remote sensing

MDPI

Article

3D Instance Segmentation and Object Detection Framework Based on the Fusion of Lidar Remote Sensing and Optical Image Sensing

Ling Bai [1,*], Yinguo Li [2], Ming Cen [2] and Fangchao Hu [3]

[1] Department of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

[2] Department of Automation, Chongqing University of Posts and Telecommunications, Chongqing 400065, China; liyg@cqupt.edu.cn (Y.L.); cenming@cqupt.edu.cn (M.C.)

[3] Department of Mechanical Engineering, Chongqing University of Technology, Chongqing 400054, China; fangchaohu@cqut.edu.cn

* Correspondence: d170201001@stu.cqupt.edu.cn

Abstract: Since single sensor and high-density point cloud data processing have certain direct processing limitations in urban traffic scenarios, this paper proposes a 3D instance segmentation and object detection framework for urban transportation scenes based on the fusion of Lidar remote sensing technology and optical image sensing technology. Firstly, multi-source and multi-mode data pre-fusion and alignment of Lidar and camera sensor data are effectively carried out, and then a unique and innovative network of stereo regional proposal selective search-driven DAGNN is constructed. Finally, using the multi-dimensional information interaction, three-dimensional point clouds with multi-features and unique concave-convex geometric characteristics are instance over-segmented and clustered by the hypervoxel storage in the remarkable octree and growing voxels. Finally, the positioning and semantic information of significant 3D object detection in this paper are visualized by multi-dimensional mapping of the boundary box. The experimental results validate the effectiveness of the proposed framework with excellent feedback for small objects, object stacking, and object occlusion. It can be a remediable or alternative plan to a single sensor and provide an essential theoretical and application basis for remote sensing, autonomous driving, environment modeling, autonomous navigation, and path planning under the V2X intelligent network space–ground integration in the future.

Keywords: urban transportation; lidar and camera sensors fusion; 3D object detection; stereo regional proposal network; octree-based hypervoxels

check for updates

Citation: Bai, L.; Li, Y.; Cen, M.; Hu, F. 3D Instance Segmentation and Object Detection Framework Based on the Fusion of Lidar Remote Sensing and Optical Image Sensing. *Remote Sens.* **2021**, *13*, 3288. https://doi.org/10.3390/rs13163288

Academic Editor: Csaba Benedek

Received: 30 June 2021
Accepted: 17 August 2021
Published: 19 August 2021

1. Introduction

With the rapid development of artificial intelligence and automation, an intelligent fusion system with top-down and multi-tier architecture is constructed by multiple vehicular sensors, combining environmental perception, path planning, intelligent behavior decision-making, automatic control, and vehicle architecture. As an essential basis for realizing self-driving and safe driving, the perception of driving environment information around intelligent vehicles has always been of tremendous application challenge and theoretical research value.

Nowadays, the application of multi-sensor data fusion has gradually revealed its advantages in intelligent terminal equipment and augmented reality technology [1]. It is challenging to obtain complex three-dimensional (3D) spatial information using the two-dimensional image information acquired by the visual sensor, and the computational cost is expensive. Even if the same object is regarded as an invariant, the calculation result of its segmentation will be affected by many variables such as illuminant source and noise.

The Lidar sensor uses the reflection time of the emitted radial laser to estimate the depth, projecting points in the 3D space, and obtaining a point cloud in solid space independent of the illuminant source. However, the density of points per unit area of Lidar is not uniformly distributed, and the point cloud representation of the same object is entirely different depending on the number of shots and the reflection distance. In this paper, the fusion advantages of vision sensor and Lidar sensor are used as an effective way to achieve object detection, classification, and tracking perception of intelligent vehicles, which can be used not only as an integrated auxiliary system but also as a practical auxiliary alternative application in the case of a single sensor failure. In addition, since 3D object detection enables a higher degree of information representation of high-dimensional spatial samples that contain more depth information and allow more accurate estimation of the object pose, its feature representation is not affected by scale, rotation, and illumination [2]. Therefore, this paper focuses on 3D object detection and classification based on multi-source sensor fusion, especially the vision sensor and Lidar sensor, to achieve instance segmenting of objects from the scene.

Researchers have proposed various object detection methods based on 3D data processing technology. Their recognition mechanisms are mainly divided into two categories: retrieval methods based on the geometric equation representation mechanism of curved surfaces and unsupervised methods based on feature mechanisms. For instance, for 3D object detection methods of retrieval mechanisms, Guo et al. [3] obtained the most similar object containing the complete object from a reference dataset by a partial 3D shape retrieval method. They calculated the similarity between the query and candidate shapes as their feature vector distance. Kuansheng et al. [4] utilized the Mokhtarian method [5] to estimate the mean curvature of the mesh surface for the 3D models and the curvature of the 3D surface, and divided the 3D models into delicate, flat, and steep parts of curvature region by the New Weighted Fuzzy C-Means scheme [6] embedded with the shape distribution to achieve the corresponding 3D object retrieval. Garro et al. [7] described a 3D object retrieval framework that depends on tree-based shape representations (TreeSha). The framework analyzed the maxima of the Auto Diffusion Function and the associated basins of attraction, and it proposed the custom graph root–leaves path kernels to provide good results of an object retrieval. For other 3D object detection methods based on feature mechanisms, Chen et al. [8] utilized a method named ground segmentation by discriminant image (GSDI) to segment ground point clouds, and a dynamic distance threshold clustering (DDTC) method was designed for the point cloud data of different densities, which improves the detection effect of long-distance objects. Zhou et al. [9] used the embedded attention branch of attention mechanism and the boundary thinning branch combined with the bottom information of RGB and depth image as the two branch architectures of the three-dimensional (S3D) saliency object detection (SOD) network, and added the global view branch that integrates high-level information to retain a large number of cross modal data effectively. Luo et al. [10] designed a network termed 3D-SSD for the modal 3D object detection. The model employed two subnetworks in hierarchical feature fusion parts to learn and fuse the complementary feature information in RGB and depth images. Moreover, the 3D poses of the attached bounding boxes were determined by the depth image in the multi-layer prediction parts. Most of the existing feature-based 3D object detection methods use a single sensor to segment objects in a single dimension. The calculation results are affected by the performance and limitations of the sensor itself. The 3D object detection methods based on the retrieval mechanism strongly depend on the repeatability, descriptive, reliable, and accurate quantitative feature correspondence between the scene to be segmented and the storage model to be detected. The stability of these methods is weak, and it is not easy to analyze the scene with occlusion or truncation of objects. It lacks detection feedback under a certain degree of object stacking and is limited to scenes with specific scales and specific granular knowledge. Good inspiration is given in the literature [11]. Ong et al. [11] proposed an online multi-camera multi-object tracker, which is suitable for optimal Bayesian multi-view multi-object filtering. It is seamlessly integrated into a single Bayesian recursion, tracking management, state estimation, clutter suppression, and occlusion/false detection processing sub-task, and it runs in the 3D object detection framework. The use of multi-camera sensors and multi-view angles has particularly good detection feedback on stacked occlusion objects. However, it is currently limited to a laboratory environment, and the positioning error and execution time are lacking in application effectiveness for large-scale scenarios, predominantly urban traffic scenarios.

To solve the above problems, we propose a framework to enhance the index position of the object in the 3D Lidar point cloud segmentation. Furthermore, we use the framework to restore the classification and posture of the 3D object and isolate the object-like points in the Lidar point cloud. First, we acquire 3D point clouds and stereo image pairs through 3D information scanning acquisition equipment, such as vehicle-borne vision sensors (Lidar and stereo binocular camera). Secondly, the calibration of sensors and coordinate transformation of 3D data are carried out to fuse the data of Lidar and the camera. Remarkably, the data collected by sensors here have already been synchronized and amended. Moreover, the 2D object information is identified using the candidate boundary box based on the stereo regional selective search and the Directed Acyclic Graph Neural Network (DAGNN) detector trained by backbone network architecture. Then, we perform secondary bounding box regression on the cursor anchor points of the two-dimensional object, and perform label alignment with the aforementioned data fusion result to obtain the corresponding local point cloud. Statistical filtering is performed on the local point cloud to remove discrete data points. In addition, an octree-based spatial index structure of voxel is established for the fused point clouds, and the point cloud is over-segmented by the region growth based on clustering. Then, considering the distribution of 3D attributes and geometric features, the quadratic hypervoxel clustering is carried out. Finally, the local point cloud and the quadratic clustering results are registered and fused to realize 3D object detection and instance segmentation, and 3D information and semantic context information are used to orient, place, and score the projected 3D bounding box around the object in the image. The results of object detection and classification with sensor fusion are obtained.

The contributions of this paper are as follows:

(1) The fusion application of Lidar remote sensing technology and optical image sensing technology are fully leveraged to determine the pre-fusion and alignment of the field of view, reduce redundant data processing, and reduce the algorithm complexity of a certain level.

(2) The stereo regional proposal selective search-driven DAGNN expands the receptive field under the trick of the dilated convolution, avoids the scale loss, and the redouble loss function effectively integrates the positioning and semantic information. The detection result for small objects, object occlusion, and object stacking all have significant feedback.

(3) Similarly to superpixels, the point cloud data are calculated at a certain granularity while considering the unique and innovative 2D object information, point cloud color, texture, size, and physical concave-convex geometric features of the 3D point cloud voxelization and hypervoxel clustering boundary. The proposed point cloud instance segmentation is excellent, and the octree-driven voxel storage and cluster growth calculation also make the layout of the segmentation class more accurate and precise, and the calculation becomes faster.

(4) Finally, the visualization of 2D and 3D object boundary box mapping is carried out, which provides certain accurate positioning information and semantic information, and can provide an essential basis for intelligent navigation and path planning. The proposed framework of multi-sensor, multi-dimensional data, multi-mode fusion, and multi-layer interaction remedies a single sensor failure under complex weather, vehicle transportation environment, and lighting conditions. The framework can be used as a lever or alternative application.

With the development and application of high-speed mobile communication technology, the proposed framework in this paper will become very interesting to realize the application of a priori data under the V2X intelligent network connection based on the integration of the space–ground for remote sensing information.

2. Related Works

With the wide application of artificial intelligence, virtual reality, intelligent transportation, autonomous navigation, cultural relics' restoration, video games, and other 3D point cloud data processing, segmentation processing, as a critical technology of point cloud data processing in reverse engineering, has become an important research topic. Object detection based on instance segmentation processing of 3D point clouds has significant theoretical research value and application value in the area of understanding of intelligent transportation and decision-making behavior planning of automatic driving.

Based on the instance segmentation of point cloud data, the 3D sampled point clouds on the surface of real-world scenes are acquired and digitized using 3D data acquisition equipment. Then, the disordered point cloud data are divided into a series of point cloud sets using the corresponding segmentation algorithm, so that the point cloud in each homogeneous set corresponds to the entities in the scene and has similar data properties. Next, this paper briefly reviews the six existing kinds of segmentation algorithms based on the point cloud.

- Model-based point cloud segmentation algorithms.

Model-based segmentation algorithms divide the point cloud data with the same mathematical expression into homogeneous regions using the mathematical model of basic original geometric shapes as a priori knowledge. For example, existing model-based point cloud segmentation approaches [12–16] are based on the development of the classical algorithm of random sampling consistency model fitting. The minimum variance estimation is used to calculate the model parameters of the random sample subset, and the deviation between the sample and the model is compared with the preset threshold, which can be used to detect the mathematical features such as line and circle. Model-based point cloud segmentation approaches are based on mathematical principle and geometric prototypes, and are not disturbed by outliers and noise outliers. However, this kind of algorithm has poor adaptability. The segmentation calculation is massive for large-scale complex scenes and unevenly distributed point cloud data, and the segmentation quality is generally low.

- Attribute-based point cloud segmentation algorithms.

Attribute-based point cloud segmentation algorithms are not constrained by the spatial relationship of the point cloud, and use the feature attributes [17–19] in the feature space to robustly cluster the feature vectors of the point cloud. For example, Holz et al. [20] proposed a high frame rate real-time segmentation algorithm that uses integral images to cluster the points of local surface normal vectors and can be used to sense and detect obstacles in robot navigation scenarios. However, the accuracy and the time efficiency of the point cloud segmentation based on attributes highly depend on the choice of feature space and clustering, and to some extent, it is vulnerable to the density change of the point cloud.

- Boundary-based point cloud segmentation algorithms.

Boundary-based point cloud segmentation algorithms use the boundary information of the 3D data region and the shape characteristics of the object to segment point clouds that depend on the boundary points of sharp intensity change [21]. For example, Luo et al. [22] suggested computing the object boundary area information to generate 3D boundary point clouds, thus limiting the initial value and the object range of the clustering growth segmentation. The principle of a segmentation algorithm based on the boundary information is relatively basic, but the algorithm is vulnerable to the noise and the density of the point cloud, and the robustness of the algorithm is not high.

- Region-based point cloud segmentation algorithms.

A region-based point cloud segmentation algorithm fuses points with the same attributes in a specific neighborhood to form a segmentation region. The discrete point clouds around seed surfaces are grouped and expanded into larger surface patches by a similarity measure. Compared with the boundary-based point cloud segmentation algorithm, it has a stronger anti-noise ability and is not easily affected by the density of point clouds and their outliers. It is more suitable for dealing with large-scale complex scenarios. For example, Vo et al. [23] proposed an octree-based region growing for point cloud segmentation, which achieved fast and accurate segmentation of 3D point clouds in the urban environment through two stages from coarse to fine. Region-based segmentation algorithms [24–26] are efficient and straightforward, but are affected by the growth strategy. We need to pay attention to the problem of over-segmentation and under-segmentation.

- Graph-based point cloud segmentation algorithms.

Graph-based point cloud segmentation algorithms convert 3D point cloud data into graph structure data, i.e., point–edge set, which are not affected by mathematical geometry and spatial distribution of a point cloud [27]. Edge is the similarity weight of a pair of points in point cloud data. The similarity in the segmentation process meets the minimum between different segmentation regions, while the same segmentation region is the largest. Among them, probabilistic reasoning models are often used to solve graph segmentation problems, such as that of Tatavarti et al. [28–30], who utilized a plane model, Markov random field and efficient Bayesian belief propagation to segment geometry-only depth images. The graph-based method can be used to process point cloud data in complex scenes, but the complexity of constructing graphs or energy functions cannot be estimated.

- Learning-based point cloud segmentation algorithms.

In recent years, machine learning-based point cloud segmentation algorithms have gained attention and development. Charles et al. [31] first put forward a deep neural network for directly dealing with the original 3D point cloud. MaxPooling was used as a symmetric function to deal with the disorder of the point cloud model, and input transform and feature transform were used to maintain the spatial invariance of point cloud data. However, it lacked the ability to extract local information, and it was inappropriate to extract the nearest neighbors under the uneven density of point cloud. The point cloud learning networks [32–34] under normalized input strongly depend on data sources. The point cloud is affected by the acquisition equipment and the coordinate system, and its arrangement is changeable. For robotic and automatic driving scenarios, the coverage of sampling points is relatively sparse compared with the scene scale, and the amount of information of point cloud is very limited. Therefore, there is great potential for the development space of point cloud segmentation algorithms based on learning.

The above-mentioned segmentation algorithms have their own characteristics. However, there are some deficiencies in robustness and time-consumption by using only one segmentation strategy. The hybrid segmentation algorithm proposed in this paper takes full advantage of the spatial index structure of point clouds and the growth strategy under geometric attributes, and the object points detected by the learning classifier can be used as regional constraints. Finally, 3D object detection and semantic information perception can be realized in large-scale and complex scenes of automatic driving.

3. The Proposed Framework Overview

The framework consists of three parallel pipelines, in which the modules perform serial interaction coupling and decoupling operations, i.e., the entire framework implements a fusion framework of multi-layer architecture and multi-dimensional information interaction. The interfaces and calling modes between the modules are explicit, depending on the dimension of current data information, and do not limit the scale size and determined knowledge granularity. From the perspective of different attributes, the proposed framework generates a hierarchical knowledge space chain so that the intelligent vehicle sensor data of synchronization and rectification can be perceived by sensor fusion, and it completes the object detection and classification of the 3D object with the manifestation attribute and semantic functional meaning of instance segmentation under the sensor fusion perception. The proposed framework of this paper has the ability to deal with more extensive processes, and the scalability is very impressive.

3.1. Data Fusion

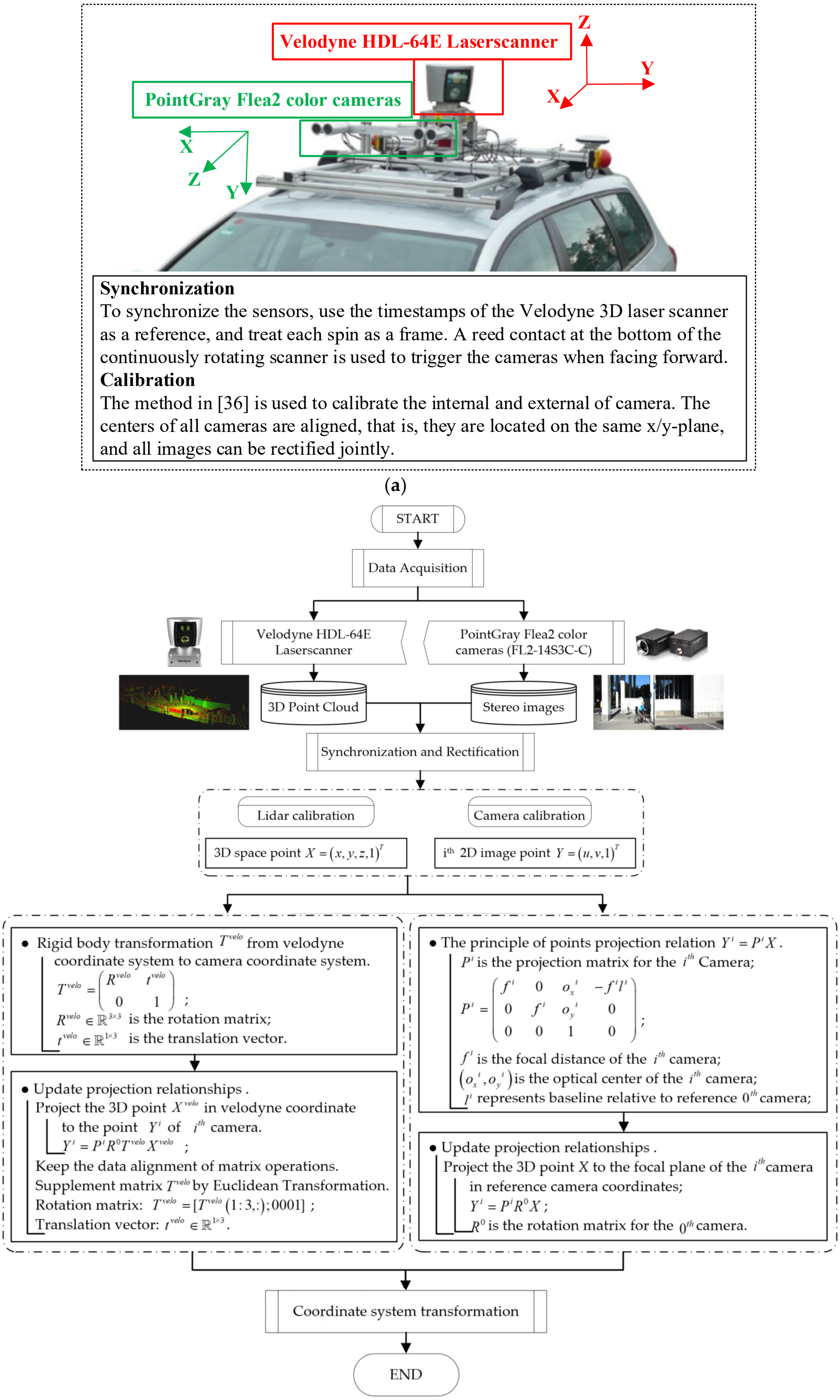
The simultaneous interpreting of data can integrate the advantages of different sensors and is currently a popular method of environmental awareness. The Lidar sensor can improve the detection rate of the small object. The advantage of Lidar is that the accurate 3D position information can be collected, but the disadvantage is that the detail resolution is low. The image data of the camera sensor nevertheless contain plentiful details and semantic information; the accuracy of depth estimation is universally low. Integrating the advantages of Lidar depth estimation into camera sensor data is a fusion perception advantage in the field of artificial intelligence, augmented reality, and other applications, which improves the positioning accuracy of small objects without losing data details [35]. This paper uses the following task flow in Figure 1 to perform preliminary multi-modal Lidar and camera data fusion alignment. The laser sensor is used to obtain a 3D point cloud, and the camera sensor is used to obtain 2D stereo images. In order to avoid data drift and error accumulation over time, the data of sensor acquisitions are synchronization and rectification data. The plane configuration diagram of the coordinate system is shown in Figure 2.

Data fusion of Lidar and camera sensor: *The 3D points of the Lidar point cloud and the 2D pixels points of the image in the field of view are fused and aligned, and points outside the boundary of the perceptive view angle, points that are too close to the vehicle itself, and background points that are too far away, i.e., close to the vanishing point, are all removed. Granularity-sized subsampling is performed.*

- The projection matrix P^{cam}_{velo} of Lidar point cloud data to the image plane is calculated with Equation (1):

$$P^{cam}_{velo} = P^i R^0 T^{velo} \tag{1}$$

  where the projection transformation from a 3D point cloud to an image plane can be simply understood as projecting the 3D information in the physical world onto the 2D information surface under a certain perspective. The projection transformation is at the expense of depth information, as shown in Figure 3.
- The point cloud is sampled at the granularity size of the unit radius $G_{radius} = n$, and the near points of the obstacles in the image plane are removed, i.e., the laser point cloud with negative x^{vel} behind the camera plane is deleted, so that the detection range is forward, and the near point $x^{vel} < depth_{\min}$ is removed.
- As the task flow of preliminary sensor fusion and alignment, the projective transformation of the 3D point cloud and the image plane is performed by the projection matrix P^{cam}_{velo}. The image points in the homogeneous coordinate system are calculated and normalized.
- The results of data fusion between Lidar point clouds and images are drawn in the 2D space plane, and the color values are assigned to the depth of the dot matrix to represent the colormap of the depth of field.



Synchronization
To synchronize the sensors, use the timestamps of the Velodyne 3D laser scanner as a reference, and treat each spin as a frame. A reed contact at the bottom of the continuously rotating scanner is used to trigger the cameras when facing forward.
Calibration
The method in [36] is used to calibrate the internal and external of camera. The centers of all cameras are aligned, that is, they are located on the same x/y-plane, and all images can be rectified jointly.

(**a**)

START
Data Acquisition
Velodyne HDL-64E Laserscanner
PointGray Flea2 color cameras (FL2-14S3C-C)
3D Point Cloud
Stereo images
Synchronization and Rectification
Lidar calibration
Camera calibration
3D space point $X=(x,y,z,1)^T$
ith 2D image point $Y=(u,v,1)^T$
Coordinate system transformation
END

- Rigid body transformation T^{velo} from velodyne coordinate system to camera coordinate system.
  $T^{velo}=\begin{pmatrix} R^{velo} & t^{velo} \\ 0 & 1 \end{pmatrix}$;
  $R^{velo}\in\mathbb{R}^{3\times 3}$ is the rotation matrix;
  $t^{velo}\in\mathbb{R}^{1\times 3}$ is the translation vector.

- Update projection relationships .
  Project the 3D point X^{velo} in velodyne coordinate to the point Y^i of i^{th} camera.
  $Y^i=P^iR^0T^{velo}X^{velo}$;
  Keep the data alignment of matrix operations.
  Supplement matrix T^{velo} by Euclidean Transformation.
  Rotation matrix: $T^{velo}=[T^{velo}(1:3,:);0001]$;
  Translation vector: $t^{velo}\in\mathbb{R}^{1\times 3}$.

- The principle of points projection relation $Y^i=P^iX$.
  P^i is the projection matrix for the i^{th} Camera;
  $P^i=\begin{pmatrix} f^i & 0 & o_x{}^i & -f^il^i \\ 0 & f^i & o_y{}^i & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$;
  f^i is the focal distance of the i^{th} camera;
  $(o_x{}^i, o_y{}^i)$ is the optical center of the i^{th} camera;
  l^i represents baseline relative to reference 0^{th} camera;

- Update projection relationships .
  Project the 3D point X to the focal plane of the i^{th} camera in reference camera coordinates;
  $Y^i=P^iR^0X$;
  R^0 is the rotation matrix for the 0^{th} camera.

(**b**)

Figure 1. Task flow and two acquisition devices. (**a**) [36] Two acquisition devices; (**b**) Task flow of preliminary sensor fusion and alignment.

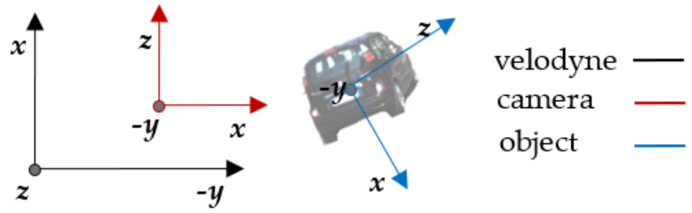


Figure 2. Plane configuration diagram of coordinate system. For camera, x = right, y = down, z = forward; For velodyne, x = forward, y = left, z = up, and coordinates are real numbers; For object, coordinates are integer values.

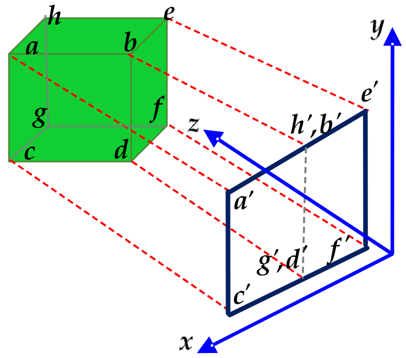


Figure 3. Schematic diagram of projection transformation.

The advantage of the multi-mode learning data fusion of Lidar and camera [37] is that it takes into account the spatial distribution information of 3D point clouds and the color information of images, which reduces the difficulty of solving subsequent problems, such as interest region segmentation and object recognition. The data dimension of the problem to be processed is improved, which is conducive to the fine classification of the learning mechanism. The corresponding extended segmentation of the effective region can also reduce the spatial complexity of an image search and provide a basis for object segmentation, such as environment modeling based on semantic understanding.

3.2. Stereo Regional Proposal Selective Search-Driven DAGNN

In this paper, the candidate bounding boxes generated by the stereo regional selective search and the whole image are utilized as inputs, and the Directed Acyclic Graph Neural Network (DAGNN) is trained by using the backbone network architecture of the VGG16 of the ImageNet network model, which enables the extraction of image features. Generally, image segmentation and region growing technologies are mainly used in the screening strategy of boundary candidate boxes for object detection. It not only needs to output the class probability of the object, but also needs to select the box to locate the specific location of the object. In order to locate the specific object location, the image can be divided into many patches as input and transmitted to the object recognition model. The region-selective search of traditional monocular vision has a limited field of view. Only the stereo vision that simulates the binocular mechanism can accurately locate the object closer to the real world. For example, as shown in Figure 4, the object class in the right view of Figure 4b is mostly occluded. The object area cannot be detected by the monocular region selective search. The left view of Figure 4a presents a relatively complete object with a small occluded area. In the right view of Figure 4d, the stacking of double objects is serious, which cannot effectively distinguish whether it is a single object or a double object area. In the left view of Figure 4c, the stacking degree of double objects is weakened, which can distinguish double objects to a certain extent. Therefore, this paper aims to improve the object stacking and occlusion detection feedback through the cooperative, interactive regional selective search with binocular stereo vision.

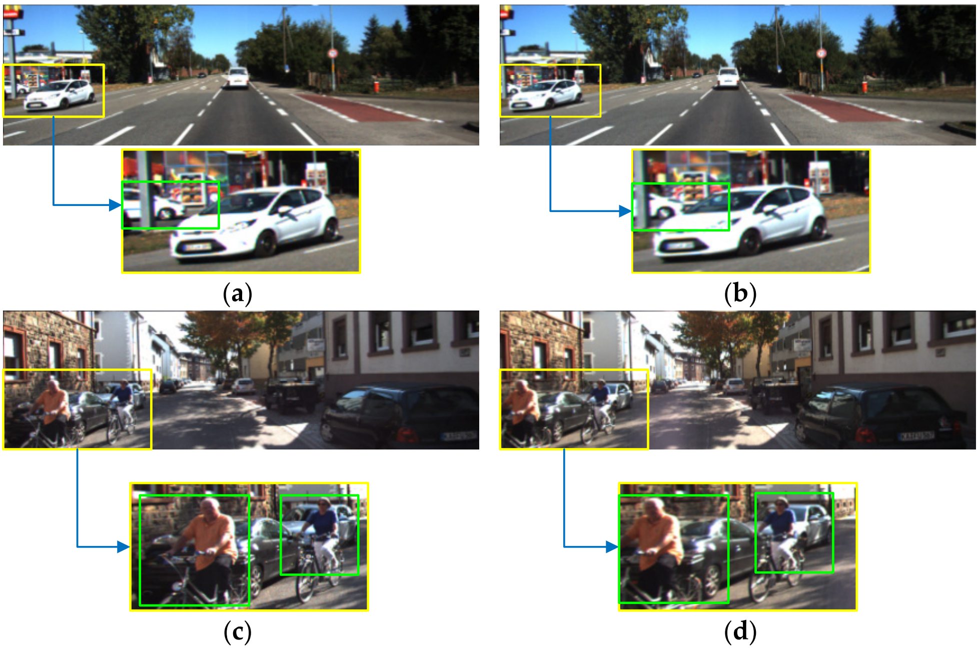


Figure 4. Object occlusion and stacking scenes. (**a**) Left image of frame *t* in sequence A; (**b**) Right image of frame *t* in sequence A; (**c**) Left image of frame *t* in sequence B; (**d**) Right image of frame *t* in sequence B.

As shown in Figure 5, the regional selective search method of the sliding window firstly performs sliding window movement of different window sizes on the input image, i.e., from left to right, from top to bottom. Convolution operations are performed simultaneously on the current window of each sliding process, and the existing probability of the object with the trained classifier is determined. If the class probability is high, the detected object is considered to exist. The corresponding object markers can be obtained by detecting sliding windows with different window sizes. However, there are overlapping parts in the window. At this time, the non-maximum suppression needs to be filtered, and finally, the detected object is obtained.

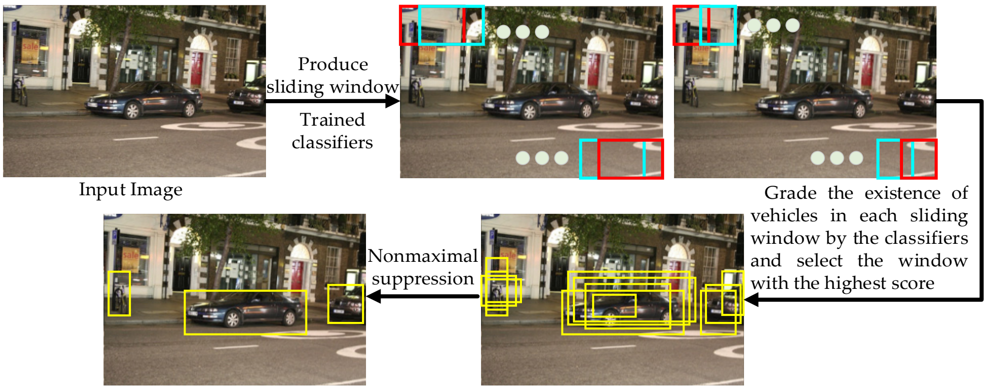

Produce sliding window
Trained classifiers
Input Image
Grade the existence of vehicles in each sliding window by the classifiers and select the window with the highest score
Nonmaximal suppression

Figure 5. The region selective search method of the sliding window.

Although the sliding window approach is uncomplicated and easy to implement, there are redundancy data because it enumerates all sub-image blocks on the whole image according to the size of the patches. Considering the length: width ratio of objects, the global search with different window sizes often leads to inefficiency. For the high real-time classifier for object detection of automatic driving intelligent vehicles, most sub-patches do not register objects in the process of an exhaustive search of image sub-patches. Therefore, this paper only searches for the most likely regions to contain objects to improve computational efficiency. In this paper, the candidate bounding boxes of the object are extracted by region iterative merging based on the similarity between the sub-patches. In each iteration, the merged sub-patches are circumscribed as rectangles, i.e., the candidate bounding boxes. Among them, the similarity standard of the sub-patch p_i and the sub-patch p_j is shown in Equation (2), which combines the four parameters of color, size, texture, and shape compatibility:

$$\begin{aligned} Similarity\left(p_i, p_j\right) &= \lambda_1 Color\left(p_i, p_j\right) + \lambda_2 Texture\left(p_i, p_j\right) \\ &\quad + \lambda_3 Size\left(p_i, p_j\right) + \lambda_4 Shape\left(p_i, p_j\right) \end{aligned} \tag{2}$$

where $\lambda_i (i = 1, 2, 3, 4)$ is the value of the damping coefficient measured by the parameter 0 or 1.

Color Parameter:

$$Color\left(p_i, p_j\right) = \sum_{k=1}^{n} \min\left(h_i^k, h_j^k\right) \tag{3}$$

where h_i^k and h_j^k are the k^{th} bin histogram values of two patches in the color descriptor.

Size Parameter:

$$Size\left(p_i, p_j\right) = 1 - size(p_i) + size\left(p_j\right) / size(image) \tag{4}$$

where $size(\cdot)$ is the image size in pixels.

Texture Parameter:

$$Texture\left(p_i, p_j\right) = \sum_{k=1}^{n} \min\left(h'^k_i, h'^k_j\right) \tag{5}$$

where h'^k_i and h'^k_j are the k^{th} bin histogram values of two patches in texture descriptor.

Shape compatibility Parameter:

$$Shape\left(p_i, p_j\right) = 1 - \frac{size\left(BB_{ij}\right) - size(p_i) - size\left(p_j\right)}{size(image)} \tag{6}$$

where $size\left(BB_{ij}\right)$ is the bounding box of the merged patch of p_i and p_j.

The computational efficiency of the region selective search is better than that of the sliding window method. The sub-patch merging strategy can obtain suspected object bounding boxes of different sizes, and the similarity index of sub-patch merging is diverse, which improves the probability of object detection. The corresponding regions of the left and right images are taken to conform to the similarity of the boundary boxes. The stereo object region searches and detection are carried out based on the sensor fusion results of the updated boundary box corresponding to the perspective of maximum similarity. Among them, it is worth noting that in the region similarity measurement of independent perspective, once the candidate region bounding box is generated, the object region of the eight-neighborhood bounding box of another perspective is calculated correspondingly. Then, the stacking and occlusion of the object area are merged and indexed. Finally, the regional proposal for merging or splitting object regions of stacking and occlusion is realized. The hierarchical grouped region selective search is given as Algorithm 1.

Algorithm 1. Region Selective Search

```
Input: Color left image $I_1$,right image $I_2$
Output: Object location boxes hypotheses B{}
begin
  Initialize regions R={$r_1$,…,$r_n$} by graph-based segmentation;
    Calculate the dissimilarity between each pixel point and its {1,2,3,4,5,6,7} neighborhoods in Eight Neighborhood;
    Sort edges by dissimilarity in order of non-decreasing;
    while E{}≠∅ do
      Get edge $e_1$ in the sorted edges set E{$e_1$,$e_2$,…,$e_n$};
      for the current edge $e_2$ do
        judgement of merger;
        let the vertices connected by $e_2$ be ($v_i$,$v_j$);
        if the dissimilarity less than the internal dissimilarity then
          update the threshold and object class label;
          else if $i < n$ then
            select the next edge in the sorted;
          else
            break;
  Initialize similarity set S{} = ∅;
    for each neighboring region pair{$r_i$,$r_j$} do
      calculate each similarity $s(r_i,r_j)$ of region $i$ and $j$;
      $S = S \cup s(r_i,r_j)$;
      while S{} ≠ ∅ do
        obtain the highest similarity $s(r_i,r_j)=max(S)$;
        merge the corresponding regions $r_{merge}=r_i \cup r_j$;
        remove the similarities regarding $r_i$: $S=S \setminus s(r_i,r_{temp})$
                                           $r_j$: $S=S \setminus s(r_{temp},r_j)$
        Calculate similarity set $S_{merge}$ between $r_{merge}$ and its neighbours;
        $S = S \cup S_{merge}$;
        $R = R \cup r_{merge}$;
      Extract object location boxes B{} from all regions in R{}
  return B{};
```

The proposed stereo regional proposal selective search-driven DAGNN adopts an end-to-end network architecture with the training sets of Pascal VOC and ImageNet. The network structure is shown in Figure 6, and the network structure configuration is shown in Table 1.

The activation function is set after each convolution layer and full connection layer. The activation function is introduced to increase the nonlinearity of the neural network model. In addition, the stereo regional proposal is mapped to the feature map of the last convolution layer. When the size dimension of the feature map reaches 512, the multi-configured atrous convolution, i.e., dilated convolution, is established after the last convolutional layer of each layer. The dilated rates are (2, 2, 1, 2, 4), respectively. Compared with the previous pooling, the context information lost is retained. Without changing the size of the feature map, the receptive field is enlarged, more dense data are obtained, and the calculation is fast. The segmentation and detection effects of small objects are sound. After that, an ROI pooling layer is added, so that a fixed-size feature map can be generated for each proposal region. Then, singular value decomposition (SVD) is used to decompose the fully connected layer to simplify the calculation. Finally, the softmax loss of classification probability and the smooth loss of boundary box regression of object detection are integrated into the multi-task loss function for joint training.

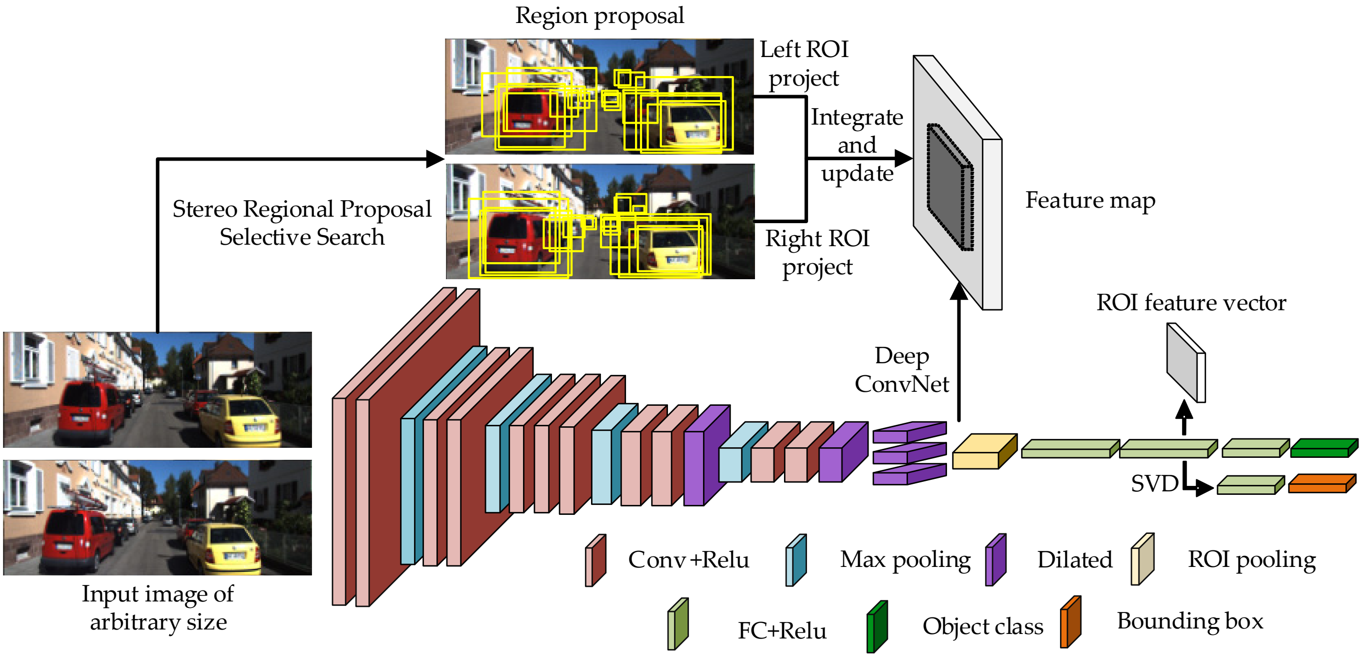


Figure 6. The architecture of stereo regional proposal selective search-driven DAGNN.

Table 1. Configuration of network.

Layer	Size Dimension	Kernel Size	Stride	Padding
conv1_1	224 × 224 × 64	3 × 3	1	1
conv1_2	224 × 224 × 64	3 × 3	1	1
pooling1	112 × 112 × 64	2 × 2	2	0
conv2_1	112 × 112 × 128	3 × 3	1	1
conv2_2	112 × 112 × 128	3 × 3	1	1
pooling2	56 × 56 × 128	2 × 2	2	0
conv3_1	56 × 56 × 256	3 × 3	1	1
conv3_2	56 × 56 × 256	3 × 3	1	1
conv3_3	56 × 56 × 256	3 × 3	1	1
pooling3	28 × 28 × 256	2 × 2	2	0
conv4_1	28 × 28 × 512	3 × 3	1	1
conv4_2	28 × 28 × 512	3 × 3	1	1
dilated4_3_2	28 × 28 × 512	5 × 5	1	0
pooling4	14 × 14 × 512	2 × 2	2	0
conv5_1	14 × 14 × 512	3 × 3	1	1
conv5_2	14 × 14 × 512	3 × 3	1	1
dilated5_3_2	14 × 14 × 512	5 × 5	1	0
dilated5_4_1	14 × 14 × 512	5 × 5	1	0
dilated5_4_2	14 × 14 × 512	9 × 9	1	0
dilated5_4_4	14 × 14 × 512	17 × 17	1	0
Pooling6	7 × 7 × 512	2 × 2	2	0
fc6_1	1 × 1 × 4096	-	-	-
fc6_2	1 × 1 × 4096	-	-	-

Atrous convolution, i.e., dilated convolution, which introduces a new parameter called dilated rate into the convolution layer, defines the interval of each value when the convolution kernel processes the data. The normal convolution and dilated convolution are shown in Figure 7 (take 3 × 3 convolution as an example).

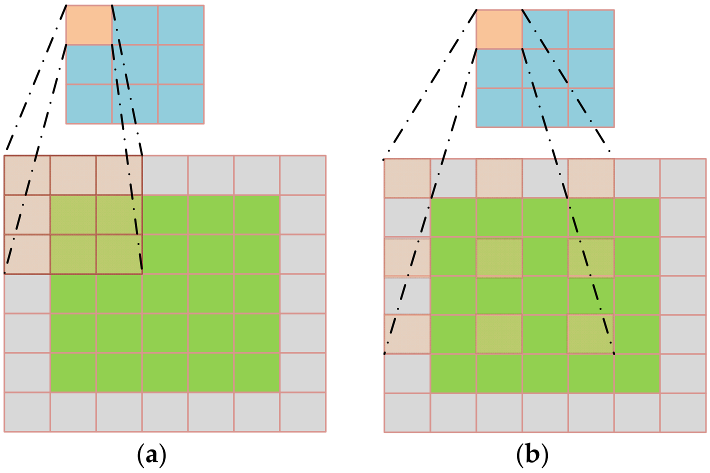


Figure 7. Different types of convolution. (**a**) Normal convolution; (**b**) dilated convolution.

Compared with the traditional convolution operation, the three 3 × 3 convolution layers can only reach a receptive field of $(kernel - 1) \times layer + 1 = 7$ if the stride is 1, which is linearly related to the number of layers, while the receptive field of dilated convolution grows exponentially. The formula for calculating the equivalent convolution kernel size for dilated convolution is as follows:

$$Se = s + (s - 1) \times (d_{rate} - 1) \tag{7}$$

where Se is the equivalent convolution kernel size, s is the actual convolution kernel size, and d_{rate} is the dilation parameter of convolution.

In addition, as shown in Figure 6 of the neural network, this paper has two peer output layers. One outputs the discrete probability distribution $p = (p_0, \ldots, p_k)$ of $k + 1$ class, which represents the probability of belonging to k class. The softmax function is used to evaluate the cost loss of object classification:

$$Loss_{class} = -\log p_k \tag{8}$$

Another one outputs the boundary box regression offset, i.e., a $4 \times k$ dimensional array, which represents the boundary box adjusted parameters of translation and scaling belonging to the k-index class, respectively. The *Smooth L1* function is utilized to evaluate the cost loss of the location of the bounding box:

$$Loss_{locate} = \sum_{i \in \{x,y,w,h\}} smooth_{L_1}\left(O_i^k - v_i\right) \tag{9}$$

where the *Smooth L1* error is not sensitive to discrete points and outliers:

$$smooth_{L_1} = \begin{cases} 0.5x^2 & , \quad if|x| < 1 \\ |x| - 0.5 & , \quad otherwise \end{cases} \tag{10}$$

The total multi-task cost loss function is the integrated weighted sum of object detection classification $Loss_{class}$ and bounding box regression $Loss_{locate}$:

$$Loss\left(p, k, O^K, gt\right) = Loss_{class}(p, k) + \sigma[k \geq 1] Loss_{locate}\left(O^K, v\right) \tag{11}$$

where k is the class of the object, O^K is the object detection result of class k, gt is the ground truth, and σ is the harmonic weight between two loss functions. If the classification is background, there is no need to consider the bounding box regression cost.

The detection quality of DAGNN obtained by the stereo regional proposal selective search is higher than that obtained by traditional methods. Moreover, the rough localization of candidate boxes has contributed to the subsequent precise localization. The multi-task loss function, which trains the classification probability and the bounding box regression jointly, can share the convolution feature with strong robustness.

3.3. Octree-Guided Hypervoxels Over-Segmentation

A three-dimensional point cloud is a collection of sample points on the surface of an object, which has the characteristics of sparseness, messiness, unevenness, and possession of massive streams of data information. The distribution between points is discrete and sparse, and there is no topological relationship of a set similar to the 2D traditional data model. Therefore, the point cloud storage mode and access index are particularly important for the subsequent processing of the point cloud. By establishing the point cloud index structure in 3D space, the storage and search of point cloud data can be accelerated. In this paper, based on the point cloud data fusion of camera sensor and Lidar sensor and the object region index of stereo regional proposal selective search-driven DAGNN, the octree-guided hypervoxels' over-segmentation of the 3D point cloud is carried out, and the different classification storage and adjacency relationship of the object point cloud are judged. Using the 3D physical attributes and geometric features between voxel blocks for over-segmentation can save the storage space of point cloud data, improve the operation efficiency of the algorithm, and realize the detection and classification of the 3D object point cloud. The most significant contribution of this paper is that the framework is more conducive to the fast segmentation of massive point cloud data.

The spatial index of point cloud data is mainly a tree-like index structure of top-down stepwise division and space reduction, such as KD-tree, R-tree, BSP-tree, KDB-tree, quadtree, octree, etc. In point cloud data organization, the common structures are KD-tree and octree. KD-tree uses a hyperplane to divide a space into several disjoint subspaces. Each layer divides the contained space into two subspaces, while the top-level nodes are divided into unidimensional, and the next-level nodes are divided into another dimension. The attributes of all dimensions of the KD-tree circulate among layers. In fact, the binary search tree is extended to a multi-dimensional data structure to realize the organization and storage of multi-dimensional spatial data. However, the amount of point cloud data is complex. The index pointer data built by the KD-tree occupy a large amount of memory space. The depth of the tree is very large, and the search efficiency of the KD-tree is low due to the data search and backtracking. The octree structure divides the aggregated entities in the three-dimensional space into voxels, making each voxel have the same complexity of time and space. The geometric objects in the three-dimensional space are divided using the recursive cyclic partitioning method to form a directional pattern with root nodes. Octree, which has uniform tree structure rules and lower depth than the KD-tree, facilitates geometric operations such as union, intersection, and difference of objects. It has higher performance for finding accurate data points and has certain advantages in spatial decomposition. Therefore, this paper uses the octree structure to establish the spatial index structure of the point cloud. The octree structure is shown in Figure 8.

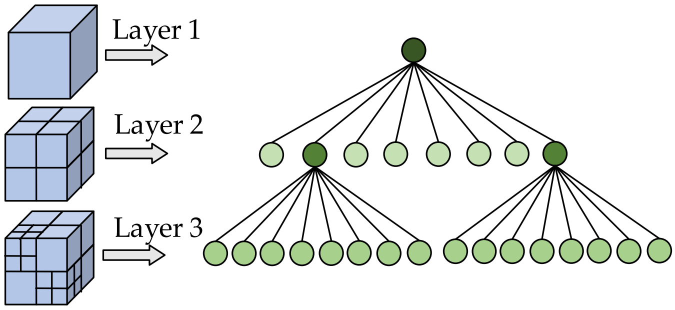


Figure 8. The octree structure.

Voxels are abbreviated as volume elements. They are the generalization of two-dimensional image pixels in the three-dimensional space. They are a set of uniformly distributed cubic geometries located in the center of orthogonal grids and the smallest unit on regular grids. Voxelization is a process in which voxels are used to approximate the spatial structure and geometric shape of a scene or object. In this paper, the basic principle of voxelization is as follows:

Principle of voxelization:

- Firstly, a set of three-dimensional cube grids is established on the input data of point clouds.
- Then, each three-dimensional cube has meshed, and all points in the grid are approximated by the center points of all point clouds' data.
- Finally, the voxel cloud data are generated.

Voxel cloud can represent the surface geometric features and internal attribute information of the model, and the relative position relationship of voxel data can be used to represent the corresponding scene and the three-dimensional information of the object.

Hypervoxels (Supervoxels) are similar to the concept of superpixels in 2D images, and their set elements are voxel data. Similar to the voxel filters, hypervoxels are generated clusters of irregular shapes according to the position relations and other similarity attributes. The essence of hypervoxels is a geometric subset of 3D meshes of atomic voxels with certain sense-perception information in the 3D space. The hypervoxels generated in this paper have regular geometry shape, uniform voxel density, good dependence on boundary information, and rich attribute information. Additionally, they are a summary of local information, which is conducive to the subsequent classification and recognition work, and is easier to manage than other data types.

In this paper, the octree voxelization clustering is used. Firstly, the spatial index structure based on the octree voxelization is established for the fused point cloud data, then the scene point cloud clustering is divided into similar voxels, and the geometric attributes between the sub-blocks are over-segmented by hypervoxels. Nevertheless, different from two-dimensional images, point clouds do not have a pixel adjacency relationship. Therefore, this paper firstly divides point clouds into octree spaces and obtains adjacency relationships among point clusters, such as face adjacency (6 adjacency), line adjacency (18 adjacency), and point adjacency (26 adjacency) (as shown in Figure 9). Then, point adjacency (26 adjacency) is used as an adjacency criterion to absorb similar volume elements in the octave space continuously.

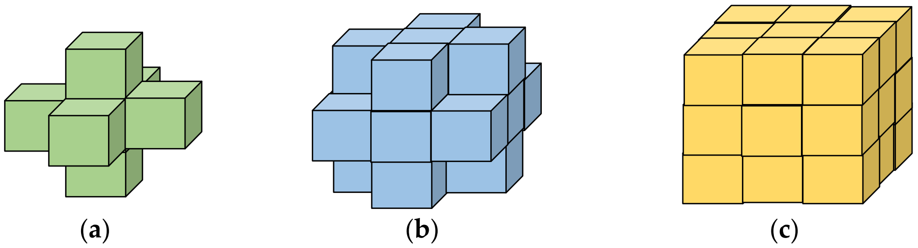


Figure 9. The adjacency criterion. (**a**) 6 adjacency; (**b**) 18 adjacency; (**c**) 26 adjacency.

The similarity distance $Similarity_{dis}$ of adjacent voxels is calculated by combining spatial location, color attributes, and local geometric features, as shown in Equation (12):

$$\begin{cases} Spatial_{dis} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \\ Color_{dis} = \|C_i - C_j\| \\ Geometric_{dis} = \sum\limits_{\mu=0}^{\mu} \frac{1}{2^{\mu}} N_{\mu} \end{cases} \tag{12}$$

where $Spatial_{dis}$ is the spatial distance of i^{th} voxel point (x_i, y_i, z_i) and j^{th} voxel point (x_j, y_j, z_j), $Color_{dis}$ is the Euclidean distance of i^{th} voxel color C_i and j^{th} voxel color C_j, $Geometric_{dis}$ is the geometric distance, and N_μ is the match number for the distribution of adjacent data, as shown in Equations (13) and (14):

$$N_\mu = IS(H_\mu(i), H_\mu(j)) - IS(H_{\mu-1}(i), H_{\mu-1}(j)) \tag{13}$$

$$IS(H(i), H(j)) = \sum_{r=1}^{r} \min\left(H(i)^r, H(j)^r\right) \tag{14}$$

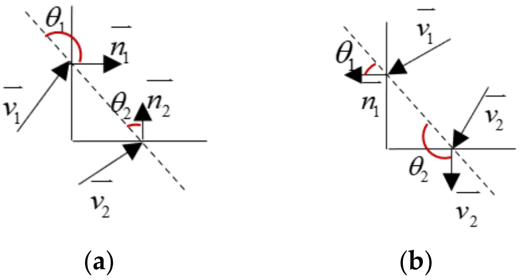
where $IS(\cdot)$ is the histogram intersection, and r is the number of histogram bins.

The integrated similarity distance is the weighted sum of these distances in Equation (15):

$$Similarity_{dis} = \sqrt{\frac{\alpha Spatial_{dis}^2}{3Voxel_{dis}^2} + \frac{\beta Color_{dis}^2}{k^2} + \gamma Geometric_{dis}^2} + \sqrt{\frac{d_{conc}^2}{-\pi/k} + \frac{d_{conv}^2}{\pi/k}} \tag{15}$$

where the three distances are normalized by the maximally distant point with a distance of $\sqrt{3}Voxel_{dis}(Voxel_{dis}$ is the distance between voxels), a constant k in the CIELab space, and a histogram intersection kernel in the FPFH space, respectively; the influence weighted factors α, β, γ satisfy $\alpha + \beta + \gamma = 1$. d_{conv}, d_{conc} represent the concavity and the convexity of the current seed and the merged voxel block, respectively; and they are normalized in different zones of the circle angle.

The judgment termination condition of the voxel clustering uses the geometric properties between the 3D hypervoxel blocks to merge, i.e., the concavity and convexity, as shown in Figure 10. The connection relationship between adjacent hypervoxels is obtained by judging the relationship between the normal vector of voxels and the connected vector of the centroids. According to the similarity measure, the seed hypervoxel and the adjacent hypervoxels of convex features are clustered until the growth boundary is concave, then the clustering stops and hypervoxels' over-segmentation is completed.



(**a**) (**b**)

Figure 10. The geometric properties between 3D hypervoxel blocks. (**a**) Concavity; (**b**) convexity.

The two arbitrary vectors $\vec{v_1}$, $\vec{v_2}$ and the angle θ (where $\theta = \angle\left(\vec{v_1}, \vec{v_2}\right)$) satisfy the following identical Equation (16):

$$\vec{v_1} \cdot \vec{v_2} = \left|\vec{v_1}\right| \cdot \left|\vec{v_2}\right| \cdot \cos(\theta) \tag{16}$$

Then, the angle between the connected vector $\vec{c}\,(\vec{c} = \vec{v_1} - \vec{v_2})$ of two centroids and the normal vectors $\vec{n_1}$,$\vec{n_2}$ of two voxels can be calculated in Equations (17) and (18):

$$\begin{cases} \cos\theta_1 = \vec{n_1} \cdot \hat{c} \\ \cos\theta_2 = \vec{n_2} \cdot \hat{c} \end{cases} \tag{17}$$

$$\hat{c} = \vec{v_1} - \vec{v_2}/\|\vec{v_1} - \vec{v_2}\| \tag{18}$$

Concave-Convex Criterion:

- When $\vec{n_1} \cdot \hat{c} - \vec{n_2} \cdot \hat{c} < 0$, i.e., $\theta_1 > \theta_2$, the connection between the two super-voxels is concave.
- Otherwise, when $\vec{n_1} \cdot \hat{c} - \vec{n_2} \cdot \hat{c} > 0$, i.e., $\theta_1 < \theta_2$, the relative connection between the two super-voxels is convex.
- The concave-convex degree is the sum of $d_{conc} = -\sqrt{\left(\vec{n_1} \cdot \hat{c} - \vec{n_2} \cdot \hat{c}\right)^2} / (\vec{n_1} \cdot \hat{c} + \vec{n_2} \cdot \hat{c})$ and $d_{conv} = \sqrt{\left(\vec{n_2} \cdot \hat{c} - \vec{n_1} \cdot \hat{c}\right)^2} / (\vec{n_1} \cdot \hat{c} + \vec{n_2} \cdot \hat{c})$.

We determine the traversal sequence between voxels according to the adjacency relation graph, judge the neighboring voxels according to the similarity distance, and add them to the clusters, repeat the loop iteration to complete the region growth of the voxel clusters, and take the concave-convex feature of hypervoxels as the growth boundary to realize the over-segmentation of hypervoxels. The horizontal distribution map of voxels and the breadth-first traversal order of the voxel tree structure are shown in Figure 11.

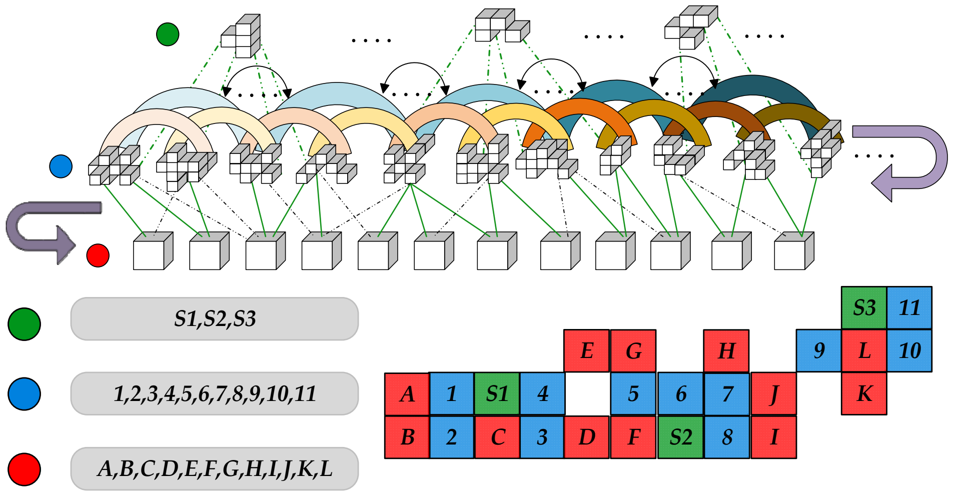


Figure 11. Horizontal distribution and search traversal sequence diagram and layout.

Among them, it is particularly important to note that the clustering process does not perform the next growth after the growth of one hypervoxel is completed. Instead, all hypervoxels undergo growth and clustering at the same time. Then, one layer of hypervoxel competes fairly and continues to develop the next layer. In this cycle, the clustering growth of all hypervoxels is finally completed, while the corresponding voxel structures of the point cloud data are segmented and the particle properties of each individual element of the voxel grid are similar.

3.4. 3D Object Instance Segmentation and Bounding Box Mapping

The over-segmentation results of Section 3.3 are fused with the local point cloud of the proposal object region fused by the camera and Lidar sensor for index calibration and fusion to achieve 3D object detection, classification, tracking, and instance segmentation with the object semantic context information. In addition, the 3D object information and the semantic context information are used to locate, place and grade the projected 3D boundary box into the 2D and 3D space, i.e., the outermost rectangular frame (the size of length, width, and height for the object) of the segmentation result for the object point cloud. The process for the point cloud calibrated fusion of instance segmentation and the 2D/3D boundary boxes mapping is shown Algorithm 2.

Algorithm 2. Process for point cloud calibration and boundary box mapping

```
Input: 3D object region index region; The over-segmentation result of
       voxelization region^; The projection matrix P in data fusion of
       coordinate system transformation in sensor fusion;
       The boundary box with sequential index.
Output: The object detection result set result{}
begin
  Initialize $P=1/|P|\times P^*$;
    for top plane
        the vertices are defined as $a, b, c, d$ in proper order;
      else for bottom plane
        the vertices are defined as $a^*, b^*, c^*, d^*$ in proper order;
    while result{} $\neq \varnothing$ do
      if $region \cap region^\wedge / region \cup region^\wedge \geqslant 0.5$ then
        $result = region^\wedge \times P^{-1}$;
        else if $region \cap region^\wedge / region \cup region^\wedge < 0.5$ then
          $result = region \times P^{-1}$;
        else
          break;
      calculate the rotational matrix around yaw axis;
      determine the corners of 3D boundary box in $x,y,z$ direction;
      mapping the boundary box into 3D space;
        $x_{corner}=\{l/2, l/2, -l/2, -l/2, l/2, l/2, -l/2, -l/2\}$;
        $y_{corner}=\{0,0,0,0,-h,-h,-h,-h\}$;
        $z_{corner}=\{w/2, -w/2, -w/2, w/2, w/2, -w/2, -w/2, w/2\}$;
              where the approximate values of the length, width and height
              of the object are $l, w, h$, respectively.
      mapping the 3D boundary box into 2D space;
        $space2D=P^*[space3D;1]$;
        $[x;y]_{mapping}=[x/z;y/z]$;
      image the category and score of the result
        by the semantic context information of the 2D result;
  return result{}
```

4. Results and Discussion

4.1. Implementation Details and Inputs

The method in this paper has been verified and experimentally discussed on KITTI for complex scenes in the field of autopilot technology, which is the most challenging and representative vision benchmark suite at present. Moreover, we compare this proposed framework with the existing segmentation and detection methods. The implementation, testing, and evaluation platform and the database configuration involved in this experiment are as follows.

Implementation, testing and evaluation platform and database configuration

- **Database:** *KITTI benchmark; Pascal VOC; ImageNet*
- **Platform:** *VisualStudio 2017; Matlab 2016a; Window 10 / Ubuntu 14; Ros rviz*
- **Implementation:** *C++ / python; opencv 3.4.0; pcl 1.8.1*
- **Configuration:** *CPU(IntelCore i7-7700K@4.20GHz×8)/GPU NVIDA GeForce GTX 1060*

  Velodyne HDL-64E rotating 3D laser scanner, 10 Hz, 64 beams;
  2×PointGray Flea2 color cameras (FL2-14S3C-C), 1.4 Megapixels,
  1/2″ Sony ICX267 CCD, global shutter

Figure 12 shows the Lidar point cloud of different viewing angles and the stereo image pairs obtained after the synchronized and calibrated sensor. The color of the point cloud represents the echo intensity of the Lidar.

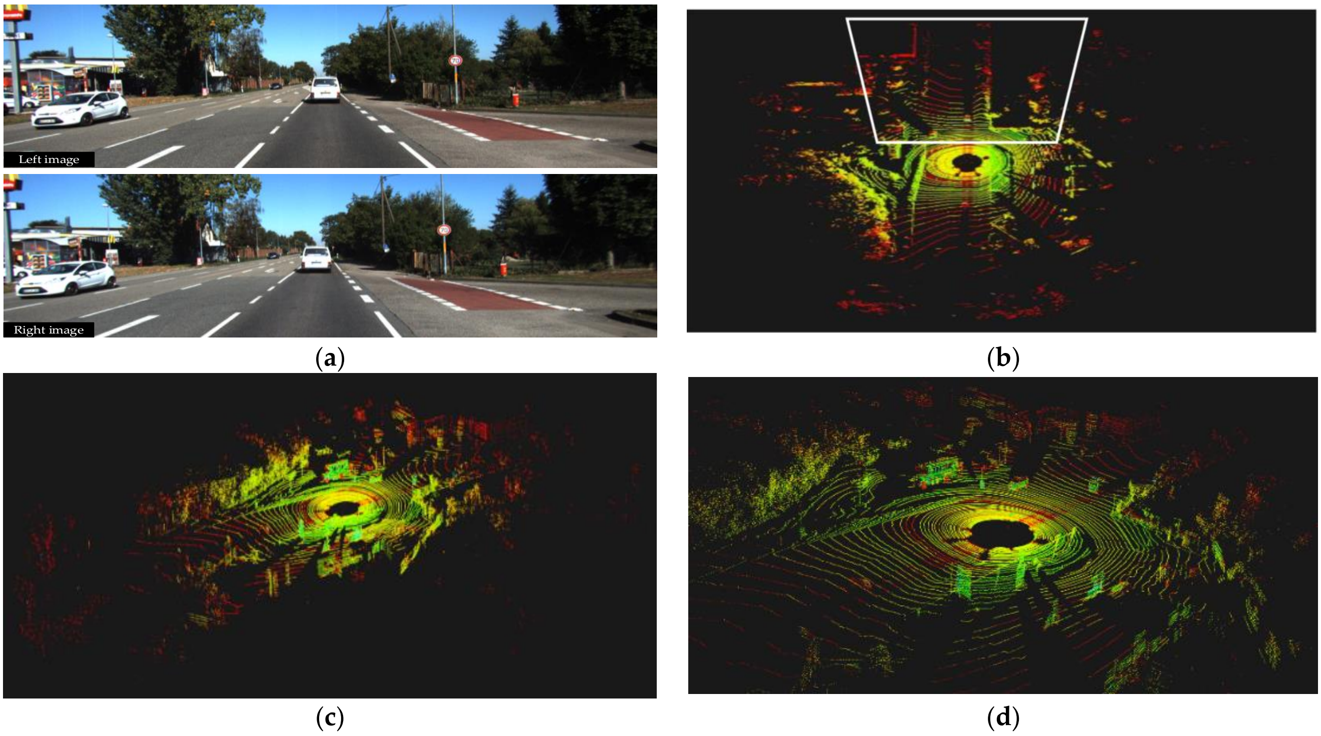


Figure 12. 2D and 3D data of the binocular color camera sensor and Lidar senor, respectively. (**a**) Stereo images; (**b**) top view of the point cloud (the trapezoid frame is the corresponding driving perspective area of the vehicle); (**c**) side view of the point cloud; (**d**) large-scale view of the point cloud.

4.2. Analysis of Fusion

As shown in Figure 13, the 2D and 3D display results of the multi-mode pre-fusion and alignment of 3D point clouds and 2D images under different sampling thresholds are shown. In the complex original point cloud data of Lidar, the current field of view is optimized effectively to reduce the redundant calculation and avoid the current invalid detection. Figure 14 shows the sampled point cloud data in the field of view after the fusion and alignment of the 3D Lidar point cloud and the camera sensor image. Figure 15 shows the fitting curves for the influence of different sampling thresholds n on the velodyne point cloud data and the shown cloud data results. It can be seen that the sampling points of the shown cloud are significantly lower than those of velodyne points, and the number of points drops sharply from the beginning with the increase in the threshold until 4, 5, 6 become flat. Therefore, on the premise of ensuring a certain amount of information, this paper adopts the sampling distance between 4 and 6. In the following experiments, this paper takes the average value of 5 for subsequent verification.

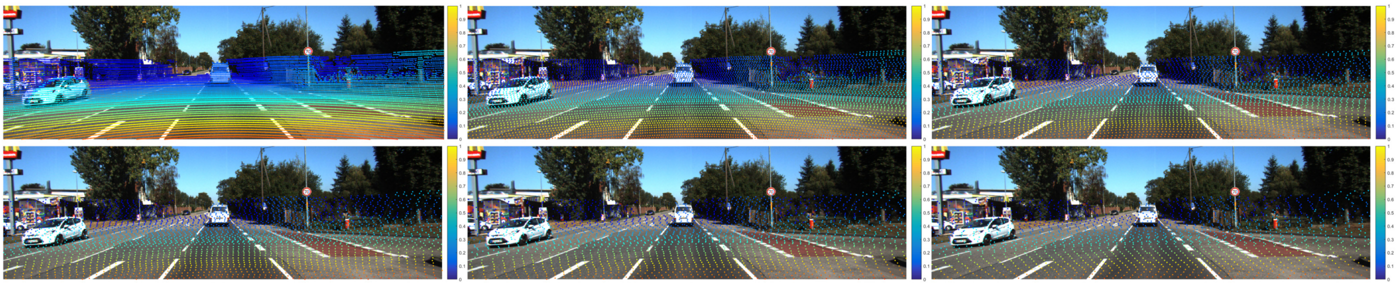


Figure 13. Multi-mode pre-fusion and alignment of 3D point clouds and 2D images under different sampling thresholds. The image shows the results for the threshold of the first few odd digits, i.e., the point cloud mapping map, in which the colorbar represents the depth of field, the warm color represents a close range, and the cold color represents a long range.

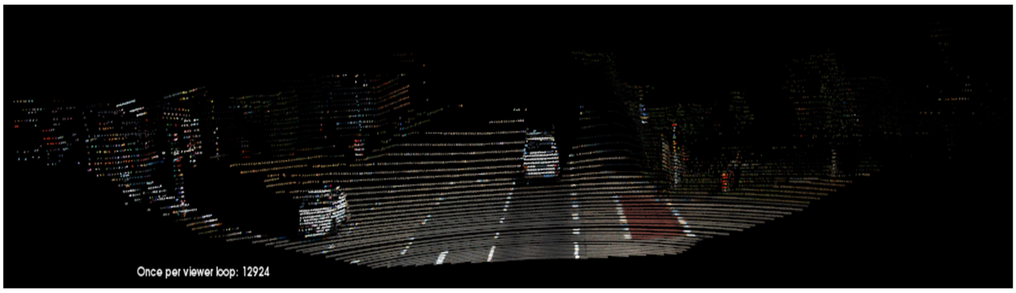


Figure 14. Sampling point cloud data in the field of view.

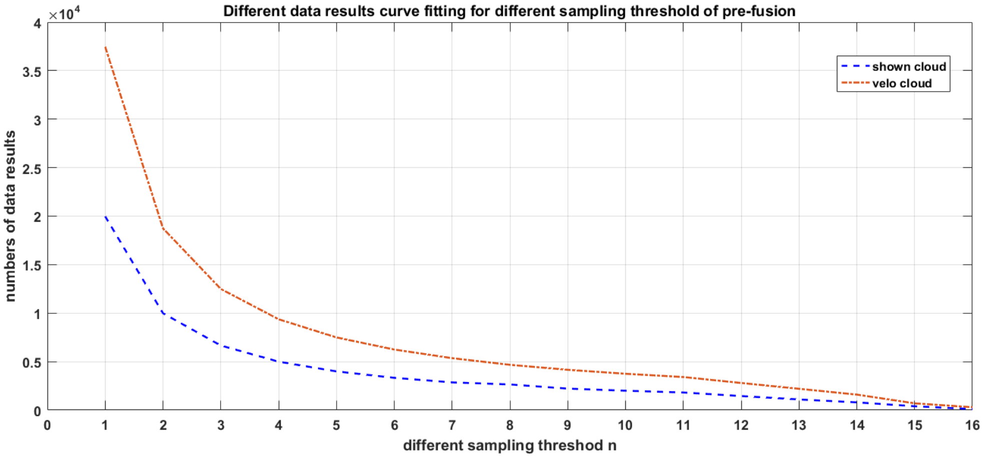


Figure 15. The curve fitting of different results for different sampling thresholds of pre-fusion.

4.3. Results of 2D Detection and 3D Segmentation

Figure 16 shows the detection results and the semantic information of the stereo regional proposal graph neural network, which outputs the categories and probabilities of 2D objects. The data cursor of the object will be further determined based on the 2D-driven results to restrict the indexing of the following precise and fast clouds' voxelization and over-segmentation. It can be seen that the network structure of this paper takes into account the regional proposal of the stereo image pair as well as the dilated convolution and double integrated loss function to expand the view field of calculation, so it has a certain detection effect gain and compensation amount for small objects, stacking, and occlusion.

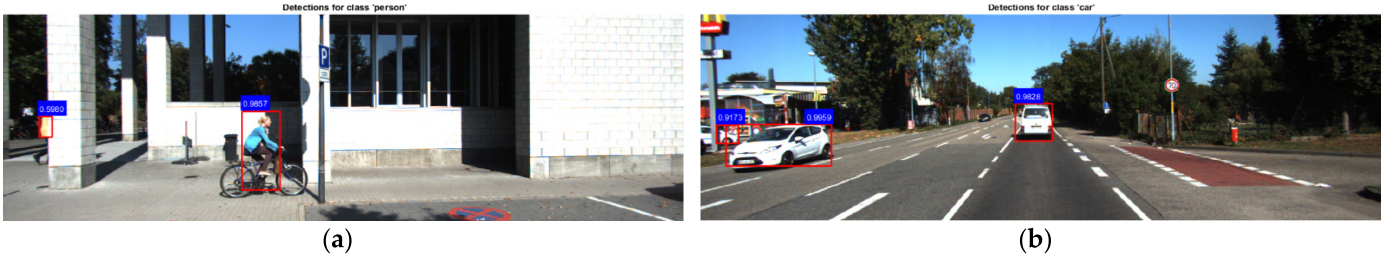


Figure 16. Detection results and semantic information of stereo regional proposal graph neural network. (**a**) Class Person; (**b**) Class Car.

In addition, different activation function curves are shown in Figure 17. Since the input of this paper is non-negative image data, the activation function adopts a linear rectification function of non-saturated function (Rectified Linear Unit, ReLU), also known as a modified linear unit. Compared with saturated functions such as Sigmoid and Tanh and other variant functions (Leaky ReLU, ELU, PReLU), ReLU can speed up the convergence of the model and solve the problem of gradient disappearance.

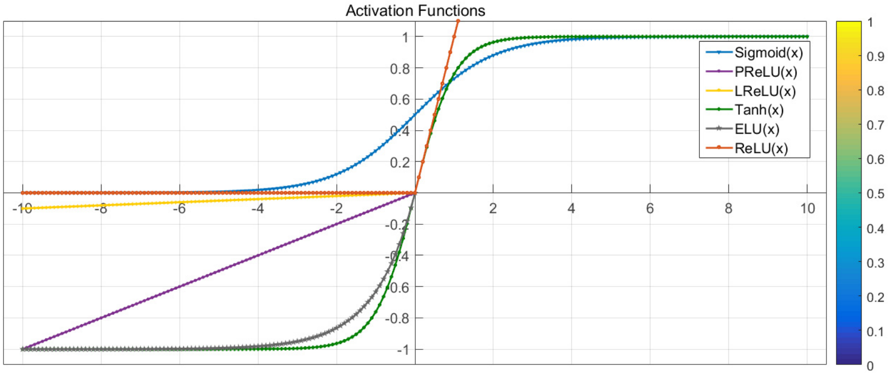
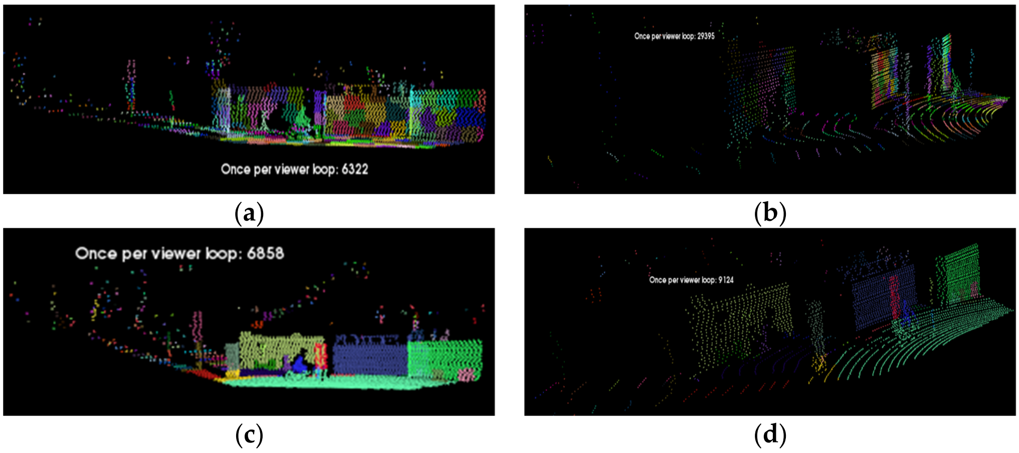


Figure 17. The different activation functions.

Figure 18a,b show our point cloud voxel-wise clustering results of the regional growth under the octree index structure. As we can see, the segmentation of voxels from random seeds of the point cloud has shown a preliminary trend of the object. The result of voxelization is partitioned into different random color values. In addition, Figure 18c,d are the results of over-segmentation of the quadratic hypervoxels' clustering. It can be seen that the over-segmentation effect is more obvious than the previous voxelization. The number of classifications is reduced to make the point cloud classification more targeted. The instance segmentation effect of the object is prominent, and the detection effect of the occlusion, connection, and other similar areas is remarkable. As shown in Figure 18c, the segmentation of person and bicycle is more accurate than other methods.


Once per viewer loop: 6322
Once per viewer loop: 29395
Once per viewer loop: 6858
Once per viewer loop: 9124

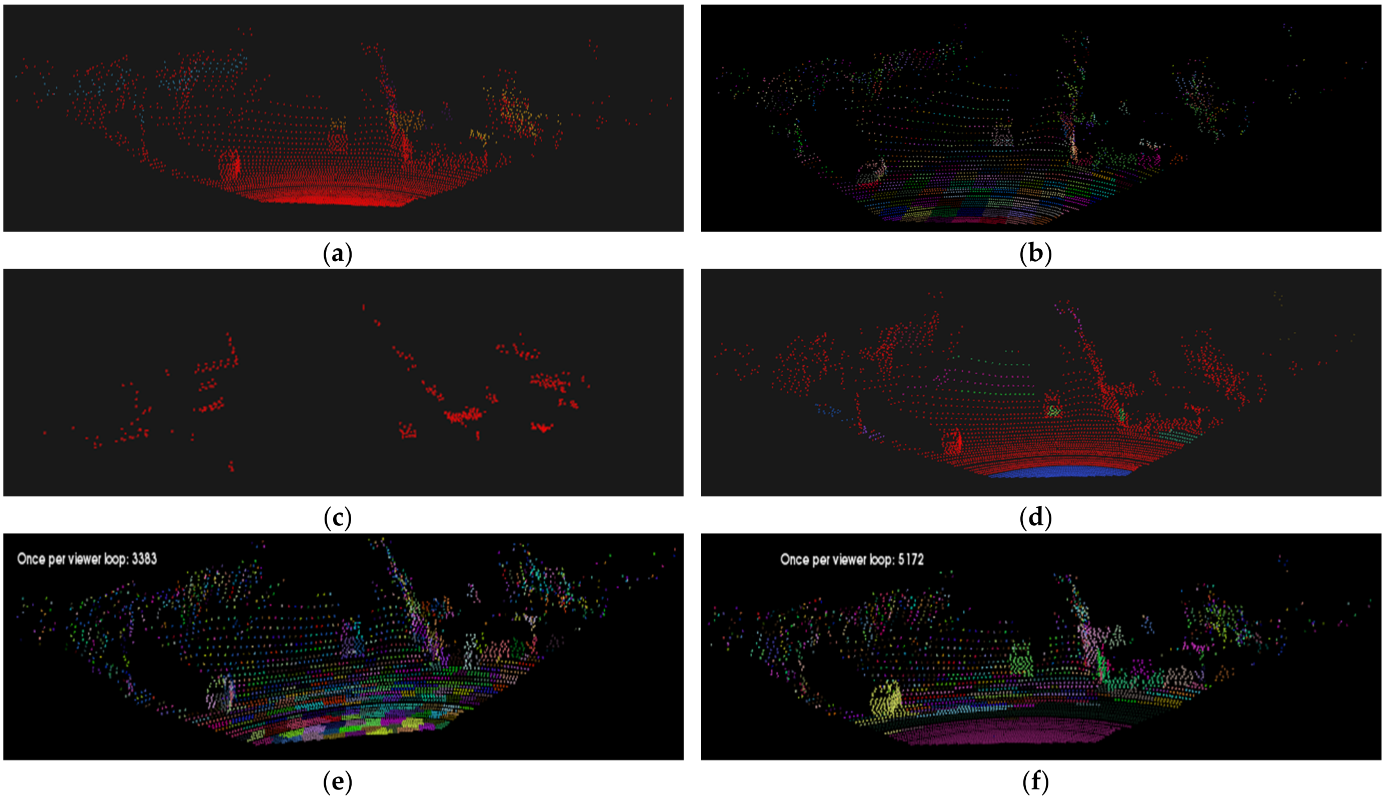
(**a**) (**b**)

(**c**) (**d**)

Figure 18. Region growth results of voxel clustering and hypervoxels' segmentation results. (**a**) Small-scale view; (**b**) Large-scale side view; (**c**) Small-scale view; (**d**) Large-scale side view.

4.4. Evaluation and Discussion for 3D Segmentation of 2D-Driven Results

Figure 19 shows the segmentation results of this paper and different methods. It can be seen that the segmentation category in this paper is clear, the object category and background category are clearly distinguished, and the segmentation is relatively correct.



(**a**) (**b**) (**c**) (**d**) (**e**) (**f**)

Figure 19. Segmentation results of different methods. (**a**) Color-based region growth method [38]; (**b**) Peer hypervoxel method [39]; (**c**) Minimum cut of graph model method [40]; (**d**) Distance grow method [41]; (**e**) Octree voxel method [42]; (**f**) Our result.

In this paper, we discuss our algorithm and different state-of-the-art single route segmentation algorithms programmed by the large-scale and open project of the Point Cloud Library (PCL) framework using quantitative indicators that benefit from the existing excellent indicators of two-dimensional pixel-level segmentation, some of which are also inspired by ideas in the literature [43]. The indicators' descriptions and Table 2 of the results are as follows:

- The point cloud accuracy (*PA*) represents the proportion of the point cloud of the predicted category to the point cloud aligned by the fusion of Lidar remote sensing and optical image sensing.
- The accuracy of the category point cloud (*CPA*) indicates the accuracy of the point cloud that actually belongs to category *i* in the prediction of category *i*.
- The mean point cloud accuracy (*MPA*) represents the average proportion of all types of point clouds in the point cloud after the sensor and camera sensor are fused and aligned.
- The average accuracy of the point cloud category (*MCPA*) indicates the average proportion of each predicted category point cloud to all categories.
- The average intersection of union (*MIOU*) represents the ratio of the intersection and union of the predicted category point cloud and the ground truth.

The higher the above indicators, the lower the omissions and errors of the point cloud instance segmentation and object detection.

- The horizontal positioning error (*HPE*) represents the difference between the centroid of the point cloud of the predicted category object and the ground truth in the north and east, that is, the component of the x-y coordinate axis.

- Object positioning error (*OPE*) indicates the difference of 3D rigid body motion between the centroid of the point cloud of the predicted category object and the ground truth.
- The average horizontal positioning error (*MHPE*) and the average object positioning error (*MOPE*) represent the average *HPE* and average *OPE* between the object point cloud and the ground truth for all prediction categories.

Table 2. Quantitative indicators of segmentation and detection results compared to some single route methods. Note: Larger values of *PA, CPA, MCPA, MPA, MIOU, RN* are preferred; smaller values of *HPE, OPE, MHPE, MOPE, Runtime* are preferred.

Methods	*PA*			*CPA*			*MCPA*	*MPA*	*MIOU*
	O1	**O2**	**O3**	**O1**	**O2**	**O3**			
Color-based region grow [38]	/	0.48%	/	/	24.68%	/	8.23%	0.16%	8.26%
Peer hypervoxel [39]	3.33%	1.98%	/	95.00%	100.00%	/	65.00%	1.77%	91.30%
Min-cut of graph model [40]	/	0.45%	/	/	23.38%	/	7.79%	0.15%	7.83%
Distance cluster [41]	/	0.43%	0.48%	/	22.08%	100.00%	40.69%	0.30%	13.04%
Octree voxel [42]	2.91%	1.75%	0.45%	82.86%	90.91%	100.00%	91.26%	1.70%	86.52%
OURS	3.46%	1.55%	0.33%	98.57%	80.52%	100.00%	**93.03%**	**1.78%**	**92.61%**

Methods	*HPE(m)*			*OPE(m)*			*MHPE(m)*	*MOPE(m)*	*Runtime(s)*	*RN(%)*
	O1	**O2**	**O3**	**O1**	**O2**	**O3**				
Color-based region grow	/	0.7989	/	/	1.1473	/	0.9330	1.0491	1.426	90%
Peer hypervoxel	0.1082	0.0927	/	0.1082	0.0945	/	0.4003	0.4009	1.155	70%
Min-cut of graph model	/	0.5492	/	/	1.1856	/	0.8497	1.0619	1.287	70%
Distance cluster	/	0.5266	0.4057	/	0.5529	0.4591	0.6441	0.6707	2.812	90%
Octree voxel	0.0156	0.0594	0.9409	0.1210	0.1548	0.9594	0.3387	0.4117	1.216	80%
OURS	0.0900	0.3779	0.0859	0.0901	0.3781	0.0942	**0.1846**	**0.1874**	**0.068**	**100%**

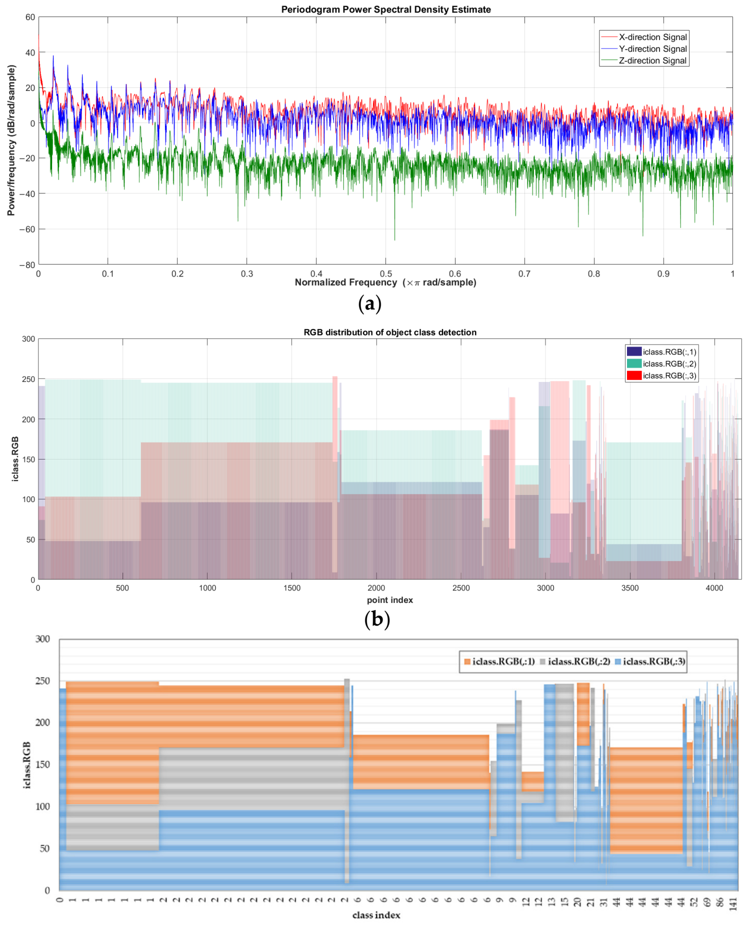
The lower the indicators of the above error values, the more accurate the point cloud centroid of the detected category object.

In addition, under a certain structure and size parameter perturbation, the system continues to perform *n* operations without interruption. Here, the average running time of one result output is calculated as the *Runtime*, and the probability that all detected results maintain consistent performance in a stable state without crashing is used as the robustness (*RN*).

It can be seen from Table 2 that the effective detection rate of this paper is higher than that of a single segmentation algorithm. The segmentation algorithms based on the color-based region growth and the minimum cut model can only extract a positive object O2, and the contralateral object O1 and the stacked occlusion object O3 are entirely invalid. Although the detection rate *PA* and the category accuracy *CPA* of pure hypervoxels' segmentation for the positive orientation object O2 are higher than our 1.55% and 80.52%, its predicted objects are segmented into *n* subclasses due to the presence of over-segmentation. In addition, it cannot detect the stacked occlusion object O3. Although the distance clustering method has a certain detection output for the stacked occlusion object O3, its relatively high detection rate *PA* is accompanied by false positives, i.e., over-clustering of the non-category object point clouds. The same problem exists for the octree voxelization method with the detection output for the stacked occlusion object O3. Therefore, we consider the advantages of octree voxelization, make full use of the spatial index structure of the point cloud, and increase the concave-convex growth strategy of geometric attributes to reduce the false positive detection and ensure effectual detection outputs of occluded objects. Our detection rate and each accurate index of all categories are higher than other single algorithms, where *MCPA, MPA,* and *MIOU* are 93.03%, 1.78%, and 92.61%, respectively.

Consistent with the above detection rate response, our method suppresses the false positive rate for the 3D point cloud instance segmentation and object detection in large-scale scenes due to the presence of 2D indexing of DAGNN under the stereo region constraint-driven conditions, resulting in lower localization errors not only in horizontal but also in the 3D rigid body motion for all object classes with valid detections. The localization errors *HPE* and *OPE* (0.0859 m, 0.0942 m) are much lower than the 0.9409 m and 0.9594 m of the octree voxel method when the localization errors of our method for the frontal and lateral orientation objects are relatively small. Moreover, the results of multiple runs are 100% effective, and the consistency probability is close to 100%, while the calculation errors of other methods are the accumulation of magnitude due to the existence of passive parameters, resulting in only 70% to 90% of the results on the running timestamps being stable output. In addition, compared to other single methods that directly or indirectly calculate the point cloud-level segmentation operation, the several contribution tricks in this paper all reduce the amount of data computation and speed up the running time to a certain extent.

By analyzing the results of over-segmentation detections, as shown in Figure 20a–c, the spectrum and the RGB distribution of the object class of segmentation detection results are given. It can be seen that the spectral relationship of the initial segmentation results has attenuation and noise in the time domain, and the spectral density is high. In the case of random assignment of RGB classes, the final object instance segmentation results tend to be stable, and the classification effect is evident.



(**a**)

(**b**)

(**c**)

Figure 20. The spectrum and the RGB distribution of the object classes for segmentation detection results. (**a**) The spectrum of voxelized over-segmentation result; (**b**) The RGB distribution of the object point cloud class detection; (**c**) The RGB distribution of object class detection with functional semantic information.

4.5. Object Detection and Visualized Mapping Results

The visualized mapping of the minimum three-dimensional rectangular bounding boxes for the object clustering in the LIDAR point cloud and the fusion data results are shown in Figures 21 and 22. The 3D bounding boxes basically frame objects in the point cloud data. In addition, the object with occlusion from the perspective of an intelligent vehicle camera also has certain detection feedback, as shown in Figure 21b, in which the visual field is shown in Figure 16.

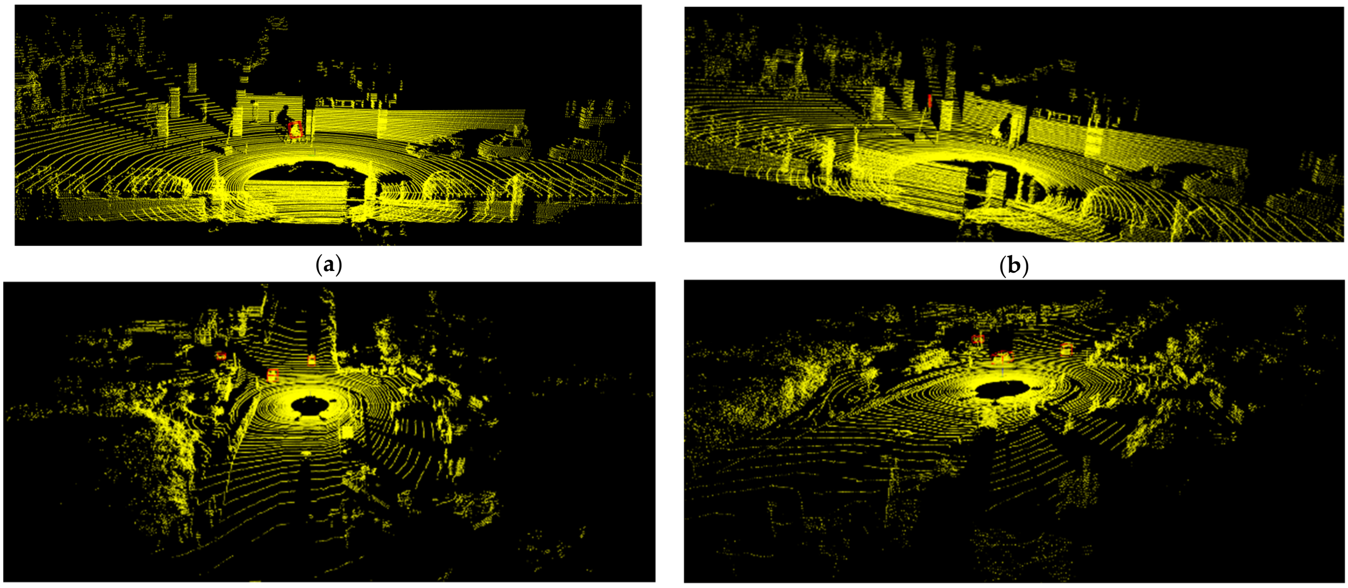

(**a**) (**b**)
(**c**) (**d**)

Figure 21. Mapping of minimum object bounding box in Lidar data. (**a**) Object1 of Figure 18; (**b**) Object2 of Figure 18; (**c**) Objects of Figure 19; (**d**) Objects of Figure 19.

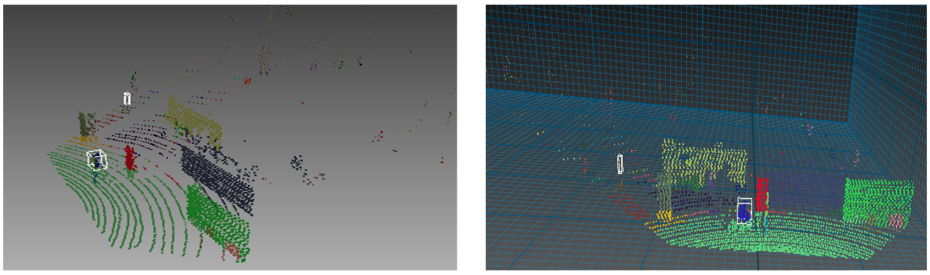

(**a**) (**b**)

Figure 22. Mapping of the minimum object bounding box in point cloud segmentation of fusion data. (**a**) Mapping in side view for ROS rviz shown without grids; (**b**) Mapping in frontal view for ROS rviz shown in grids.

The category distribution of the segmental result of point coordinates in Figure 23 tends to be stable, and the outliers and noise outliers can be suppressed. The clustering and the object detection effect of this paper are preferable. The visualization of 2D and 3D object boundary boxes and the attributed semantic information mapping are shown in Figure 24.

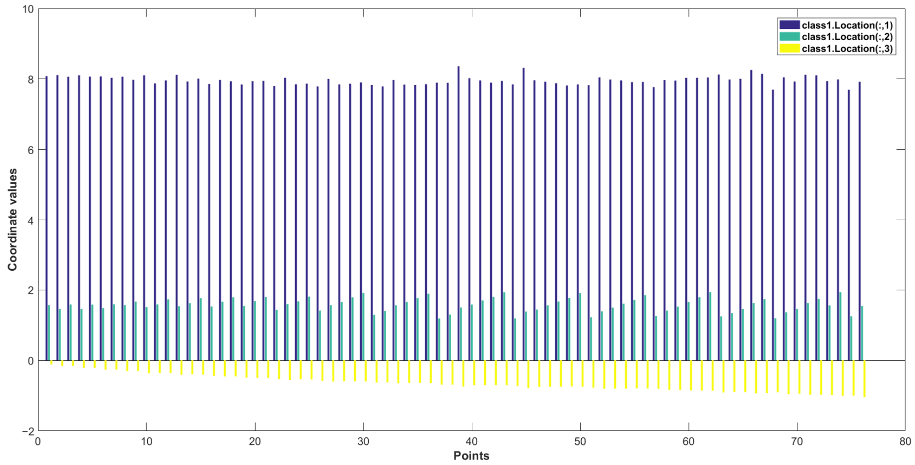
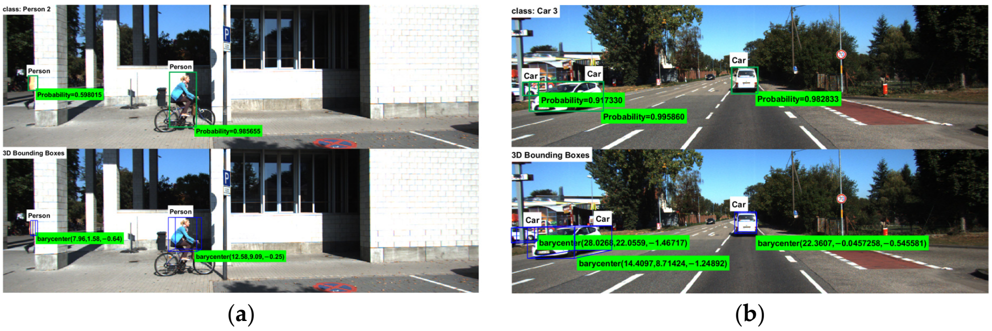


Figure 23. Point coordinate distribution map of the object class.



(**a**) (**b**)

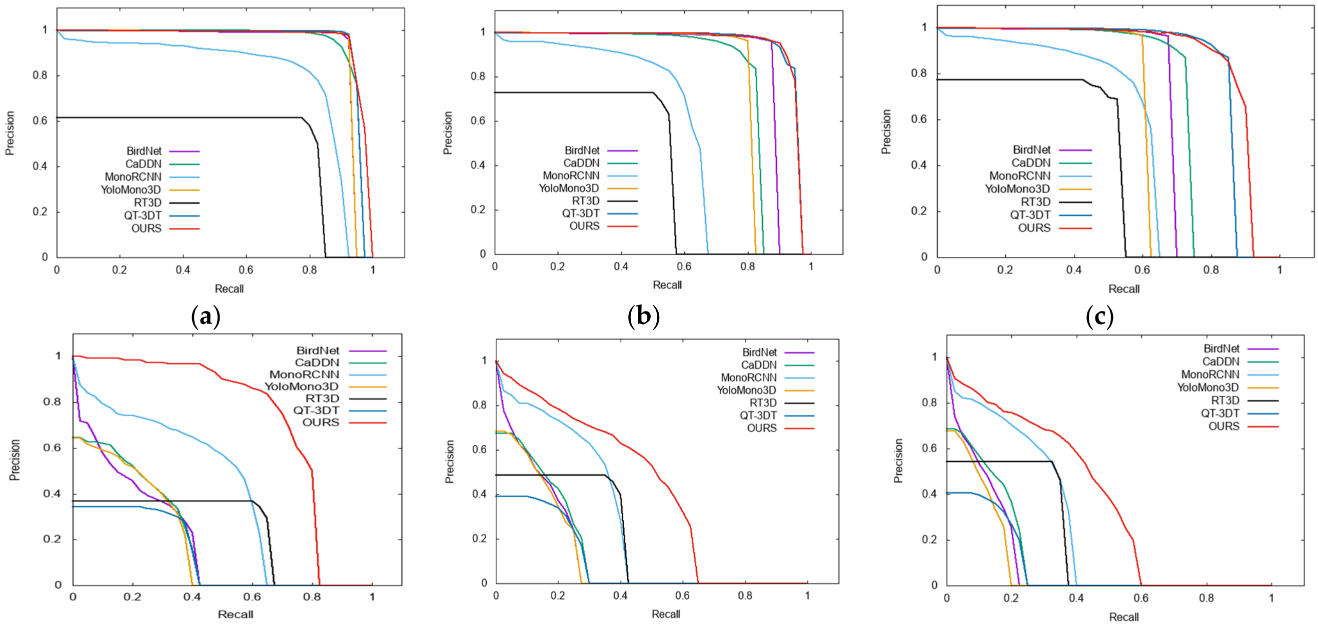
Figure 24. Mapping detection results of 2D bounding boxes and 3D bounding boxes with semantic information and attributes, in which Probability is the probability of class confidence and barycenter is the centroid coordinate of the object detection. (**a**) Class Person; (**b**) Class Car.

4.6. Comparison and Discussion for Object Detection

The precision (*P*) and recall (*R*) rates of different state-of-the-art methods [44–49] under different solution paths for object detection are evaluated. Based on the following three kinds of object detection difficulty degree, the *P*-*R* curves of 2D detection feedback are shown in Figure 25a–c, and the 3D detection feedback is shown in Figure 25d–f. It can be seen that the detection effect of this paper is almost close to *1-1* of the *P*-*R* curve under the object with an easy degree of difficulty and not much occlusion, and moderate and hard levels of occlusion do not hinder the significant output effect of this paper. The advantages of this paper can be clearly seen in the 3D feedback results. Since not only the data signals of a single image and a single sensor are considered, the advantages of Lidar and camera fusion are obvious, so the 3D object detection of the point cloud can be more effectively driven by the two-dimensional space. Among them, even if the degree of occlusion is relatively difficult, this paper can also generate the detection results with high accuracy and a low rate of missed detection.

Three kinds of object detection difficulty degree

- **Easy:** *Min. bounding box height: 40px; Max. occlusion level: Fully visible; Max.truncation: 15%*
- **Moderate:** *Min. bounding box height: 25px; Max. occlusion level: Partly occluded; Max.truncation: 30%*
- **Hard:** *Min. bounding box height: 25px; Max. occlusion level: Difficult to see; Max.truncation: 50%*



(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

Figure 25. The P-R curve of detection results with different methods (BirdNet [44], CaDDN [45], MonoRCNN [46], YoloMono3D [47], RT3D [48], QT-3DT [49], OURS). (**a**) 2D results of easy degree; (**b**) 2D results of moderate degree; (**c**) 2D results of hard degree; (**d**) 3D results of easy degree; (**e**) 3D results of moderate degree; (**f**) 3D results of hard degree.

Furthermore, this method is unlike the 3D object shape retrieval mechanism approaches for object detections, for example, the approaches in [3,4,7], which strongly rely on the correspondences between the model to be detected and the stored model, such as the repetitive, descriptive, and quantitative features, and they are limited to small-scale scenes and objects at a specific scale or the specific granularity knowledge. Inspired by the idea of retrieval, this paper uses the detection of 3D objects under the 2D region constraints, but independent of their correspondence, which is effective for object detection in large-scale scenes, especially for applications to urban traffic scenes. Similarly, compared to the recall rate of 59.13% for the simplest degree of the incomplete object detection in the 3D data retrieval mechanism in [3], this paper can reach more than 80%. Furthermore, compared to the dynamic distance clustering segmentation method of 3D point cloud data in [9] for 3D object detection with an average precision (*AP*) of 64.05%, this paper is 8.59 percentage points higher. This is due to the idea of hybrid segmentation in this paper, which adopts the growth strategy under the spatial index structure of the point cloud with spatial location, color attributes, local geometric features, and hypervoxels' concave-convex geometric attributes. This makes the point cloud segmentation more accurate, and the *AP* of the object detection is high. Furthermore, it is equally effective for the occluded truncated objects. The approach in [10] is based solely on the 2D image attention mechanisms for object detection, and the performance of its model will degrade and may fail when the background in the image is complex, or the depth map quality is poor. In contrast to the algorithm in [10], this paper adds the multi-dimensional information interaction with 3D point cloud data to the stereo region selection search DAGNN model, making it effective and robust even for scenes with complex backgrounds or obscured truncated objects.

In summary, the proposed framework of the Lidar sensor and camera sensor fusion for 3D object detection is fairly complete. The multi-dimensional information interaction makes the final detection information richer and more reliable. The multi-source fusion of multi-mode data also makes the results of single processing more accurate and stable. The parallel pipeline processing of multi-tier architecture also makes the implementation of the framework more efficient. It is particularly worth noting here that because this paper is based on the fusion of stereo images graph neural network and Lidar point clouds, there is a certain dependence on the sparseness and the density of the point clouds. Once the collection is too sparse, it may cause the lever of the detection result to be biased towards the output of the stereo image position, and the three-dimensional information, especially the positioning parameters of the object, may encounter certain calculation errors. In addition, it is a pity that the verification in this paper does not consider the use of GPU and fine-tuning training, so the efficiency is not high. Compared with the millisecond-level calculations of other algorithms, the second-level calculations in this paper are indeed worth pondering and improving. In the future, considering the GPU and more effective point cloud processing and fusion computing, better object detection results and higher efficiency will be obtained. This is also where we will learn and commit to these excellent algorithms in the future.

5. Conclusions

This paper proposes a 3D instance segmentation and object detection framework based on the fusion of Lidar remote sensing and optical image sensing.

Firstly, the coordinate system transformation, fusion, and alignment of sensor data under the synchronization and rectification effectively reduce the complexity of redundant noise data processing. Compared with pure Lidar point cloud data, our experiment reduces the amount of redundant data for processing by 93.94% and can reduce the redundant data by 69.71% at least.

Secondly, the meaningful stereo regional proposal selective search-driven graph neural network provides a certain positioning and semantic information feedback for small objects, object stacking, and object occlusion. It is effective if the detection category probability of occluding stacked objects exceeds 50%.

Then, based on the octree voxelized point cloud, combining the two-dimensional information, multi-point cloud features, and unique concave-convex properties, the remarkable hypervoxels' clustering growth and instance segmentation of the object point cloud are realized. The calculation speed becomes fast, and the segmentation is effective. Compared with the single segmentation algorithm, the object detection rates *MCPA, MPA, MIOU* in this paper are high. Compared with the ground truth, the positioning errors of this paper are not only lower in *MHPE*, but also lower in the positioning *MOPE* in the three-dimensional rigid body motion space. Moreover, the object category of the point clouds' segmentation detection under the region constraint is more accurate.

Finally, the significant visualized 2D/3D positioning and semantic information of the object provide the basis for the intelligent navigation system. Compared with other monocular algorithms and classifier algorithms, the *P-R* curves of detection feedback for two-dimensional objects with different difficulty levels are closer to 1, i.e., the upper right corner. Among them, the average precision (*AP*) of 15% truncation objects, i.e., the area under the curve, is 96.66%, the *AP* of 30% truncation objects is 94.88%, and the *AP* of 50% truncation objects is 87.96%. Compared with the lower 3D object detection feedback accuracy of these algorithms with different truncation and occlusion levels, the *P-R* curve of the algorithm in this paper is obviously better, and the *AP* of all 3D objects with different difficulty levels exceeds 50%. Among them, 15% of truncated objects have an *AP* of 72.64%.

Leveraging the Lidar 3D signal data and the 2D image pixel data can compensate for the failure of a single sensor under complicated conditions such as severe weather, complex traffic environments, and weak illumination, and can also play certain interesting coordination and remedial measures. However, as mentioned in the results and discussion sections, this paper has a certain dependence on the sparse density of the point cloud and does not consider the fine-tuning training and the use of GPU. Therefore, more accurate data source tags for the object classification in the future deserve to be considered.

In the future, we will continue to improve the more efficient and fast performance of this structure, as well as the mobile communication technology based on high speed, low latency, and large connections. If the communication technology and cloud computing mode of the intelligent network connection are taken into consideration, the UAV, vehicle, satellite, roadside unit, and other vehicles can conduct the integrated data exchange in V2X-based space–ground integration. It is worth researching and considering the location storage link layer of moving objects and stationary objects, respectively.

Author Contributions: Conceptualization, L.B.; methodology, L.B.; software, L.B.; validation, L.B.; formal analysis, Y.L.; resources, L.B.; data curation, L.B.; writing—original draft preparation, L.B.; writing—review and editing, L.B.; visualization, L.B.; supervision, Y.L.; project administration, F.H.; funding acquisition, Y.L. and M.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Chongqing Key Technology Innovation and Application Development Project, grant number cstc2019jscx-mbdxX0004, cstc2020jscx-msxmX0153; Research and Innovation Project for Postgraduate of Chongqing, grant number CYB18173; Doctoral Innovative High-end Talents Project, grant number BYJS201808; State Scholarship Fund of China Scholarship Council; grant number 201908500145; National Natural Science Foundation of China, grant number 61773082.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data available in a publicly accessible repository.

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

References

1. Pang, C.; Zhong, X.; Hu, H.; Tian, J.; Peng, X.; Zeng, J. Adaptive Obstacle Detection for Mobile Robots in Urban Environments Using Downward-Looking 2D LiDAR. *Sensors* **2018**, *18*, 1749. [CrossRef]
2. Guo, Y.; Bennamoun, M.; Sohel, F.; Lu, M.; Wan, J. 3D Object Recognition in Cluttered Scenes with Local Surface Features: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 2270–2287. [CrossRef]
3. Guo, Y.; Wen, C.; Sun, X.; Wang, C.; Li, J. Partial 3D Object Retrieval and Completeness Evaluation for Urban Street Scene. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019; pp. 1252–1255.
4. Zou, K.; Zhang, Z.; Zhang, J.; Zhang, Q. A 3D model feature extraction method using curvature-based shape distribution. In Proceedings of the IEEE International Conference on Fuzzy Systems & Knowledge Discovery, Zhangjiajie, China, 15–17 August 2015; pp. 1809–1813.
5. Mokhtarian, F.; Khalili, N.; Yuen, P. Curvature Computation on Free-Form 3-D Meshes at Multiple Scales. *Comput. Vis. Image Underst.* **2001**, *83*, 118–139. [CrossRef]
6. Hung, C.-C.; Kulkarni, S.; Kuo, B.-C. A New Weighted Fuzzy C-Means Clustering Algorithm for Remotely Sensed Image Classification. *IEEE J. Sel. Top. Signal Process.* **2010**, *5*, 543–553. [CrossRef]
7. Garro, V.; Giachetti, A. Scale Space Graph Representation and Kernel Matching for Non-Rigid and Textured 3D Shape Retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 1258–1271. [CrossRef] [PubMed]
8. Chen, B.; Chen, H.; Yuan, D.; Yu, L. 3D Fast Object Detection Based on Discriminant Images and Dynamic Distance Threshold Clustering. *Sensors* **2020**, *20*, 7221. [CrossRef] [PubMed]
9. Zhou, W.; Pan, S.; Lei, J.; Yu, L.; Zhou, X.; Luo, T. Three-branch architecture for stereoscopic 3D salient object detection. *Digital Signal Process.* **2020**, *106*, 1051–2004.
10. Luo, Q.; Ma, H.; Tang, L.; Wang, Y.; Xiong, R. 3D-SSD: Learning hierarchical features from RGB-D images for amodal 3D object detection. *arXiv* **2017**, arXiv:1711.00238. [CrossRef]
11. Ong, J.; Vo, B.-T.; Kim, D.Y.; Nordholm, S. A Bayesian Filter for Multi-view 3D Multi-object Tracking with Occlusion Handling. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *12*, 6009–6027. [CrossRef]
12. Awadallah, M.; Abbott, L.; Ghannam, S. Segmentation of sparse noisy point clouds using active contour models. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Paris, France, 27–30 October 2014; pp. 6061–6065.
13. Wang, Y.; Shi, H. A Segmentation Method for Point Cloud Based on Local Sample and Statistic Inference. *Geoinform. Resour. Manag. Sustain. Ecosyst.* **2015**, *482*, 274–282.
14. Li, L.; Yang, F.; Zhu, H.; Li, D.; Li, Y.; Tang, L. An Improved RANSAC for 3D Point Cloud Plane Segmentation Based on Normal Distribution Transformation Cells. *Remote Sens.* **2017**, *9*, 433. [CrossRef]
15. Zhao, C.; Guo, H.; Lu, J.; Yu, D.; Zhou, X.; Lin, Y. A new approach for roof segmentation from airborne LiDAR point clouds. *Remote Sens. Lett.* **2021**, *12*, 377–386. [CrossRef]
16. Xu, B.; Jiang, W.; Shan, J.; Zhang, J.; Li, L. Investigation on the Weighted RANSAC Approaches for Building Roof Plane Segmentation from LiDAR Point Clouds. *Remote Sens.* **2016**, *8*, 5. [CrossRef]

17. Zhang, J.; Cao, J.; Liu, X.; Chen, H.; Li, B.; Liu, L. Multi-Normal Estimation via Pair Consistency Voting. *IEEE Trans. Vis. Comput. Graph.* **2018**, *25*, 1693–1706. [CrossRef]
18. Dey, E.; Kurdi, F.T.; Awrangjeb, M.; Stantic, B. Effective Selection of Variable Point Neighbourhood for Feature Point Extraction from Aerial Building Point Cloud Data. *Remote Sens.* **2021**, *13*, 1520. [CrossRef]
19. Bergamasco, F.; Pistellato, M.; Albarelli, A.; Torsello, A. Cylinders extraction in non-oriented point clouds as a clustering problem. *Pattern Recognit.* **2020**, *107*, 107443. [CrossRef]
20. Dirk, H.; Stefan, H.; Radu, B.R.; Sven, B. Real-Time Plane Segmentation Using RGB-D Cameras. In *Robot Soccer World Cup*; Springer: Cham, Switzerland, 2011; Volume 7416, pp. 306–317.
21. Hu, F.; Tian, Z.; Li, Y.; Huang, S.; Feng, M. A Combined Clustering and Image Mapping based Point Cloud Segmentation for 3D Object Detection. In Proceedings of the Chinese Control and Decision Conference, Shenyang, China, 9–11 June 2018; pp. 1664–1669.
22. Luo, H.; Zheng, Q.; Wang, C.; Guo, W. Boundary-Aware and Semiautomatic Segmentation of 3-D Object in Point Clouds. *IEEE Geosci. Remote Sens. Lett.* **2021**, *18*, 910–914. [CrossRef]
23. Vo, A.-V.; Truong-Hong, L.; Laefer, D.; Bertolotto, M. Octree-based region growing for point cloud segmentation. *ISPRS J. Photogramm. Remote Sens.* **2015**, *104*, 88–100. [CrossRef]
24. Li, L.; Yao, J.; Tu, J.; Liu, X.; Li, Y.; Guo, L. Roof Plane Segmentation from Airborne LiDAR Data Using Hierarchical Clustering and Boundary Relabeling. *Remote Sens.* **2020**, *12*, 1363. [CrossRef]
25. Hasirci, Z.; Ozturk, M. The comparison of region growing algorithms with using EMST for point clouds. In Proceedings of the International Conference on Telecommunications and Signal Processing (TSP), Prague, Czech Republic, 9–11 July 2015; pp. 1–5.
26. Wu, H.; Zhang, X.; Shi, W.; Song, S.; Tristan, A.C.; Li, K. An Accurate and Robust Region-Growing Algorithm for Plane Segmentation of TLS Point Clouds Using a Multiscale Tensor Voting Method. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *12*, 4160–4168. [CrossRef]
27. Strom, J.; Richardson, A.; Olson, E. Graph-based segmentation for colored 3D laser point clouds. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 2131–2136.
28. Tatavarti, A.; Papadakis, J.; Willis, A.R. Towards real-time segmentation of 3D point cloud data into local planar regions. In Proceedings of the SoutheastCon, Concord, NC, USA, 30 March–2 April 2017; pp. 1–6.
29. Zhang, S.; Cui, S.; Ding, Z. Hypergraph Spectral Clustering for Point Cloud Segmentation. *IEEE Signal Process. Lett.* **2020**, *27*, 1655–1659. [CrossRef]
30. Sd, A.; Mhb, C.; Nk, D.; Pk, A. Combining graph-cut clustering with object-based stem detection for tree segmentation in highly dense airborne lidar point clouds. *ISPRS J. Photogramm. Remote Sens.* **2021**, *172*, 207–222.
31. Charles, R.Q.; Wei, L.; Chenxia, W.; Hao, S.; Leonidas, J.G. Frustum PointNets for 3D Object Detection from RGB-D Data. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 918–927.
32. Lin, Z.-H.; Huang, S.Y.; Wang, Y.-C.F. Learning of 3D Graph Convolution Networks for Point Cloud Analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**. [CrossRef] [PubMed]
33. Cui, Y.; Liu, X.; Liu, H.; Zhang, J.; Zare, A.; Fan, B. Geometric attentional dynamic graph convolutional neural networks for point cloud analysis. *Neurocomputing* **2021**, *432*, 300–310. [CrossRef]
34. Wang, J.; Xu, C.; Dai, L.; Zhang, J.; Zhong, R.Y. An Unequal Learning Approach for 3D Point Cloud Segmentation. *IEEE Trans. Ind. Inform.* **2021**. [CrossRef]
35. Nagy, B.; Benedek, C. On-the-Fly Camera and Lidar Calibration. *Remote Sens.* **2020**, *12*, 1137. [CrossRef]
36. Geiger, A.; Moosmann, F.; Car, O.; Schuster, B. A toolbox for automatic calibration of range and camera sensors using a single shot. In Proceedings of the International Conference on Robotics and Automation (ICRA), Saint Paul, MN, USA, 14–18 May 2012.
37. Bai, L.; Li, Y.; Hu, F.; Zhao, F. Region-proposal Convolutional Network-driven Point Cloud Voxelization and Over-segmentation for 3D Object Detection. In Proceedings of the IEEE Chinese Control and Decision Conference (CCDC), Nanchang, China, 3–5 June 2019; pp. 3553–3558.
38. Qingming, Z.; Yubin, L.; Yinghui, X. Color-Based Segmentation of Point Clouds. *Laser Scanning* **2009**, *38*, 155–161.
39. Stein, S.C.; Schoeler, M.; Papon, J.; Worgotter, F. Object partitioning using local convexity. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 304–311.
40. Golovinskiy, A.; Funkhouser, T. Min-cut based segmentation of point clouds. In Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV), Kyoto, Japan, 27 September–4 October 2009; pp. 39–46.
41. Rabbani, T.; Van Den Heuvel, F.; Vosselmann, G. Segmentation of point clouds using smoothness constraint. *Int. Arch. Photogramm. Remote. Sens. Spat. Inf. Sci.* **2006**, *36*, 248–253.
42. Papon, J.; Abramov, A.; Schoeler, M.; Worgotter, F. Voxel Cloud Connectivity Segmentation—Supervoxels for Point Clouds. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 2027–2034.
43. David, C.; Nafornita, C.; Gui, V.; Campeanu, A.; Carrie, G.; Monnerat, M. GNSS Localization in Constraint Environment by Image Fusing Techniques. *Remote Sens.* **2021**, *13*, 2021. [CrossRef]

44. Beltran, J.; Guindel, C.; Moreno, F.M.; Cruzado, D.; Garcia, F.; De La Escalera, A. BirdNet: A 3D Object Detection Framework from LiDAR Information. In Proceedings of the IEEE 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 1–7.
45. Reading, C.; Harakeh, A.; Chae, J.; Waslander, S.L. Categorical depth distribution network for monocular 3d object detection. In Proceedings of the International Conference on Computer Vision and Pattern Recognition, Paris, France, 19–25 June 2021; pp. 1–11.
46. Shi, X.; Ye, Q.; Chen, X.; Chen, C.; Chen, Z.; Kim, T.K. Geometry-based distance decomposition for monocular 3D object de-tection. *arXiv* **2021**, arXiv:2104.03775.
47. Liu, Y.; Wang, L.; Liu, M. Yolostereo3D: A step back to 2D for efficient stereo 3D detection. *arXiv* **2021**, arXiv:2103.09422.
48. Zeng, Y.; Hu, Y.; Liu, S.; Ye, J.; Han, Y.; Li, X.; Sun, N. RT3D: Real-Time 3-D Vehicle Detection in LiDAR Point Cloud for Autonomous Driving. *IEEE Robot. Autom. Lett.* **2018**, *3*, 3434–3440. [CrossRef]
49. Hu, H.N.; Yang, Y.H.; Fischer, T.; Darrell, T.; Yu, F.; Sin, M. Monocular quasi-dense 3D object tracking. *arXiv* **2021**, arXiv:2103.07351.