShapeNet dataset to generate the multiview point cloud for the Argus training, where point clouds are simulated from continuous frames with different views (e.g. view 1 and view 2). The seen (black) or unseen (grey) parts of point clouds sensed from view 1 or 2 are also different, which is similar to the vehicle perceived by LiDAR sensors during driving. Besides, we downsample the point cloud from CAD models to make their density as close as possible to that collected from LiDAR sensors. Due to the training target of Argus being to restore the vehicle shape from partial point clouds, the vehicle point clouds from CAD models do own the complete shape that satisfies our requirement. Each vehicle object is observed from six different viewpoints, resulting in six groups of point clouds per object as training samples. The transformation matrices between these six groups are provided as the ground truth for the registration module, while the

complete point cloud of the vehicle object, including unseen regions, serves as the ground truth for the overall framework.

Testing stage. This study evaluates the fusion performance of vehicle point clouds using object detection data from the KITTI dataset [21]. Since KITTI lacks ground truth labels specifically for vehicle registration and completion, the performance of vehicle detection is used to quantitatively analyze the fusion results of multiview point clouds. The vehicle detection task in the KITTI dataset includes 7,481 frames in the training set and 7,518 frames in the testing set. However, as the testing set lacks explicit ground truth, which is only accessible via server submission for vehicle detection tasks, we follow the approach in [24] that uses 2,240 frames for evaluating the performance of vehicle detection, based on which we build the testing dataset of vehicle fusion task with 2,401 vehicle objects in Section 4.2 to 4.4. To enhance the diversity of the testing samples, frames from different sequences are selected wherever possible. Additionally, the first 50 frames contain 31 vehicle objects to verify the effectiveness of individual components in the Ablation study in Section 4.5.

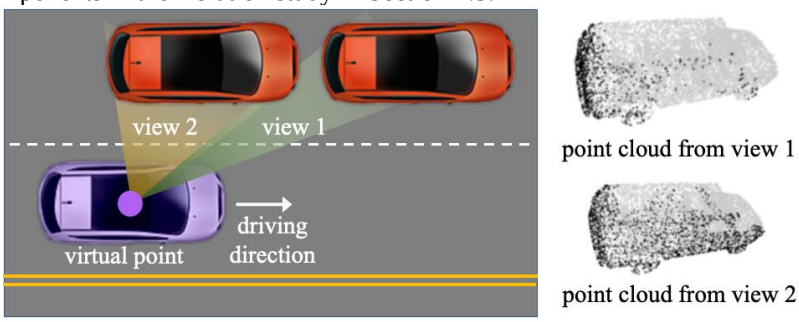

Fig. 3. Simulation of point clouds from multiview are generated based on vehicle objects from the ShapeNet dataset (CAD model). Black points mean the seen part, and the gray points mean the unseen parts.

4.2 Compare algorithm

To evaluate the fusion performance of our proposed Argus model, we compare its results with several popular point cloud completion algorithms, including 3DCapsule [11], PCN [10], PFNet [13], GRNet [2], and PoinTr [25]. The comparison results are presented in Table 1, where we use three indicators—minimal matching distance (MMD), fidelity distance (FD), and consistency—to assess the completion performance. MMD is computed as the Chamfer Distance (CD) between the predicted output point clouds and the most similar vehicle point clouds in ShapeNet. FD is defined as the average distance between each point in the input and its nearest neighbor in the predicted output. Consistency measures the average CD between the predicted outputs of point clouds for the same vehicle instance across multiple views. As shown in Table 1, Argus achieves lower values for MMD, FD, and consistency compared to other models, indicating superior performance. In particular, the significantly improved MMD and consistency metrics highlight the effectiveness of Argus. A lower MMD indicates

that the completed shapes are more accurate and closely resemble general vehicle instances, while better consistency reflects the robustness and uniformity of Argus's completion results across multiple views.

Table 1. Comparison results of our proposed Argus and other popular complete algorithms.

Methods	MMD (10^3)	FD (10^3)	Consistency (10^{-3})
3DCapsule	2.962	3.508	1.951
PCN	1.366	2.235	1.557
PFNet	1.016	1.137	0.792
GRNet	0.568	0.836	0.313
PointTR	0.526	0.000	0.683
Ours(Argus)	**0.506**	0.802	**0.280**

4.3 Detection results

To quantitatively evaluate the vehicle detection performance in driving scenes, we analyze the experimental results on the KITTI dataset using three indicators: mean inter over union (mIoU), mean average orientation similarity (mAOS), and mean average precision (mAP) at two IoU thresholds, 0.5 and 0.75. The mIoU is calculated as the IoU of bounding boxes in Euclidean space between predicted outputs and ground truth. A higher mIoU indicates a larger overlap region, reflecting better vehicle detection performance. The mAOS measures the similarity between the predicted vehicle orientation and the ground truth, with higher values indicating more accurate orientation predictions. The mAP at an IoU threshold of 0.5 is the mean value of the average precision, computed for detection samples with an IoU greater than 0.5. These metrics comprehensively evaluate the detection accuracy and orientation precision of the proposed approach, providing a robust assessment of its performance in real-world driving scenes.

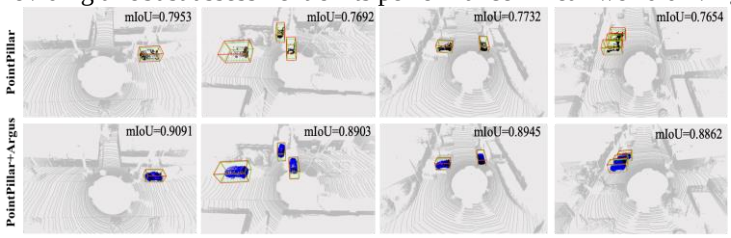

Fig. 4. Comparison experimental results of vehicle detection visualization by PointPillar and combination of PointPillar and Argus models on KITTI dataset. The bounding box with red and green are the detection of predicted and ground truth, respectively. The mIoU is computed as the average value of the predicted vehicle objects in the current single frame; for instance, the mIoU is computed as the average of the three detected vehicles in the second column.

Using the popular vehicle detection method PointPillar [26], we compare the detection performance of PointPillar alone and the combination of PointPillar and Argus

(PointPillar+Argus), as shown in Table 2. The results indicate significant improvements in detection performance when incorporating the Argus model. Notably, the mIoU improves substantially from 0.7722 to 0.8917, attributed to Argus enhancing the completeness of vehicle point clouds, which benefits the prediction of 3D bounding boxes in detection tasks. Additionally, the mAP values at both IoU thresholds (0.5 and 0.75) show approximately a 10% improvement with Argus, further confirming its positive impact on vehicle detection. Moreover, Fig. 4 provides a visual comparison of vehicle detection results between PointPillar alone and PointPillar combined with Argus. The visualization clearly demonstrates that detection performance is significantly enhanced when the Argus model is integrated. These findings validate the effectiveness of Argus in supporting and improving vehicle detection tasks.

Table 2. Comparison results of our proposed Argus and other popular complete algorithms.

Methods	PointPillar	PointPillar+Argus
mIoU	0.7722	0.8917
mAOS	0.8876	0.9141
mAP(IoU threshold=0.5)	0.7416	0.8534
mAP(IoU threshold=0.75)	0.6687	0.7638

4.4 Multiview analysis

We evaluate the optimal number of views for the cumulative registration module in Argus, as illustrated in Fig. 5. The quantitative results in Fig. 5 (left) indicate that fusing point clouds from three views achieves superior vehicle detection performance compared to using a single view, two views, or more than three views, regardless of whether the registration is performed using ICP or Argus. This improvement occurs because point clouds fused from fewer than three views lack sufficient information for effective vehicle detection, while merging more than three views introduces excessive redundant points and requires estimating a larger number of transformation matrices, which increases the likelihood of registration errors. Fig. 5 (right) presents visualization results for different view numbers obtained from the cumulative registration module. It is evident that point clouds fused from three views are significantly denser and more complete. The next subsection further compares the detection performance achieved by the cumulative registration module and ICP.

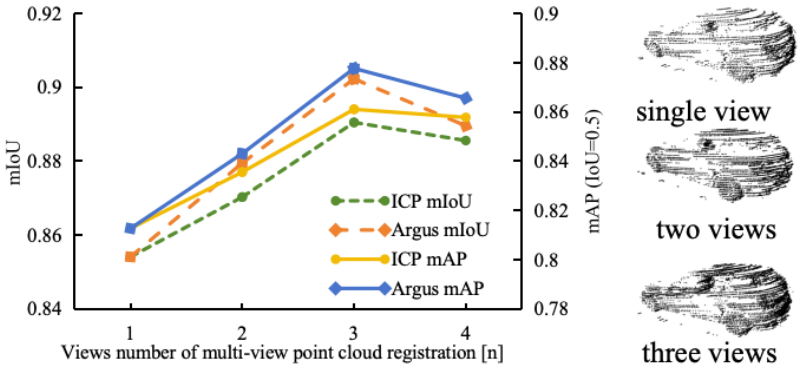

Fig. 5. Registration results of different view numbers based on baseline ICP and our proposed Argus model. our proposed multi-view cumulative registration module from multi-view vehicle point clouds.

4.5 Ablation study

To verify the effectiveness of each module of Argus, we design an ablation study to estimate the performance of vehicle detection in Table 3. The experimental results are computed on the first 50 consecutive frames in the validation set of the KITTI datasets. We estimate the modules' effectiveness of Argus, including the single view (SV), registration module set as ICP method as vanilla baseline (ICP), our proposed multiview cumulative registration module (MVR), and coarse-to-fine completion module (C2FC). Besides, the effectiveness is evaluated by the mIoU and mAP with the IoU threshold equal to 0.5. It is obvious that using both the designed MVR and C2FC are the most efficient modules for the detection performance. Especially after adding the C2FC modules, both mIoU and mAP are significantly increased around 0.05 and 0.07 respectively.

Table 3. Ablation study of our proposed Argus method.

No.	SV	ICP	MVR	C2FC	mIoU	mAP
1	√				0.7706	0.7499
2		√			0.7956	0.7546
3			√		0.8003	0.7588
4	√			√	0.8541	0.8127
5		√		√	0.8905	0.8611
6			√	√	0.9023	0.8777

4.6 Missing detection case

Missed detection cases sometimes occur in the results produced by PointPillar. Argus has the potential to mitigate these issues as illustrated in Fig. 6. By fusing vehicle point

clouds from previous frames, Argus focuses on regions near the vehicle detection to increase the likelihood of successfully detecting the vehicle.

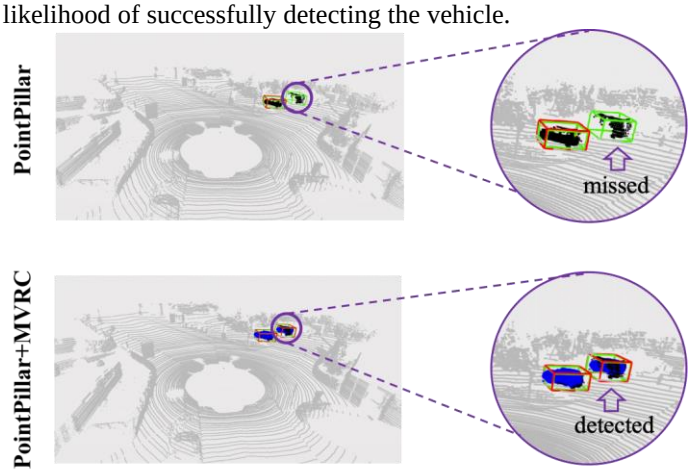

Fig. 6. Missing Detection Case. The bounding box with red and green are the detection of predicted and ground truth, respectively.

5 Conclusion

This paper proposes Argus to fusion point clouds from multiple views, enhancing vehicle detection efficiency during autonomous driving. The training dataset is extracted from vehicle objects in the ShapeNet dataset, based on which sparse point clouds are simulated in the virtual perspectives to generate training samples. By leveraging the proposed multi-view cumulative registration and coarse-to-fine completion modules, vehicle detection performance on the KITTI dataset is significantly improved. Additionally, this paper compares the proposed approach with current popular completion models, which verify superior completion results on front, side, and rear perspectives. Furthermore, the proposed multi-view cumulative registration module shows better comparison results than the classical alignment baseline ICP algorithm under our proposed multi-view cumulative strategy. Finally, this method enables improving the situation of the missed detection problem by fusing multi-view point clouds from multiple perspectives. In the future, we plan to extend the proposed model to additional applications, such as environmental perception for mobile robots to enhance their performance.

Acknowledgments. This work was supported in part by the National Nature Science Foundation of China (No. 62402236), in part by the Natural Science Research Start-up Foundation of Recruiting Talents of Nanjing University of Posts and Telecommunications under Grant (Nos. NY222065 and NY222102).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Fei B, Yang W, Chen W M, et al. Comprehensive review of deep learning-based 3d point cloud completion processing and analysis[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(12): 22862-22883.
2. Xie H, Yao H, Zhou S, et al. Grnet: Gridding residual network for dense point cloud completion[C]//European conference on computer vision. Cham: Springer International Publishing, 2020: 365-381.
3. Liu M, Sheng L, Yang S, et al. Morphing and sampling network for dense point cloud completion[C]//Proceedings of the AAAI conference on artificial intelligence. 2020, 34(07): 11596-11603.
4. Gu J, Ma W C, Manivasagam S, et al. Weakly-supervised 3d shape completion in the wild[C]//Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16. Springer International Publishing, 2020: 283-299.
5. Koide K, Yokozuka M, Oishi S, et al. Voxelized GICP for fast and accurate 3D point cloud registration[C]//2021 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2021: 11054-11059.
6. Pan L, Chen X, Cai Z, et al. Variational relational point completion network[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021: 8524-8533.
7. Wen X, Xiang P, Han Z, et al. Pmp-net: Point cloud completion by learning multi-step point moving paths[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021: 7443-7452.
8. Khurana T, Hu P, Held D, et al. Point cloud forecasting as a proxy for 4d occupancy forecasting[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023: 1116-1124.
9. Guo Z, Zhang R, Qiu L, et al. Joint-mae: 2d-3d joint masked autoencoders for 3d point cloud pre-training[J]. arXiv preprint arXiv:2302.14007, 2023.
10. Yuan W, Khot T, Held D, et al. Pcn: Point completion network[C]//2018 international conference on 3D vision (3DV). IEEE, 2018: 728-737.
11. Zhao Y, Birdal T, Deng H, et al. 3D point capsule networks[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 1009-1018.
12. Wu J, Zhang C, Xue T, et al. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling[J]. Advances in neural information processing systems, 2016, 29.
13. Huang Z, Yu Y, Xu J, et al. Pf-net: Point fractal network for 3d point cloud completion[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 7662-7670.
14. Wu Z, Song S, Khosla A, et al. 3d shapenets: A deep representation for volumetric shapes[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 1912-1920.
15. Cheng X, Zhang N, Yu J, et al. Null-Space Diffusion Sampling for Zero-Shot Point Cloud Completion[C]//IJCAI. 2023: 618-626.
16. Pan L, Wu T, Cai Z, et al. Multi-view partial (mvp) point cloud challenge 2021 on completion and registration: Methods and results[J]. arXiv preprint arXiv:2112.12053, 2021.
17. Yi L, Gong B, Funkhouser T. Complete & label: A domain adaptation approach to semantic segmentation of lidar point clouds[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021: 15363-15373.

18. Wang H, Liu Q, Yue X, et al. Unsupervised point cloud pre-training via occlusion completion[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2021: 9782-9792.
19. Zhao X, Zhang B, Wu J, et al. Relationship-based point cloud completion[J]. IEEE transactions on visualization and computer graphics, 2021, 28(12): 4940-4950.
20. Fan Z, He Y, Wang Z, et al. Reconstruction-aware prior distillation for semi-supervised point cloud completion[J]. arXiv preprint arXiv:2204.09186, 2022.
21. Geiger A, Lenz P, Urtasun R. Are we ready for autonomous driving? the kitti vision benchmark suite[C]//2012 IEEE conference on computer vision and pattern recognition. IEEE, 2012: 3354-3361.
22. Zhu L, Liu D, Lin C, et al. Point cloud registration using representative overlapping points[J]. arXiv preprint arXiv:2107.02583, 2021.
23. Chang A X, Funkhouser T, Guibas L, et al. Shapenet: An information-rich 3d model repository[J]. arXiv preprint arXiv:1512.03012, 2015.
24. Chen X, Kundu K, Zhu Y, et al. 3d object proposals for accurate object class detection[J]. Advances in neural information processing systems, 2015, 28.
25. Yu X, Rao Y, Wang Z, et al. Pointr: Diverse point cloud completion with geometry-aware transformers[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2021: 12498-12507.
26. Lang A H, Vora S, Caesar H, et al. Pointpillars: Fast encoders for object detection from point clouds[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 12697-12705.