sensors

MDPI

Article

Joint Object Detection and Re-Identification for 3D Obstacle Multi-Camera Systems †

Irene Cortés [1,*], Jorge Beltrán [2], Arturo de la Escalera [1] and Fernando García [1]

[1] Department of Systems Engineering and Automation, Universidad Carlos III de Madrid (UC3M) 28911 Madrid, Spain; escalera@ing.uc3m.es (A.d.l.E.); fegarcia@ing.uc3m.es (F.G.)
[2] Department of Signal Theory, Telematics, and Computer Science, Rey Juan Carlos University (URJC), 28922 Madrid, Spain; jorge.beltran@urjc.es
* Correspondence: irecorte@ing.uc3m.es
† This paper is an extended version of our paper published in 2020 at the IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, 19 October–13 November 2020.

Abstract: The growing on-board processing capabilities have led to more complex sensor configurations, enabling autonomous car prototypes to expand their operational scope. Nowadays, the joint use of LiDAR data and multiple cameras is almost a standard and poses new challenges for existing multi-modal perception pipelines, such as dealing with contradictory or redundant detections caused by inference on overlapping images. In this paper, we address this last issue in the context of sequential schemes like F-PointNets, where object candidates are obtained in the image space, and the final 3D bounding box is then inferred from point cloud information. To this end, we propose the inclusion of a re-identification branch into the 2D detector, i.e., Faster R-CNN, so that objects seen from adjacent cameras can be handled before the 3D box estimation takes place, removing duplicates and completing the object's cloud. Extensive experimental evaluations covering both the 2D and 3D domains affirm the effectiveness of the suggested methodology. The findings indicate that our approach outperforms conventional Non-Maximum Suppression (NMS) methods. Particularly, we observed a significant gain of over 5% in terms of accuracy for cars in camera overlap regions. These results highlight the potential of our upgraded detection and re-identification system in practical scenarios for autonomous driving.

Keywords: 3D object detection; multi-camera setup; Siamese network; non-maxima suppression

Citation: Cortés, I.; Beltrán, J.; de la Escalera, A.; García, F. Joint Object Detection and Re-Identification for 3D Obstacle Multi-Camera Systems. *Sensors* **2023**, *23*, 9395. https://doi.org/10.3390/s23239395

Academic Editors: Iván García Daza, Javier Alonso Ruiz, Carlota Salinas and Rubén Izquierdo

Received: 9 October 2023
Revised: 12 November 2023
Accepted: 22 November 2023
Published: 25 November 2023

1. Introduction

In the previous decade, research on perception systems for autonomous driving was mainly focused on understanding the traffic situation in front of the vehicle, mainly powered by popular datasets such as KITTI [1] or Cityscapes [2]. Although this approach limits the complexity of the problem and the computational requirements, its outcome restricts the set of use cases to simple scenarios where an automated operation of the car can be safely performed (e.g., highways).

More recently, the increase in GPU capabilities and the great advances in deep learning models have led to the emergence of new research platforms and prototypes targeted to drive in more challenging traffic environments, with a higher degree of interaction with other road users and involving difficult maneuvers, like in cities [3,4]. In order to achieve this milestone, the perception pipeline of an autonomous vehicle must be able to identify the different participants and potential hazards in the whole scene, not only in the forward direction.

To keep pace with these new needs, more complex sensor configurations aimed at covering the whole horizontal field of view around the vehicle have become the prevailing trend. Nowadays, using setups composed of multiple cameras and one or more LiDAR

devices is a common choice to capture meaningful information in 360° with sensor redundancy, so that safe navigation can be achieved [5].

Apart from redundancy, the popularization of perception systems made of multiple units of different technologies brings many opportunities to build robust pipelines capable of providing a precise understanding of the driving environment. For example, combining the data from cameras and LiDAR devices allows for a better estimation of object positions, sizes, and velocities. Furthermore, fusing heterogeneous information from different modalities can help overcome each sensor's limitations, such as the incapability of cameras to perceive objects in low light situations or the sensitivity of LiDARs to adverse weather conditions like rain or fog.

However, all these advances also bring with them new challenges that must be addressed, such as extrinsic calibrations between sensors of different types [6], the synchronization of all sensors to obtain information at known time instants [7], and the merging of data and detections between sensors of the same and different types [8,9]. Nevertheless, the potential benefits of these perception systems for autonomous driving, including increased safety and efficiency, are promising.

In this paper, we propose a solution to address the last mentioned challenge, which manages the entire space around the vehicle by merging detections in areas covered by more than one camera of the same type and would otherwise be detected in duplicate or with partial information. The work is based on a two-step 3D detection pipeline [10], which uses a 2D detection model in the camera modality to generate candidates so that the LiDAR cloud can be filtered into smaller regions of interest to estimate final 3D boxes efficiently. While the performance of this approach was generally robust in a variety of traffic situations, the architecture suffered when objects appeared on the overlapping areas of consecutive cameras due to truncated or duplicate detections.

To tackle this pitfall, the image detection network is modified to include the re-identification module presented in [8] as a third branch. In the second step, the point cloud subsets associated with those detections identified as belonging to the same obstacle are merged. That way, the filtered 3D information for each obstacle does not suffer truncations due to limits in the horizontal field of view (HFOV) of the cameras. As a result, the proposed pipeline ensures that the estimation of the parameters of every obstacle is based on its complete representation in the 3D space, avoiding the inference of a single instance from multiple partial views.

Therefore, the main contribution of this paper is the addition of a re-identification branch to the image detector of a sequential multi-modal pipeline which:

- Shares features with the existing branches, allowing for seamless end-to-end training and improving the overall 2D results.
- Adds no extra computational load due to its parallel multitask design.
- Allows for the early elimination of false positives by detecting duplicate objects.
- Enhances 3D object detection by combining point clouds to reduce truncations.

The remainder of this paper is organized as follows. In Section 2, a brief review of related work is provided. Section 3 presents a general overview of the proposed algorithm. Section 4 provides experimental results that assess the performance of the method. Finally, conclusions and open issues are discussed in Section 5.

2. Related Work

Several significant advancements have been noted in recent years in the field of 3D obstacle detection with multi-camera systems. The first topic examines the evolution of in-vehicle object detection datasets. Compared to older datasets, like KITTI, newer ones provide detailed annotations under diverse conditions. The next area tackles the challenge of managing multiple detections of a single object. Several techniques, from improved non-maximal suppression methods to the innovative deterministic point process, have been developed to address this. The final section explores Siamese networks' role in re-identifying obstacles, a vital task when dealing with views from multiple cameras. These

networks have shown their value by distinguishing small differences among objects in crowded settings.

2.1. In-Vehicle Object Detection Datasets

The pursuit for greater automation levels in the automotive industry has led to a demand for richer annotated datasets that allow perception systems to cope with more complex traffic scenarios.

Compared to the classic KITTI benchmark, with daytime-only frames and focused on the forward direction, recent datasets have multiple cameras and LiDARs, recorded in both day and night situations, in rain or fog. Datasets like Waymo [11] consist of data from five LiDARs and five cameras, annotated both in 2D and 3D. Argoverse [4] features two LiDARs and seven cameras with 3D box annotations. PandaSet [12] combines data from six cameras and two LiDARs, providing annotations per LiDAR point and 3D box. Similarly, the nuScenes dataset [3] integrates one LiDAR, six cameras, and multiple radars, offering annotated 3D boxes for each object. These datasets reflect a growing trend toward more complex and information-rich perception systems, leveraging multiple sensors to capture a more complex, complete, and detailed view of the surrounding environment [13]. However, with the inclusion of multiple cameras comes the inherent challenge of efficiently handling and processing redundant and sometimes conflicting information between views. Duplicate detections, occlusions, and truncation of objects at the edges of the field of view are common problems that arise in multi-camera configurations. In addition, variability in lighting and environmental conditions, such as night, rain, or fog, adds another layer of complexity to data processing and analysis.

2.2. Elimination of Multiple Detections for the Same Element

Several papers have explored techniques related to reducing the number of detections for the same element. The suppression of redundant detections is essential to ensure accurate and consistent perception. In [14], a variant of the traditional Non-Maximal Suppression (NMS) method, called Soft-NMS, is proposed. Unlike traditional NMS, which discards detections based on a predefined threshold, Soft-NMS modifies detection scores continuously, resulting in improved detection accuracy. In crowded environments, pedestrian detection can be challenging due to multiple overlapping detections. In [15], this problem is addressed by dynamically adapting the NMS threshold according to the density of detections in the local region, enabling better discrimination between nearby pedestrians. The method presented in [16] focuses on learning pairwise relationships between detections to improve accuracy in crowded scenes. It can reduce false detections and improve discrimination between nearby objects by explicitly modeling these relationships. In [17], the use of a Deterministic Point Process (DPP) is introduced as an alternative to traditional NMS. DPP selects a subset of detections that are representative and diverse, which can be beneficial in crowded scenarios where detections overlap. In [18], they propose an alternative to NMS and IoU-based methods called “Confluence”. It aims at resolving the challenges associated with redundant detections in object detection tasks, by selecting the box that is closest to every other box within a given cluster and removing highly confluent neighboring boxes.

2.3. Siamese Networks for Obstacle Re-Identification

On multi-camera systems, the management of multiple detections of a single object needs to be handled in a different fashion, as NMS techniques cannot be applied to bounding boxes belonging to images captured from different perspectives (e.g., distributed along the roof of the car). Even so, there are usually parts of the environment that are seen from two cameras, due to an overlap between their horizontal fields of view. For specific situations where the cameras are in close proximity to each other, methods such as image-stitching can be used to form a large panoramic image composed of images from several cameras [9,19,20]. Also, new architectures that take as input a number of images and fuse

their information internally have been introduced [21,22]. In a more general approach, this problem is often solved through the use of agent re-identification networks, which get rid of redundant detections based on feature similarities [8,23].

Siamese networks are a powerful tool in object re-identification, especially in scenarios with multiple obstacles and crowded environments. In [24], Siamese networks are used to learn to recognize objects from a single example by leveraging the network structure to compare images. This capability is essential in scenarios where data collection is costly or impractical. Another example is [25], where this type of network is applied to differentiate between similar vehicles. By learning relative distances, the network is able to identify subtle differences between vehicles that, at first glance, appear identical. This technique is especially useful in applications, such as urban surveillance, where accurate vehicle re-identification is essential. Examples of the use of Siamese networks can be seen in [26,27] or [28], where re-identification is approached from a "parts" perspective. Instead of treating the person as a whole, the network is trained to recognize and compare individual parts (such as head, torso, legs), allowing for greater robustness against varied occlusions and postures.

3. Proposed Approach

In this paper, we attempt to address the issues identified in the two-step multicamera–lidar detection system presented in [10] by endowing the image detector with a new re-identification branch to better handle duplicate or truncated detections by the HFOV of the cameras.

In the selected pipeline a 3D obstacle detection is performed in two steps; the first one consists of a generation of image proposals, obtained with the faster R-CNN network [29,30] in each of the vehicle's cameras.

Subsequently, by means of the extrinsic calibration parameters, a set of frustums will be obtained by filtering the LiDAR point cloud with each of the detected bounding boxes, which will be used as input to the second step, the Frustum PointNets network [31], obtaining a 3D box for each of the proposed obstacles. A complete outlook of the whole pipeline can be seen in Figure 1.

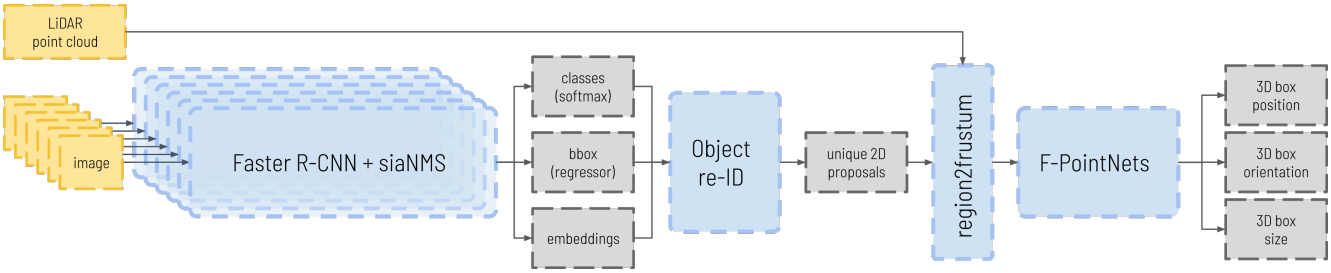


Figure 1. Complete detection pipeline. Each of the n images coming from the n cameras is introduced into an image detection network (Faster R-CNN) modified to include the proposed branch. The classes, bbox, and embeddings are obtained from the images and introduced to the re-identification module, where the embeddings are compared and a set of unique proposals is obtained. In the *region2frustum* module, the proposals are used to filter the Light Detection and Ranging (LiDAR) point cloud, taking into account the detections that are shared between cameras. Finally, the filtered point clouds obtained for each detection are passed through the Frustum PointNets network to obtain the parameters of the 3D box.

Ideally, all proposals would be unique for each obstacle and would cover it completely, surrounding the whole area of the object visible from the ego-car. Unfortunately, this is not feasible, partly because of the possible failures of the detector in the image, but also due to the limited horizontal field of view (HFOV) of the cameras, which will generate truncations in the detections and possible duplicates in obstacle detections that appear in several contiguous cameras at the same time.

To tackle this, this work integrates the siaNMS module presented in [8], allowing the end-to-end training of the model. In addition, incorporating the module as a branch of the original network ensures that the learned feature map encodes meaningful characteristics for the three tasks: class prediction, bounding box regression, and re-identification. Furthermore, utilizing this information when filtering the point cloud facilitates obtaining a single frustum per obstacle by combining the matched regions of interest from both images.

3.1. Re-Identification Branch Description

The objective of the re-identification branch introduced in the detection network is to obtain an embedding that encodes the information obtained from each obstacle. The network is trained in such a way that the embeddings generated for the different detections of the same object are very similar while maximizing the differences between the detections of distinct objects.

For this purpose, we start from a faster R-CNN detection network structure, composed of a convolutional backbone, from which regions of interest are obtained by means of a Region Proposal Network (RPN) stage. The encoded tensor is cut using these candidates and an ROI align layer is used to obtain a feature map of constant size for each proposal. These maps are fed to the Fully Connected (FC) layers stage, from which we obtain the outputs of the network: in the case of the original network, that was the class of the object and its bounding box.

In this paper, a third output branch is added and tailored for re-identification. This branch is also fed with the fixed-size feature maps obtained for each network proposal and consists of a series of convolutional and FC layers that compress the information of each obstacle into an embedding, with fixed output dimensions, d. A schematic of the modified network structure can be seen in Figure 2.

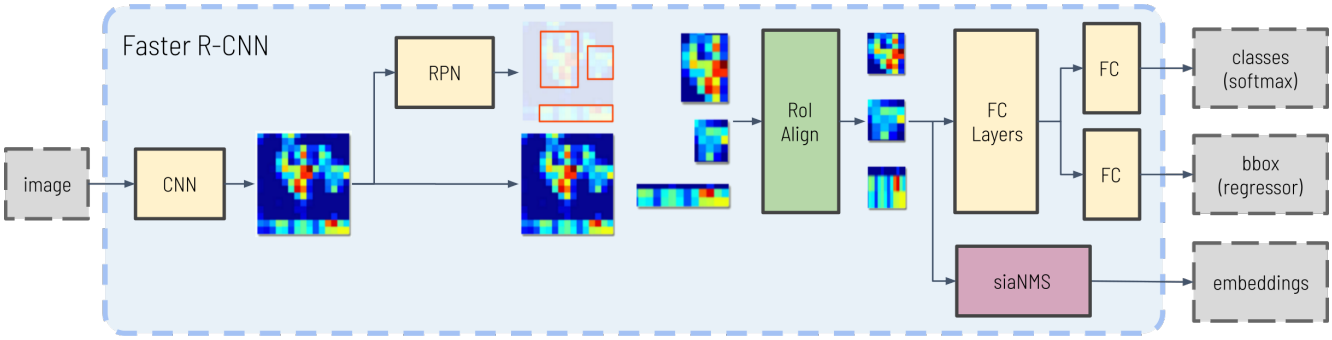


Figure 2. Image detection network.

3.2. Training Data Organization

In the original 2D stage, a set of annotated images is passed as input in a randomized order through the network (grouped in batches) until the entire data set (epoch) is completed. This is repeated as many times as necessary until the network is sufficiently trained. Although this is the standard procedure for conventional image detectors, this way of ordering the data is unsuitable for the purpose of this paper since we need the same object to appear several times in the same forward pass of the network so that the re-identification branch can be trained.

Intuitively, in order to train the class and bounding box branches, we need the annotations of the obstacles present in each image. However, to train the re-identification branch, the type of annotation needed is different, namely which pairs of annotations correspond to the same obstacle and which do not. Therefore, in each batch of images, we need examples of both negative pairs of detections, which do not correspond to the same object, and positive ones, which are detections of the same object.

In this study, we have used the nuScenes dataset, composed of six cameras distributed in such a way that they cover 360º around the car. Taking advantage of this, we have trained

the network using batches of six images, one corresponding to each of the surrounding cameras. This way, the input of the model during the training process matches the one that will be used in the inference, while objects appearing in regions shared by contiguous cameras can serve as positive pairs for the training. That said, the order of time instants for the six-image batches is randomized, mimicking the behavior of traditional CNN training.

3.3. Network Training and Loss Function Definition

During training, the process explained below is performed for each batch of images. First, a forward pass is made, and with the results obtained at the output of the class and box branches, a loss is calculated for each image individually:

$$\mathcal{L}_{box\_head} = \sum_{i=0}^{F} \mathcal{L}_{box\_reg,i} + \sum_{i=0}^{F+B} \mathcal{L}_{cls,i}, \tag{1}$$

where F and B are the numbers of foreground and background detections in that image, $\mathcal{L}_{box\_reg,i}$ is the Smooth-L1 Loss, and $\mathcal{L}_{cls,i}$ is the cross-entropy loss.

Once the loss of the detections of the batch images has been acquired, the loss of re-identification between the detections of the six images is calculated. The total loss of the batch is the sum of all losses of the individual images plus the re-identification loss:

$$\mathcal{L}_{batch} = \mathcal{L}_{reID} + \sum_{n=0}^{N} \mathcal{L}_{box\_head,n}, \tag{2}$$

where N is the number of images in the batch, six for this case. This way, the gradients calculated will take into account the information on the three kinds of output generated by the network.

To calculate the re-identification loss, all possible pairs between the foreground detections (those with an IoU > 0.7 with the ground truth) are obtained, that is, all the possible combinations between the detections of the batch images. The loss will be the sum of the individual losses of all positive pairs and the same number of negative pairs. To select which negative pairs are used, their losses are sorted in decreasing order and the ones with the highest loss value are chosen, following an Online Hard Example Mining (OHEM) technique, as presented in [32],

Then, the double margin contrastive loss [28] is calculated for each pair, as explained in [8]:

$$\mathcal{L}_{reID} = \frac{1}{2} \sum_{i}^{N} \left[\max\left(\left\| f(x_i^r) - f\left(x_i^p\right) \right\|_2 - \alpha, 0 \right)^2 + \max\left(\beta - \|f(x_i^r) - f(x_i^n)\|_2, 0\right)^2 \right], \tag{3}$$

where α and β are the two constant margins and $f(x_i^r)$, $f\left(x_i^p\right)$, and $f(x_i^n)$ are the embeddings of the reference object, their positive pair, and the hardest negative pair, respectively.

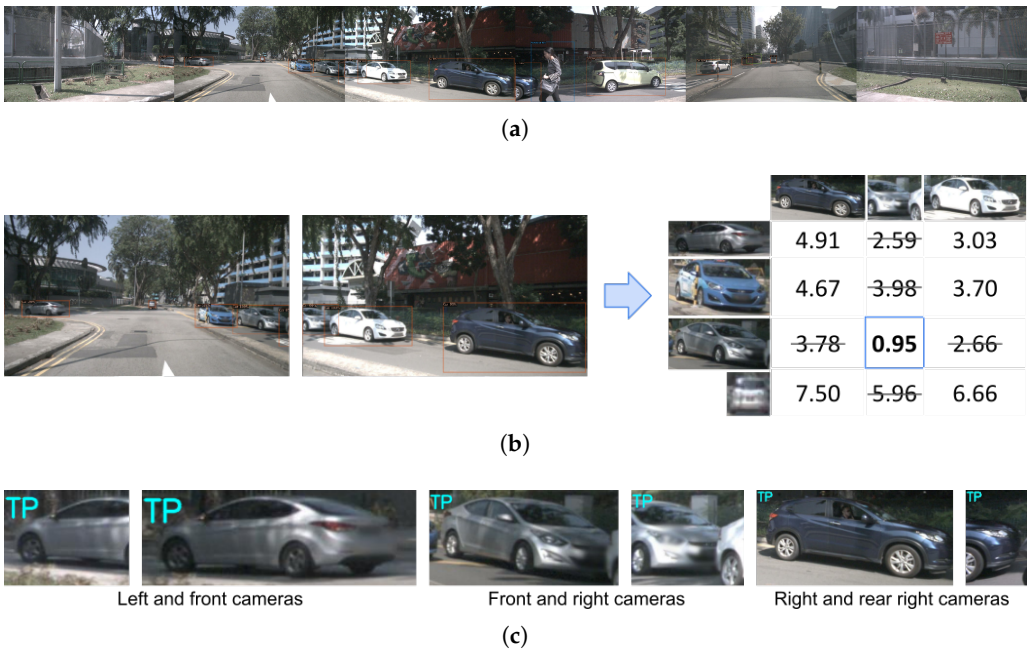
3.4. Re-Identification Evaluation

To evaluate the quality of the re-identifications at the image stage, an algorithm has been developed. It performs the following steps:

1. For each group of images at each instant, the detections in contiguous images are compared. This distinction is made to speed up the evaluation procedure since there is no possibility of re-identifiable detections between two images that are not adjacent.
2. For each pair of images, all possible combinations between detections of the same class are compared. In this case, no additional geometric constraints are imposed, since we want to test the performance of the embedding generation module and thus the challenge is more significant.
3. A matrix of distances between the comparable detected objects is calculated (having eliminated the pairs of different classes in the previous step) and the pairs whose distance is below a threshold are chosen following the Hungarian method.

4. A comparison of whether the obtained pairs actually correspond to the same object according to the ground truth is made, and the statistics of TP, TN, FP, and FN are elaborated.

A visual example of this process can be seen in Figure 3.



(**a**)

(**b**)

(**c**)

Figure 3. Re-identification evaluation process step-by-step. (**a**) The six images of the same instant, with the detections obtained in each image. (**b**) Obtained distance matrix between the detections of the front and right cameras. (**c**) Re-identified pairs made in the six images that have been marked as True Positives (TP).

4. Experimental Results

For evaluation, the nuScenes dataset has been used, as it provides suitable sensor configuration and annotations for assessing the performance of the presented approach—in the 2D space, to validate the re-identification capabilities of the networks and its effect on the baseline model and in the 3D space, so that it can be compared to traditional alternatives. To accommodate the original label for the 2D experiments, the minimum bounding boxes have been obtained from the projection of the 3D boxes in each of the cameras. Moreover, the unique ID information of each obstacle has been kept to allow subsequent re-identification validation.

For the experiments of the first part of the configuration, those concerning the 2D detection in the image, the metrics of the KITTI object [1] dataset will be used. The minimum overlap percentage between detections and annotations required to consider an object as detected is 50% for all classes and the difficulty level selected is moderate (minimum bounding box height: 25 Px, maximum occlusion level: partly occluded, maximum truncation: 30%). Although only the three most commonly used classes (pedestrian, car, and cyclist) are shown in the tables, the evaluation is performed for all the classes of the nuScenes dataset, and the *all* column is calculated with the weighted average of all of them. The reID columns have been obtained following the procedure explained in the previous Section 3.4.

In order to evaluate the performance of the whole pipeline after the integration of the re-identification module, the official nuScenes Detection Benchmark metrics are used.

4.1. Ablation Studies: Number of Embedding Dimensions

First, an analysis of the adjustment of the hyperparameters of the network has been made, choosing the original configuration of the siaNMS network [8] but varying the

number of dimensions of the output. Results are obtained for the network detection output and the re-identification branch for each of the output dimensions. In addition, we compare the results with those obtained in an equivalent training but without the re-identification branch included. This experiment has been performed with 1000 pre-NMS proposals and 500 post-NMS proposals in training, and 500 pre- and post-NMS proposals in the test stage. The results are shown in Table 1. The KITTI object image detection metrics have been used for all nuScenes classes (pedestrian, car, cycle, cyclist, bus, truck, construction vehicle, trailer, barrier, and cone) for the three frontal cameras of the validation split of the nuScenes dataset.

Table 1. Evaluation results for the proposed approach while varying the output layer dimension of the embedding branch. The best result for each column is highlighted in bold.

Num Dims	Classes (AP [%])				reID		
	Ped	Car	Cyclist	All	Precision	Recall	F1-Score
off	**58.95**	71.10	43.16	57.627	-	-	-
10	58.55	70.99	45.38	57.627	0.757	0.793	0.775
15	58.60	71.11	45.58	57.616	0.788	0.786	0.787
20	58.78	71.05	44.49	57.608	0.797	0.788	0.792
25	58.68	71.08	44.54	57.560	0.792	**0.802**	0.797
30	58.63	71.11	44.62	57.632	0.797	0.799	0.798
40	58.53	71.18	45.02	**57.648**	0.794	0.794	0.794
50	58.70	71.11	**45.59**	57.584	**0.810**	0.790	**0.800**

The *Cycle* and *Cyclist* classes have been obtained by selecting the nuScenes *Bicycle* and *Motorcycle* classes and combining them with the *with_rider* attribute. This way the *Bicycle* and *Motorcycle* objects with rider are *Cyclists*, and the ones without rider are *Cycles*.

As can be seen in Table 1, the results of both the re-identification branch and the detection and classification branches vary slightly depending on the output size of the re-ID branch. This is because the training of the network is conducted in an end-to-end manner, and the ability of the network to generalize is altered by the introduction of the third branch. That said, it can be generally observed that the inclusion of the obstacle re-identification branch does not harm the network's ability to detect and classify obstacles. In fact, some of the obtained results slightly improve the object detection accuracy with respect to the baseline, which implies that the quality of the feature maps obtained in the intermediate layers of the network is improved by its optimization for a complementary task, as previously observed in the literature [33,34]. The best result for each class is marked in bold, and it can be seen that in most cases, the architectures that include the re-identification branch achieve better results than those of the original architecture. Likewise, we see that the re-identification results for all cases are similar and relatively good.

4.2. Ablation Studies: Re-Identification Branch Configuration

In the next experiment, we study the effect of changes in the structure of the re-identification branch. For this purpose, several alternatives have been proposed, with different numbers of neurons and intermediate layers, or eliminating altogether the convolutional layers and keeping only the fully connected ones, in a similar way as the other two output branches are constructed. Table 2 shows the configuration of all the studied options. Finally, Table 3 shows the detection and re-identification results for each design. For all these configurations, a constant number of output dimensions, 25, has been chosen so that the results are comparable. For the following experiment, 500 proposals during the test phase have been chosen. Once more, the evaluation has been performed following the KITTI object image detection metrics for all nuScenes classes, but this time the six cameras of the validation split of the nuScenes dataset have been used.

Table 2. Specifications of the tested configurations for the re-identification branch. k is the kernel size used, C1–C4 is the number of output neurons of the convolutional layer, FC1–FC3 is the number of output neurons of the fully connected layer.

Conf.	#0	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15
k	3	3	5	5	5	5	3	3	3	3	3	-	-	-	-	-
C1	32	16	32	16	64	32	32	32	32	32	16	-	-	-	-	-
C2	64	32	64	32	128	64	64	64	64	64	-	-	-	-	-	-
C3	-	-	-	-	-	-	-	64	128	-	-	-	-	-	-	-
C4	-	-	-	-	-	-	-	128	256	-	-	-	-	-	-	-
FC1	1024	4096	1024	4096	512	4096	4096	1024	1024	512	512	1024	1024	1024	1024	1024
FC2	256	1024	256	1024	256	1024	1024	256	256	256	256	1024	1024	512	512	256
FC3	256	256	256	256	256	256	256	256	256	128	128	256	-	128	-	-

Table 3. Results of the evaluation of the different configurations for the re-identification branch within the image detection network. Details of each configuration are given in Table 2. The best result for each column is highlighted in bold.

Conf.	Classes (AP [%])				reID		
	Ped	Car	Cyclist	All	Precision	Recall	f1-Score
#0	58.80	71.40	43.60	57.28	0.803	**0.804**	0.803
#1	58.69	71.40	43.30	57.37	0.802	0.790	0.796
#2	58.83	71.36	43.53	57.20	0.800	0.792	0.796
#3	58.70	71.34	42.89	57.20	0.796	0.803	0.799
#4	58.68	71.31	43.62	57.25	0.819	0.793	0.806
#5	58.79	71.36	43.53	57.27	0.808	0.797	0.802
#6	58.77	71.44	43.83	57.37	0.811	**0.804**	0.807
#7	58.70	71.44	43.94	57.36	0.724	0.795	0.758
#8	58.74	71.37	43.11	57.29	0.717	0.801	0.757
#9	58.78	71.40	43.40	57.33	0.804	0.798	0.801
#10	58.66	71.37	43.34	57.32	0.799	0.802	0.800
#11	58.76	**71.64**	**46.33**	**57.53**	**0.844**	0.789	0.816
#12	**58.86**	71.53	43.34	57.45	0.840	0.784	0.811
#13	58.67	71.51	43.20	57.34	0.838	0.797	0.817
#14	58.76	71.44	43.43	57.47	0.838	0.800	**0.819**
#15	58.68	71.49	43.38	57.49	0.840	0.789	0.814

Based on the reference configuration #0, as shown in Table 3, variations made in the other configurations have a slight effect on the obtained results. Although the alterations in the network composition yield minor performance changes, configuration #11 presents the best overall results, achieving the highest precision for cars (71.64%), cyclists (46.33%), and all classes combined (57.53%), and a re-ID precision (0.844) in pair with the best case scenario.

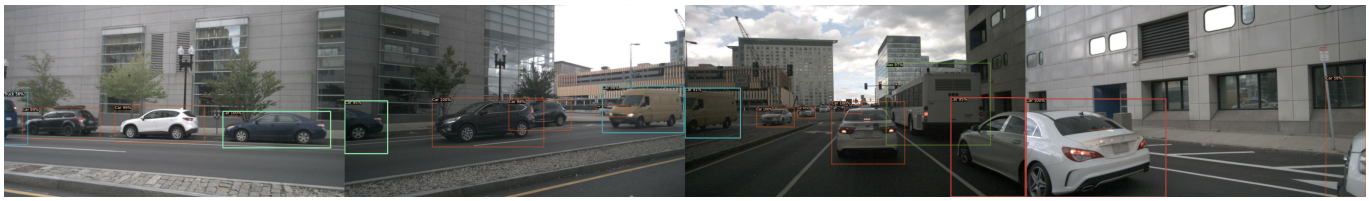
4.3. Image Detection Qualitative Results

To visually assess the operation of the pipeline, some qualitative results in the nuScenes benchmark are presented. The model used follows the configuration #11 from Table 2 with an output dimension of 30, using 500 proposals in the RPN stage. For the remaining experiments, these hyperparameters are kept.

Some examples of detection and re-identification results are shown in Figure 4. The images show frames captured by the different cameras at the same temporal instant. It can be seen how obstacle detection, even at long distances, is appropriately conducted and how the obstacles appearing in adjacent cameras are re-identified (the bounding box of the re-identified obstacles is drawn with the same color, chosen randomly).

(**a**)

(**b**)



(**c**)

Figure 4. Qualitative examples of the image detections (**a**–**c**). The same color bounding box indicates that the two obstacles from different cameras have been re-identified as the same. For the rest of the bounding boxes: orange for cars, blue for pedestrians, and purple for cyclists.

4.4. Selected Hyperparameters

From the results obtained in Table 3, the configuration producing the highest mAP, #11, is selected. For this re-identification branch model, we then analyze its effect on the two best-performing output layers for re-identification purposes (F1 score), 30 and 50, as shown in Table 1. The evaluation of these two configurations can be seen in Table 4. Based on these results, configuration #11 with 30 dimensions at the output has been selected. This configuration is used in the following experiments.

Table 4. Evaluation results for the final hyperparameters. The best result in each column is marked in bold.

Num Dims	Classes (AP [%])				reID		
	Ped	Car	Cyclist	All	Precision	Recall	F1-Score
30	58.72	**71.50**	**43.58**	**57.47**	**0.880**	**0.793**	**0.816**
50	**58.79**	71.45	43.42	57.39	0.842	0.776	0.808

4.5. Three-Dimensional Obstacle Detection Quantitative Results

In this section, we evaluate the second stage of the proposed detection pipeline, which involves taking as input a filtered point cloud in the form of a frustum for each of the objects detected in the previous image-based detection stage and processing them through a series of pointnet-based networks to estimate the parameters defining a three-dimensional bounding box—position (x, y, z), dimension (l, w, h), and orientation (θ)—that encapsulates the detected obstacle in the image space.

As introduced in [8], to leverage the outcome of the re-identification step in the 3D box estimation, the *region2frustum* operation shown in Figure 1 needs to be adapted so that it can produce more refined filtered point cloud candidates from matched detections across adjacent cameras:

1. For detections appearing in a single camera, we follow the process outlined in the original F-PointNets paper. A projection of the point cloud onto the bounding box of the obstacle is performed, and the points within the box are selected.
2. For detections appearing in two cameras simultaneously, we employ a union of the filtered point clouds from each camera, $P_i \cup P_j$, where P_i, P_j are the corresponding point sets with two detections re-identified as the same object. The process involves the following steps:
   (a) We first obtain the filtered frustum for each individual detection, following the same procedure as for unique detections.
   (b) Next, we verify if the point clouds obtained for each instance have common points.
   (c) If there are no shared points, it is considered a false positive re-identification, and the merging of objects is discarded.
   (d) If shared points exist, the match is considered spatially coherent, indicating a correct re-identification. The two filtered point clouds are merged, retaining only the unique points.
   (e) Finally, the central axis of the detection is computed by calculating the average angle between the two outermost angles from the detections in both cameras. A deeper explanation of this process can be read in [8].

To evaluate the results, the proposed method (*siaNMS*) is studied against three other approaches. The *Original* configuration, namely the results obtained by the baseline F-PointNets model with the 2D detection network unaltered. A second one uses the detection network presented in this article, but without carrying out the union of re-identified objects (*2D+embedding*). A third pipeline, where a Non-Maximum Suppression (NMS) algorithm is applied to the results obtained from the Original configuration (*Original+NMS*), such as the one presented in the original paper [8].

The assessment encompasses the different metrics provided by the nuScenes dataset, considering various aspects such as positional error, size error, and orientation error, among others, for all annotated types of obstacles. More details can be found in the nuScenes object detection task [3].

Table 5 shows the results for Average Precision (AP ↑) (%), Average Translation Error (ATE ↓) (m), Average Size Error (ASE ↓) (%), and Average Orientation Error (AOE ↓) (rad) for the car, pedestrian, and cyclist classes. Similarly, Table 6 shows the averaged results across all classes. The experiments were conducted in the following manner: The Frustum Pointnets detection network was trained with the frames of the training split of the nuScenes dataset and the validation split of the nuScenes dataset was used for evaluation. To better understand the impact of the proposed approach, the same evaluation has been performed taking into account only the object instances that appear in more than one camera.

Table 5. Comparison of the 3D object detection performance on the nuScenes validation set in different regions of interest. Results for Car, Pedestrian, and Cyclist classes. The best result for each column is highlighted in bold.

Area		Car				Pedestrian				Cyclist			
		AP	**ATE**	**ASE**	**AOE**	**AP**	**ATE**	**ASE**	**AOE**	**AP**	**ATE**	**ASE**	**AOE**
all	Original	48.2	0.434	17.1	0.493	58.4	0.263	28.5	1.194	44.2	0.252	26.1	0.898
	2D+embedding	47.8	0.427	17.1	0.486	57.1	0.259	**28.4**	1.200	43.9	0.247	26.4	0.890
	Original + NMS	48.7	0.436	17.1	0.487	56.0	0.259	28.5	**1.188**	**46.6**	0.257	**26.1**	0.903
	siaNMS	**51.5**	**0.422**	17.1	**0.476**	**59.7**	**0.246**	28.5	1.203	46.3	**0.243**	26.2	**0.874**

Table 5. *Cont.*

Area		Car				Pedestrian				Cyclist			
		AP	**ATE**	**ASE**	**AOE**	**AP**	**ATE**	**ASE**	**AOE**	**AP**	**ATE**	**ASE**	**AOE**
overlap	Original	41.7	0.439	16.8	0.477	44.2	0.355	28.4	1.136	34.0	0.301	26.7	0.843
	2D+embedding	41.6	0.433	**16.7**	0.463	43.5	0.359	**28.3**	**1.133**	32.5	0.288	26.4	0.799
	Original + NMS	44.8	0.442	16.8	0.472	47.9	0.345	28.4	1.138	35.7	0.311	26.3	0.876
	siaNMS	**50.1**	**0.416**	**16.7**	**0.435**	**50.2**	**0.330**	28.6	1.148	**41.5**	**0.261**	**26.0**	**0.597**

Table 6. Comparison of the 3D object detection performance on the nuScenes validation set in different regions of interest. Results for all classes. The best result for each column is highlighted in bold.

Area		AP	ATE	ASE	AOE
all	Original	32.87	0.526	29.81	0.928
	2D+embedding	31.87	0.525	29.72	0.930
	Original + NMS	32.46	0.529	29.75	0.925
	siaNMS	**33.18**	**0.521**	**29.58**	**0.915**
overlap	Original	25.59	0.585	30.15	0.947
	2D+embedding	24.93	0.588	29.93	0.929
	Original + NMS	26.90	0.583	29.99	0.954
	siaNMS	**28.40**	**0.573**	**29.80**	**0.899**

As can be seen in Tables 5 and 6, the AP results of all classes improve when using the method proposed in the article in contrast to conventional methods. For the car class, the average accuracy increases by more than 3% with respect to the original method, more than 1% for pedestrians, and more than 2% for cyclists. Although these numbers may not suggest a major effect of the proposed additional re-identification branch, its real impact is being diluted by the fact that only a reduced portion of objects leads to redundant detections and is affected by truncation.

As a consequence, when the results are analyzed considering only those objects falling in the regions of overlap between contiguous cameras, it can be observed that these improvements are much more accentuated, especially in bulky categories such as cars—more prone to be seen from multiple views—where the gain in AP goes up to 8.4%. In addition to improving the number of detections, the quality of the detections is also affected, reducing the average errors of position, size, and orientation of all classes with respect to the original method and practically all with respect to a conventional NMS method.

4.6. Three-Dimensional Object Detection Qualitative Results

Similar to the process of visualizing the impact of the proposal in the image space, this section includes qualitative 3D results of the final system compared to the ones of the reference framework. Here the detections are presented in the LiDAR bird's-eye view, which displays the LiDAR readings as seen from an orthographic top view, where each cell represents a square pillar lying in a theoretical ground plane. The images in Figure 5 show some frames from the validation split of the nuScenes dataset comparing the two best approaches (i.e., *Original+NMS* and *siaNMS*). In green are the ground truth boxes, while the boxes obtained by the full pipeline proposed in this article are painted in blue. As can be seen in the three cases, the number of false positives is reduced, and the quality of the detections is improved in terms of positioning and orientation of the obtained boxes.

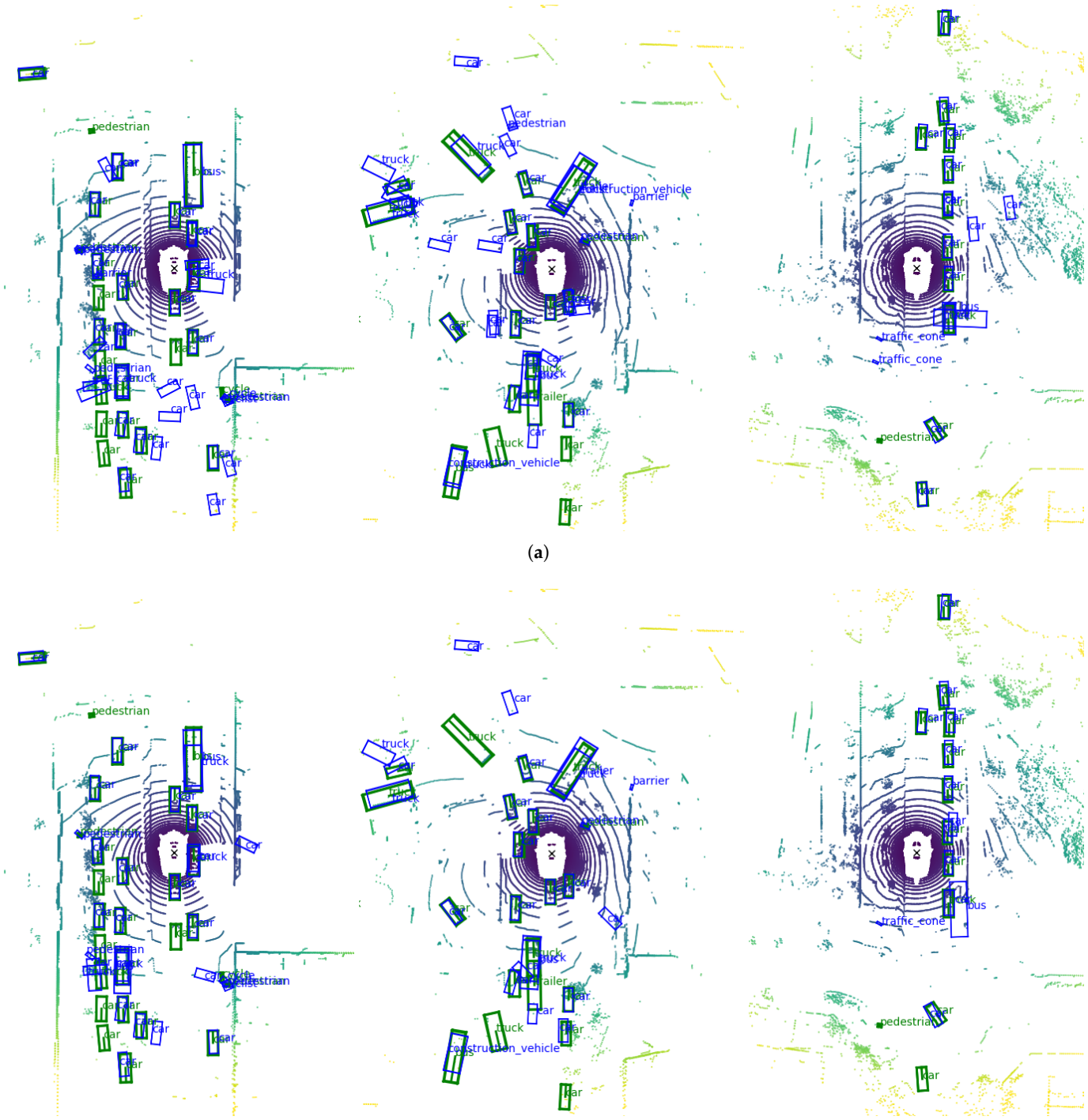


Figure 5. Results on nuScenes validation set. In (**a**) the original approach (Non-Maximal Suppression (NMS)) is used, while in (**b**), the proposed architecture is used. The images show the Light Detection and Ranging (LiDAR) point cloud in a bird's-eye view. The color coding of the point cloud corresponds to the distance of each point from the coordinate origin of the cloud, with purple indicating the points closest to the origin, transitioning through blues, greens, and yellows for points further away. The obstacles' ground truth boxes are depicted in green, while the detections of the proposed pipeline are shown in blue.

5. Conclusions

In this work, the capabilities of a well-established camera–LiDAR sequential fusion 3D object detection pipeline have been enriched to make it more suitable for its use in multi-

camera setups, where its performance deteriorates when duplicate detections of the same object appear. To this end, the 2D object detector module of its first stage has been extended by means of an embedding branch that enables the re-identification of instances across contiguous cameras. Unlike the predecessor study, the siaNMS block is now integrated as part of the decoding layers of the network in a multi-task fashion, allowing for its training in an end-to-end manner.

The presented approach significantly outperforms both the baseline model and the traditional NMS technique. Moreover, the addition of the new branch has led to an improvement in the quality of the features extracted by the encoder, enhancing the network detection and classification outcome, as demonstrated in the conducted analysis on the nuScenes dataset.

Further experimentation shows that the early elimination of redundant detections also benefits the results provided by the 3D box characterization model, reducing the negative effects of truncated boxes on the image plane and thus contributing to a significant gain in the overall performance of the system.

Building on the foundation laid by this research, we propose several avenues for future exploration: First, the application of the re-identification branch to different types of network architectures, including single-shot detectors and those using transformers. This would test the versatility and effectiveness of the re-identification branch across different state-of-the-art object detection frameworks. Another goal would be to integrate the embedding comparison process directly into the inference stage of the network. This would allow a single network to process multiple images simultaneously and output unique detections for all objects present in all views.

Author Contributions: Conceptualization, I.C. and J.B.; methodology, I.C. and J.B.; software, I.C.; validation, I.C. and J.B.; investigation, I.C.; resources, A.d.l.E. and F.G.; writing—original draft preparation, I.C.; writing—review and editing, I.C., J.B., A.d.l.E. and F.G.; visualization, I.C.; funding acquisition, A.d.l.E. and F.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been supported by the Spanish Government through the projects ID2021-128327OA-I00, PID2021-124335OB-C21, and TED2021-129374A-I00 funded by MCIN/AEI/10.13039/501100011033, by the European Union NextGenerationEU/PRTR.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: [https://www.nuscenes.org/nuscenes], accessed on 21 November 2023.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012; pp. 3354–3361.
2. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The Cityscapes Dataset for Semantic Urban Scene Understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 3213–3223.
3. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liong, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom, O. nuScenes: A multimodal dataset for autonomous driving. *arXiv* **2019**, arXiv:1903.11027.
4. Chang, M.F.; Lambert, J.; Sangkloy, P.; Singh, J.; Bak, S.; Hartnett, A.; Wang, D.; Carr, P.; Lucey, S.; Ramanan, D.; et al. Argoverse: 3D Tracking and Forecasting with Rich Maps. In Proceedings of the The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 8748–8757.
5. Janai, J.; Güney, F.; Behl, A.; Geiger, A. Computer vision for autonomous vehicles: Problems, datasets and state of the art. *Found. Trends Comput. Graph. Vis.* **2020**, *12*, 1–308. [CrossRef]
6. Beltrán, J.; Guindel, C.; García, F. Automatic extrinsic calibration method for lidar and camera sensor setups. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 17677–17689. [CrossRef]

7. Milanés, V.; González, D.; Navas, F.; Mahtout, I.; Armand, A.; Zinoune, C.; Ramaswamy, A.; Bekka, F.; Molina, N.; Battesti, E.; et al. The tornado project: An automated driving demonstration in peri-urban and rural areas. *IEEE Intell. Transp. Syst. Mag.* **2021**, *14*, 20–36. [CrossRef]
8. Cortés, I.; Beltrán, J.; de la Escalera, A.; García, F. siaNMS: Non-Maximum Suppression with Siamese Networks for Multi-Camera 3D Object Detection. In Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV), IEEE, Las Vegas, NV, USA, 19 October–13 November 2020; pp. 933–938.
9. Kinzig, C.; Cortés, I.; Fernández, C.; Lauer, M. Real-time Seamless Image Stitching in Autonomous Driving. In Proceedings of the 2022 25th International Conference on Information Fusion (FUSION), Linköping, Sweden, 4–7 July 2022; pp. 1–8.
10. Beltrán, J.; Guindel, C.; Cortés, I.; Barrera, A.; Astudillo, A.; Urdiales, J.; Álvarez, M.; Bekka, F.; Milanés, V.; García, F. Towards autonomous driving: A multi-modal 360 perception proposal. In Proceedings of the 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), Rhodes, Greece, 20–23 September 2020; pp. 1–6.
11. Sun, P.; Kretzschmar, H.; Dotiwalla, X.; Chouard, A.; Patnaik, V.; Tsui, P.; Guo, J.; Zhou, Y.; Chai, Y.; Caine, B.; et al. Scalability in Perception for Autonomous Driving: An Open Dataset Benchmark. *arXiv* **2019**, arXiv:1912.04838.
12. Xiao, P.; Shao, Z.; Hao, S.; Zhang, Z.; Chai, X.; Jiao, J.; Li, Z.; Wu, J.; Sun, K.; Jiang, K.; et al. Pandaset: Advanced sensor suite dataset for autonomous driving. In Proceedings of the 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), Indianapolis, IN, USA, 19–22 September 2021; pp. 3095–3101.
13. Wang, X.; Li, K.; Chehri, A. Multi-Sensor Fusion Technology for 3D Object Detection in Autonomous Driving: A Review. *IEEE Trans. Intell. Transp. Syst.* **2023**, *early access*. [CrossRef]
14. Bodla, N.; Singh, B.; Chellappa, R.; Davis, L.S. Soft-NMS–improving object detection with one line of code. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 5561–5569.
15. Liu, S.; Huang, D.; Wang, Y. Adaptive nms: Refining pedestrian detection in a crowd. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 6459–6468.
16. Liu, Y.; Liu, L.; Rezatofighi, H.; Do, T.T.; Shi, Q.; Reid, I. Learning pairwise relationship for multi-object detection in crowded scenes. *arXiv* **2019**, arXiv:1901.03796.
17. Some, S.; Gupta, M.D.; Namboodiri, V.P. Determinantal point process as an alternative to NMS. *arXiv* **2020**, arXiv:2008.11451.
18. Shepley, A.J.; Falzon, G.; Kwan, P.; Brankovic, L. Confluence: A robust non-IoU alternative to non-maxima suppression in object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 11561–11574. [CrossRef] [PubMed]
19. Xiang, T.; Xia, G.S.; Zhang, L. Image stitching with perspective-preserving warping. *arXiv* **2016**, arXiv:1605.05019.
20. Lin, M.; Xu, G.; Ren, X.; Xu, K. Cylindrical panoramic image stitching method based on multi-cameras. In Proceedings of the 2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), Shenyang, China, 8–12 June 2015; pp. 1091–1096.
21. Jiang, Y.; Zhang, L.; Miao, Z.; Zhu, X.; Gao, J.; Hu, W.; Jiang, Y.G. Polarformer: Multi-camera 3D object detection with polar transformer. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; Volume 37; pp. 1042–1050.
22. Hazarika, A.; Vyas, A.; Rahmati, M.; Wang, Y. Multi-camera 3D Object Detection for Autonomous Driving Using Deep Learning and Self-Attention Mechanism. *IEEE Access* **2023** , *11*, 64608–64620. [CrossRef]
23. He, S.; Luo, H.; Wang, P.; Wang, F.; Li, H.; Jiang, W. Transreid: Transformer-based object re-identification. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 15013–15022.
24. Koch, G.; Zemel, R.; Salakhutdinov, R. Siamese neural networks for one-shot image recognition. In Proceedings of the ICML Deep Learning Workshop, Lille, France, 6–11 July 2015; Volume 2.
25. Liu, H.; Tian, Y.; Yang, Y.; Pang, L.; Huang, T. Deep relative distance learning: Tell the difference between similar vehicles. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2167–2175.
26. Zheng, L.; Zhang, H.; Sun, S.; Chandraker, M.; Yang, Y.; Tian, Q. Person re-identification in the wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1367–1376.
27. Yi, D.; Lei, Z.; Liao, S.; Li, S.Z. Deep metric learning for person re-identification. In Proceedings of the 2014 22nd International Conference on Pattern Recognition, Stockholm, Sweden, 24–28 August 2014; pp. 34–39.
28. Gómez-Silva, M.J.; Armingol, J.M.; de la Escalera, A. Deep Parts Similarity Learning for Person Re-Identification. In Proceedings of the VISIGRAPP (5: VISAPP), Prague, Czech Republic, 25–27 February 2019; pp. 419–428.
29. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the 28th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 91–99.
30. Wu, Y.; Kirillov, A.; Massa, F.; Lo, W.Y.; Girshick, R. Detectron2. Available online: https://github.com/facebookresearch/detectron2 (accessed on 21 November 2023).
31. Qi, C.R.; Liu, W.; Wu, C.; Su, H.; Guibas, L.J. Frustum pointnets for 3D object detection from rgb-d data. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018 ; pp. 918–927.
32. Shrivastava, A.; Gupta, A.; Girshick, R. Training region-based object detectors with online hard example mining. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 761–769.

33. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2961–2969.
34. Liang, M.; Yang, B.; Chen, Y.; Hu, R.; Urtasun, R. Multi-task multi-sensor fusion for 3D object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7345–7353.