remote sensing

MDPI

Article

3D Sensor Based Pedestrian Detection by Integrating Improved HHA Encoding and Two-Branch Feature Fusion

Fang Tan, Zhaoqiang Xia *, Yupeng Ma and Xiaoyi Feng

School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710060, China; bjtanfang@mail.nwpu.edu.cn (F.T.); mayupenga@mail.nwpu.edu.cn (Y.M.); fengxiao@nwpu.edu.cn (X.F.)
* Correspondence:zxia@nwpu.edu.cn

Abstract: Pedestrian detection is vitally important in many computer vision tasks but still suffers from some problems, such as illumination and occlusion if only the RGB image is exploited, especially in outdoor and long-range scenes. Combining RGB with depth information acquired by 3D sensors may effectively alleviate these problems. Therefore, how to utilize depth information and how to fuse RGB and depth features are the focus of the task of RGB-D pedestrian detection. This paper first improves the most commonly used HHA method for depth encoding by optimizing the gravity direction extraction and depth values mapping, which can generate a pseudo-color image from the depth information. Then, a two-branch feature fusion extraction module (TFFEM) is proposed to obtain the local and global features of both modalities. Based on TFFEM, an RGB-D pedestrian detection network is designed to locate the people. In experiments, the improved HHA encoding method is twice as fast and achieves more accurate gravity-direction extraction on four publicly-available datasets. The pedestrian detection performance of the proposed network is validated on KITTI and EPFL datasets and achieves state-of-the-art performance. Moreover, the proposed method achieved third ranking among all published works on the KITTI leaderboard. In general, the proposed method effectively fuses RGB and depth features and overcomes the effects of illumination and occlusion problems in pedestrian detection.

Keywords: 3D sensor; multi-modal data; pedestrian detection; HHA; feature fusion

Citation: Tan, F.; Xia, Z.; Ma, Y.; Feng, X. 3D Sensor Based Pedestrian Detection by Integrating Improved HHA Encoding and Two-Branch Feature Fusion. *Remote Sens.* **2022**, *14*, 645. https://doi.org/10.3390/rs14030645

Academic Editor: Joaquín Martínez-Sánchez

Received: 18 December 2021
Accepted: 27 January 2022
Published: 29 January 2022

1. Introduction

The importance of pedestrian detection cannot be overstated. It is the basis for many computer vision tasks and is widely used in video surveillance, robotics, automatic driving and healthcare assistance. Benefitting from convolution neural network (CNN) methods and large-scale datasets, RGB image based pedestrian detection performance has made great progress with the development of RGB cameras in recent years. However, some challenges are difficult to solve if only RGB visual information is exploited, such as insufficient light, occlusion, small-scale objects and dense crowds, as shown in Figure 1a–d. Unlike RGB images, depth images generated from 3D sensors are not affected by illumination and can provide three-dimensional information, but have the problems of low resolution and high noise. Therefore, combining RGB and depth images (RGB-D data) to detect people can compensated for the other's shortcomings and provide the possibility to solve those challenges. Many pedestrian detection methods [1–6] based on RGB-D data have been presented. They also demonstrated that the using of multi-modality RGB-D data is better than using the single modality form. However, two issues with regard to using RGB-D data are worthy of further investigaton: (1) how to use depth images; and (2) how to fuse the RGB and depth modalities.

For the issue of using depth images, it is unsuitable to directly feed it into a deep network, as the depth image is usually a single channel and 16-bit image. Therefore, the depth information is usually encoded as grayscale or a three-channel pseudo-color

image (as shown in Figure 2). Among them, horizontal disparity, height above ground, and the angle of the pixel's local surface normal makes with the inferred gravity direction (HHA) [7] contains more information in three channels. Moreover, HHA has been effectively applied to semantic segmentation [8], scene understanding [9], and salient object detection [10]. However, it performs inadequately at pedestrian detection. For example, some studies [11,12] have shown that the detection performance of converting depth information to a jet colormap is better than that of converting it to HHA. But the HHA carries more information than the jet colormap. We found that the HHA encoding only extracts the gravity direction, so it is easy to introduce noises when calculating the channel of the height above ground, which seriously affects pedestrian detection. To address this problem, we propose to improve the HHA encoding method by extracting the ground plane parameters instead of just extracting the ground direction. The improved algorithm effectively reduces the error encoding of height above ground.

(a) Insufficient Light (b) Occlusions

(c) Small Scale (d) Dense Crowds

Figure 1. Visualization of RGB and depth images in some challenging pedestrian detection scenarios.

For the issue of fusing multi-modal information, most existing RGB-D [2,4] or RGB-T [13,14] pedestrian detection algorithms explored the feature maps of two modalities in weighted fusion. This process possibly obtains local features of two modalities, since the additive fusion of two corresponding channels is independent of other channels. Employing 1×1 convolution after concatenated two feature maps is also a common fusion method [1,3]. In contrast, this method extracts global features because each channel of the convolution output is a linear combination of all input channels. In our opinion, local features are as important as global features. Therefore, we designed a two-branch network to fuse these two modalities in both global and local views.

To this end, this paper proposes an RGB-D data based pedestrian detection network which takes RGB and improved HHA images as inputs. In the proposed model, an improved HHA method is firstly proposed to encode the depth image more accurately. We also perform depth equalization for the mapping of the depth values, which improves the color space utilization. The improved method runs twice as fast and estimates the direction of gravity more accurately. Then, we design a two-branch feature fusion extraction module (TFFEM) which uses two branches to extract the global and local features separately. In the local feature extraction, we also propose an adaptive channel cross fusion (ACCF) module to learn the weights of each channel and to become more efficient. We validate the designed network on two publicly-available datasets and achieve state-of-the-art detection performance. One dataset is KITTI [15]. It employs the LiDAR sensor to capture the point cloud, which can still be encoded as an HHA image. Another dataset is EPFL [16], whose

depth images are acquired directly by the 3D sensor of Kinect v2. In summary, the main contributions of this paper are summarized as

- We design an RGB-D data based pedestrian detection network which achieves the state-of-the-art detection performance on KITTI and EPFL datasets.
- We improve the HHA encoding method, which is twice as fast and can extract the full ground parameters. Moreover, the detection performance outperforms other encoding methods.
- We propose two new modules, i.e., TFFEM and ACCF, in the deep network, which can learn rich multimodal features.

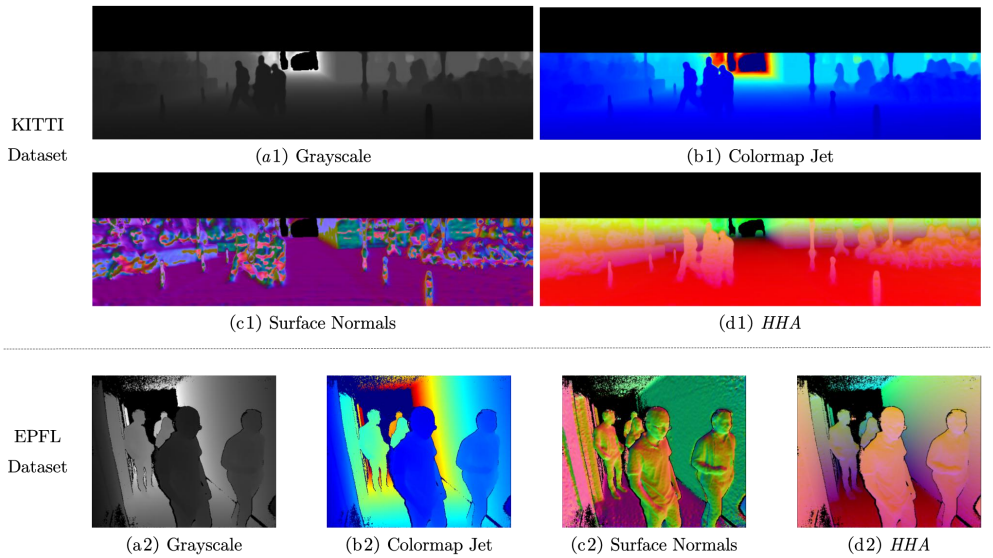


Figure 2. Sample images after encoding the depth information via different methods on the two datasets.

2. Related Work

2.1. Single Modal Based Pedestrian Detection

2.1.1. Depth Image Based Approaches

There are many pedestrian detection algorithms for single-depth images, and they can be classified into three main types, i.e., traditional methods, CNN-based methods, and the combination of the traditional and CNN model methods. These methods require high depth image quality, so it is not sufficient to only use the depth image outdoors or in large spaces.

Traditional methods generally require several steps. In the preprocessing stage, many studies [17,18] use background modeling algorithms to remove background regions, and some algorithms use plane detection to remove areas that contain planes. In the region of interest(ROI) extraction stage, the methods for depth images are also various. Hu et al. [17] use spatial clustering to obtain ROI. Tian et al. [19] utilize height information to remove ROI that contains points that are too high or low. And, more often, the head information is exploited because the size and shape of the person's head is usually fixed in space, so researchers propose many head-based features [18–21] to filter the ROI. However, the problem with these methods is that they can only detect the upper body region of the people and cannot detect the people with the head obscured.

The traditional method combined with CNN [19] usually uses CNN to classify the extracted ROI at the end. Since most invalid ROIs have been removed by depth information, the speed of these methods is also significant. Meanwhile, some studies [22,23] proposed end-to-end pedestrian detection networks that only use depth images. However, these methods are still based on head features.

2.1.2. RGB Image Based Approaches

Since the RPN network was proposed in the two-stage object detection method Faster RCNN [24], the performance of CNN-based pedestrian detection algorithms [25] has greatly exceeded that of traditional methods [26–28]. Later, single-stage object detection methods were proposed, such as YOLO [29,30], SSD [31], and EfficientDet [32], which are faster and have high detection accuracy. Many studies of pedestrian detection are based on them. The above methods require many pre-defined anchors, which reduces the flexibility of the network in training. Meanwhile, some anchor-free object detection algorithms such as CenterNet [33] and FCOS [34] have been proposed. Correspondingly, the CenterNet-based pedestrian detection algorithm CSP [35] has achieved high detection performance. Moreover, there are also some networks designed specifically for pedestrian detection, such as TAFT [36], and W3Net [37]. RGB image based pedestrian detection algorithms have made significant progress. However, RGB images do not provide 3D information and are sensitive to illumination, which may significantly degrade the detection performance of these algorithms in changing environments.

2.2. Multi-Modal Based Pedestrian Detection

Multimodal data are more stable than single modality data, and also contain richer information. Therefore, many RGB-D pedestrian detection methods have been presented in recent years. These methods mainly address two issues: the fusion position of two modals, and the fusion method. Earlier approaches [38–41] only fused the classification results of each modality without interaction in feature extraction. They used adaptive joint probabilities to combine the scores of each modality. Based on the Faster-RCNN, Ren et al. [12] adopted two parallel VGG networks to extract features and fused them in the last layer while leaving the rest unchanged. Ophoff et al. [1,42] verified through extensive experiments that fusing two features extracted in the middle network layer is superior to fusion the early and late ones. In addition. There are also many studies that focus more on fusion methods. Zhang et al. [2,4] referred to the SENet [43] and computed weight for each channel after combining the two features.

Some studies also focused on other elements. Guo et al. [5] obtained the mask of people's target labels by clustering depth images and combined this information to improve the stability of pedestrian detection. Some studies [44,45] proposed multitask models that enable the network to output the box and distance information to enhance the feature representation. Linder et al. [3] improved the network by adding a synthetic RGB-D dataset that is rendered with a semi-realistic game engine.

Furthermore, some different encoding strategies have been explored for using depth information in multimodal data. The pixel values of the raw depth image indicate the distance from the object to the camera. Since this value varies greatly for different devices and units, it is not directly suitable for training. The efficient way is to encode the depth image to a grayscale one (as shown in Figure 2a) according to Equation (1),

$$f(Z_{uv}) = \frac{Z_{uv} - \min(Z)}{\max(Z) - \min(Z)} \tag{1}$$

where Z denotes the depth value, and uv refers to the image coordinates. Existing deep learning networks are designed for three-channel color images, so many methods further encode grayscale into pseudo-color images such as jet colormap (as shown in Figure 2b) before training.

The above encoding methods, although efficient, ignore the 3D information provided by the depth. The depth image can be converted to a point cloud by camera intrinsic. Some methods extract the spatial information in the point cloud and then invert it back to the image plane to generate a new pseudo-color image, which carries adequate information. Tian et al. [19] proposed the DMH representation, which contains three channels of raw depth value, a multi-order depth template, and a height difference map. Currently, the most-commonly used depth encoding method is HHA [7,46] (as shown in Figure 2d).

3. Proposed Method

This section contains two parts. The first part introduces the improved HHA method for using depth information. We describe the original HHA encoding and its shortcomings. The improvements in gravity direction estimation and depth value mapping are also presented. The second part introduces our proposed RGB-D data-based pedestrian detection network and its submodules.

3.1. Improved HHA Encoding

The original HHA image is in three-channel, and each channel is obtained by encoding original depth information. We aim to improve the encoding procedure in two components (described in Sections 3.1.3 and 3.1.4) while obtaining a same-size image.

3.1.1. Vanilla HHA

The first channel in an HHA image is horizontal disparity, which is inversely proportional to the depth value (denoted as z) obtained from 3D sensors. The converted value in each pixel is computed according to Equation (2).

$$d = \frac{f * b}{z} \tag{2}$$

where d is the disparity of each pixel in the first channel, and f, b are the focal length and the baseline of one camera.

For the pixel values in the second and third channels, the gravity height and the angle with the gravity direction of each point are calculated respectively. The gravity direction is estimated as follows. Firstly, the depth value z is converted to a point-cloud value, and the normal is calculated [46]. Next, the gravity direction (i.e., normal of the ground plane) is estimated iteratively by point normal. The initial step is to set $\mathbf{g_o} = [0, 1, 0]^T$ (the default camera view is a horizontal forward view). All point normal is $\mathbf{n}$. The second step is to get the parallel normal set $N_{\parallel}$ and the horizontal normal set $N_{\perp}$ in $\mathbf{n}$ according to Equation (3)

$$\begin{aligned} N_{\perp} &= \{\mathbf{n_i} | \theta(\mathbf{n_i}, \mathbf{g_o}) < T_d \cup \theta(\mathbf{n_i}, \mathbf{g_o}) > 180 - T_d\} \\ N_{\parallel} &= \{\mathbf{n_i} | \theta(\mathbf{n_i}, \mathbf{g_o}) > 90 - T_d \cap \theta(\mathbf{n_i}, \mathbf{g_o}) < 90 + T_d\} \end{aligned} \tag{3}$$

where $\theta(\mathbf{a}, \mathbf{b})$ denotes the angle between the three-dimensional vectors $\mathbf{a}$ and $\mathbf{b}$, and T_d denotes the angle threshold. The third step is to estimate the new gravity direction using $N_{\parallel}$ and $N_{\perp}$. The estimation process is equivalent to searching the $\mathbf{g_n}$ that minimizes Equation (4)

$$\min_{\mathbf{g}:\|\mathbf{g_n}\|_2=\mathbf{1}} \sum_{\mathbf{n_i} \in N_{\perp}} \cos^2(\theta(\mathbf{n_i}, \mathbf{g_n})) + \sum_{\mathbf{n_i} \in \mathbf{N}_{\parallel}} \sin^2(\theta(\mathbf{n_i}, \mathbf{g_n})) \tag{4}$$

where $\mathbf{g_n}$ need to be normalized to satisfy $\|\mathbf{g_n}\|_2 = 1$. According to the trigonometric function, the Equation (4) can be rewritten as Equation (5).

$$\begin{aligned} &\sum_{\mathbf{n_i} \in N_{\perp}} \cos^2(\theta(\mathbf{n_i}, \mathbf{g_n})) + \sum_{\mathbf{n_i} \in N_{\parallel}} -\cos^2(\theta(\mathbf{n_i}, \mathbf{g_n})) \\ &= \mathbf{g_n}(N_{\perp} N_{\perp}^T - N_{\parallel} N_{\parallel}^T) \mathbf{g_n}^T \end{aligned} \tag{5}$$

Then according to the Courant-Fischer theorem from linear algebra, the $\mathbf{g_n}$ can be obtained by decomposing the matrix $N_{\perp} N_{\perp}^T - N_{\parallel} N_{\parallel}^T$. After that, $\mathbf{g_n}$ is used as the initial direction $\mathbf{g_o}$. The above steps are repeated until the stop condition is met.

The vanilla HHA [7] adopts the angle thresholds T_d to 45° and 15° and iterates five times under each threshold. After getting the gravity direction, the angle of each point is calculated directly. And the gravity height is calculated by first rotating the whole point cloud to the horizontal direction and then subtracting the smallest y coordinate value from the y coordinate value of each point.

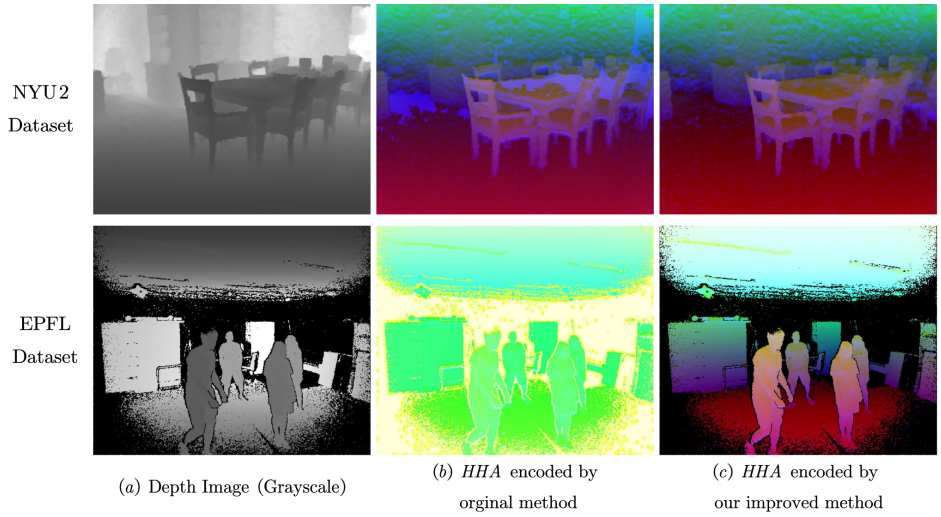
3.1.2. Shortcoming Analysis of Vanilla HHA

The method of gravity estimation in vanilla HHA uses horizontal elements (floor, ceiling, etc.) and vertical ones (wall, etc.) in the scene, which has high stability but suffers from some shortcomings.

First, it only estimates the gravity direction and cannot obtain the complete ground plane equation. The plane equation is as in Equation (6)

$$Ax + By + Cz + D = 0 \tag{6}$$

The gravity direction $\mathbf{g} = [A, B, C]^T$. Hence, there is also a missing parameter D. Due to the lack of this parameter, the original method is calculating the height by subtracting the smallest y coordinate value. This calculation is sensitive to the camera pose and noise. And if the complete ground plane parameters are available, the height can be calculated directly by the point-to-plane distance equation. As shown in Figure 3b, the original method makes a difference in generating HHA images on the NYU2 and EPFL datasets.



(*a*) Depth Image (Grayscale) (*b*) *HHA* encoded by orginal method (*c*) *HHA* encoded by our improved method

Figure 3. The encoded HHA images by the original and our improved methods on different datasets.

Secondly, the original method requires ten iterations. Each iteration has to process all points, which is inefficient and susceptible to noise interference. This would cause poor accuracy of the estimated gravity direction.

3.1.3. Improved Gravity Direction Estimation

To address the shortcomings of vanilla HHA, we improve the gravity direction estimation in HHA encoding by considering the ground plane parameters and removing a part of the points after each iteration.

Our ground estimation process is designed as follows. In the first step, after obtaining the gravity direction $\mathbf{g_n}$ according to the original method, we calculate the ground plane parameter D using all the parallel points $P_{\|}$ according to Equation (7)

$$D = \frac{1}{N} \sum_{\mathbf{p} \in P_{\|}} \mathbf{p}^T \cdot \mathbf{g_n} \tag{7}$$

where $\mathbf{p} = [p_x, p_y, p_z]^T$ denotes the points, and N is the number of parallel points. In the second step, combining the obtained ground plane equation, only the points that satisfy the Equation (8) are retained

$$P_s = \{\mathbf{p} | \mathbf{p}^T \cdot \mathbf{g_n} + D > T_{up} \cap \mathbf{p}^T \cdot \mathbf{g_n} + D < T_{down}\} \tag{8}$$

where T_{up} and T_{down} are the manual thresholds. The idea of removing points is to consider that the points belonging to the ground are under the estimated plane, so threshold T_{up} is set to keep the points under the plane. Meanwhile, the threshold T_{down} is set to remove the points too far from the plane, preventing noise interference.

Our estimation process is iterated only three times, and the final ground plane parameters are obtained using all parallel point fits at the end of the three iterations. The final ground equation was fitted using least squares estimation, where the process is to calculate the parameters $-\frac{A}{C}, -\frac{B}{C}, -\frac{D}{C}$ so that Equation (9) takes the minimum value.

$$\min \sum_{p \in P_{\parallel}} (-\frac{A}{C} p_x - \frac{B}{C} p_y - \frac{D}{C} - p_z)^2 \tag{9}$$

The whole iterative process is shown in Figure 4a1,a2,b1,b2,c1,c2, in which the red points represent the vertical points $P_{\perp}$ and the blue points represent the parallel points $P_{\parallel}$. It can be seen in Figure 4, the number of points decreases in each iteration, and the final estimated ground is shown as the green points in Figure 4d1 d2. In addition, the threshold values for the three iterations are T_d = [45°, 30°, 10°], T_{up} = [−30 cm, −15 cm, −5 cm], and T_{down} = [+∞, +∞, 15 cm]. The improved gravity direction extraction method reduces the number of iterations and removes redundant points each time to improve the fitting accuracy. After obtaining the ground equation, the point-to-surface distance is directly used as the gravity height channel of HHA images, which does not require manual adjustment of parameters and can be more adaptable to different data. The subsequent experimental results also prove the superiority of our method.

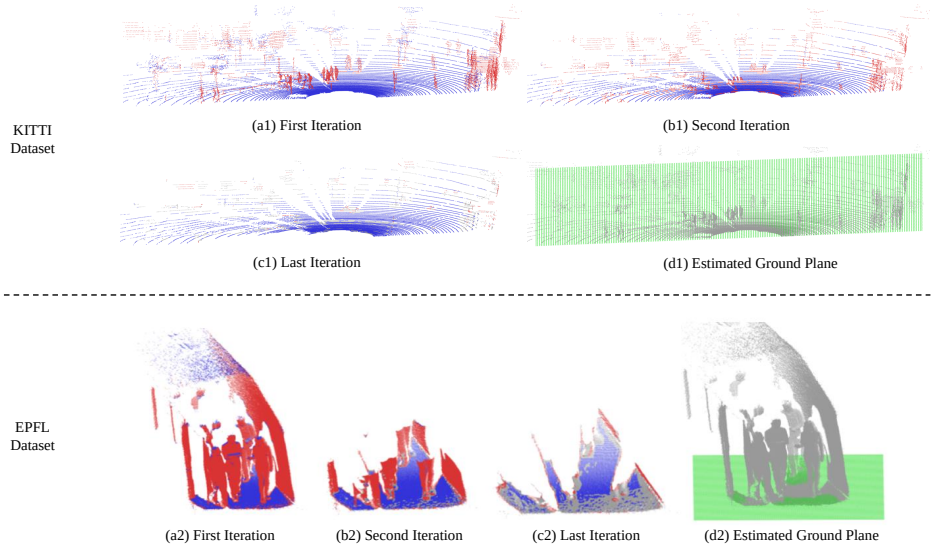
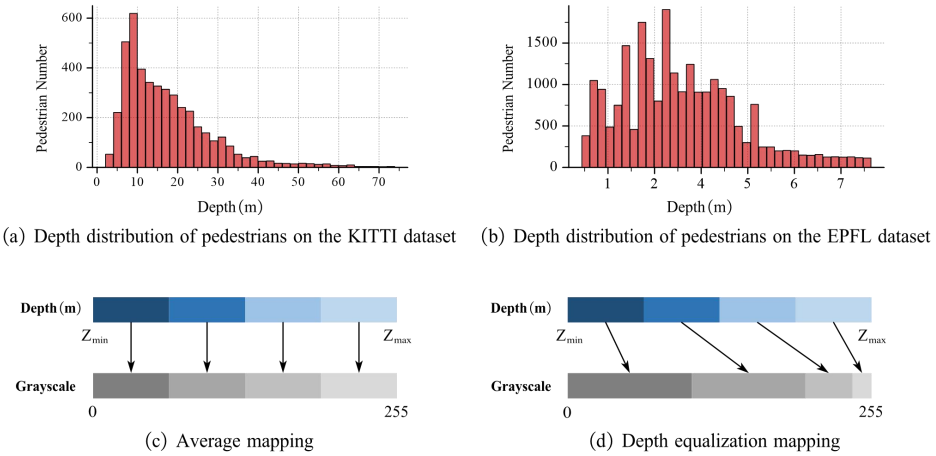


Figure 4. The iterative processes of our improved gravity direction estimation on the KITTI and EPFL datasets.

3.1.4. Improved Depth Value Mapping

Moreover, we update the depth value mapping to facilitate the calculation of the first channel in improved HHA. The first channel of HHA images is mapping the horizontal disparity to grayscale. Therefore, we directly use the depth values instead of the horizontal disparity. The mapping method uses average mapping in HHA encoding. The depth value is evenly arranged to the grayscale of 0–255, as shown in Figure 5c. This mapping is simple and easy to implement. However, the accuracy of most 3D sensing is proportional to the distance, and the depth error is larger when the distance is farther. Therefore, more objects are spread in the front and middle of the depth range, as shown in Figure 5a,b. As can be seen, there are already very few objects with a distance of more than 40 m in the KITTI dataset. Hence, if the depth values are equally mapped on the grayscale an imbalance will

result. For example, the distant regions containing few objects occupy the same range on the grayscale as the near regions containing more objects. To address this problem, we propose a depth equalization mapping (DEM) method, as shown in Figure 5d.



(a) Depth distribution of pedestrians on the KITTI dataset (b) Depth distribution of pedestrians on the EPFL dataset

(c) Average mapping (d) Depth equalization mapping

Figure 5. The depth-value distribution of pedestrian objects on two RGB-D datasets (**a**,**b**) and the two different mapping methods (**c**,**d**).

The DEM refers to the histogram equalization (HE) algorithm. The mapping process can be seen as computing the mapping function $g = f(z)$ so that the depth values z are converted to grayscale values g. f is computed by Equation (10)

$$f(z_i) = L \sum_{z=0}^{z_i} H(z) \tag{10}$$

where L is the maximum range of the grayscale (256), and $H(z)$ is the histogram of the depth values for all objects. This histogram is calculated as Equation (11)

$$H(z_i) = p(z = z_i) = \frac{n_{z_i}}{n}, z_{\min} < z_i < z_{\max} \tag{11}$$

where n is the number of all objects and n_{z_i} is the number of objects with depth z_i.

3.2. Two-Branch Pedestrian Detection Network

3.2.1. Network Overview

The structure of the proposed RGB-D data-based pedestrian detection network is shown in Figure 6. The network inputs the RGB images and HHA images encoded by the improved method. Two parallel ResNet50 [47] networks are employed to extract features from each input separately. Since it is a detection task, the average pooling and fully connected layer of the ResNet50 are removed. Then the feature maps $\mathbf{F}_n^{RGB} \in \mathbb{R}^{H \times W \times C}$ and $\mathbf{F}_n^{HHA} \in \mathbb{R}^{H \times W \times C}$ output from the same layers are fused (excluding the first layer). Where H, W, and C are the output feature map's height, width, and channel number. N is the index of layers. The fusion method is the multimodal fusion and attention module (MFAM) described in the next. Then the fusion result F_n^{Fused} is used as the input of the feature pyramid network (FPN). Finally, existing detection head modules such as FCOS [34] and Faster RCNN [24] can be directly connected to the FPN to detect people. We next select the head modules of Faster RCNN.

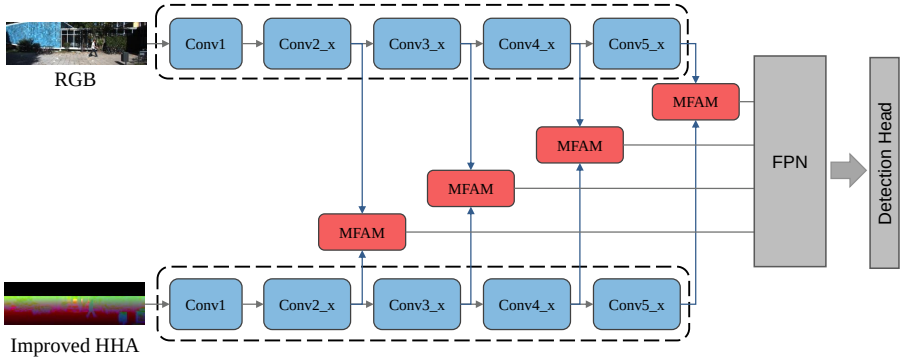


Figure 6. The structure of our pedestrian detection network.

The structure of MFAM is shown in Figure 7, which consists of two main parts: the TFFEM and the attention module. Each of them is described below.

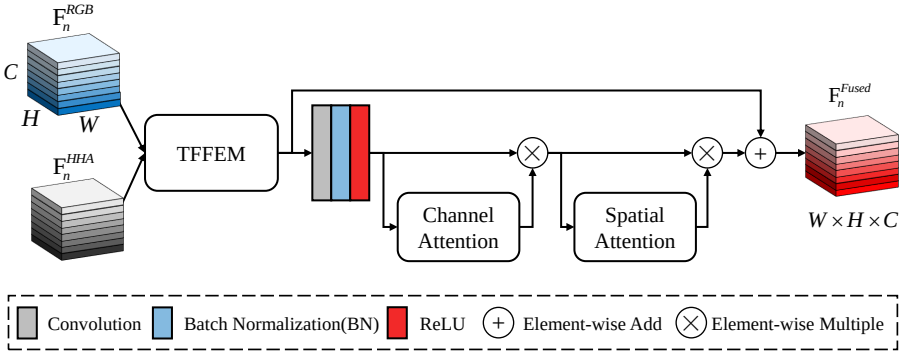


Figure 7. The structure of our proposed MFAM.

3.2.2. Two-Branch Feature Fusion Extraction Module

The main idea of TFFEM is to learn the global and local features of two modalities to enhance the ability of networks for interpreting multimodal features. The structure is shown in Figure 8. Since the detection object is a single class and to improve the module efficiency, we first reduce the channel number of the input feature maps ($\mathbf{F}_n^{RGB}$, and $\mathbf{F}_n^{HHA}$) by 50% using 1×1 convolution to obtain the new feature maps ($(\hat{\mathbf{F}}_n^{RGB}, \hat{\mathbf{F}}_n^{HHA}) \in \mathbb{R}^{H \times W \times \frac{C}{2}}$). Then two branches are constructed to extract the global feature map $\hat{\mathbf{F}}_n^{Glob} \in \mathbb{R}^{H \times W \times \frac{C}{2}}$ and local feature map $\hat{\mathbf{F}}_n^{Loc} \in \mathbb{R}^{H \times W \times \frac{C}{2}}$ from the new feature maps. Finally, the output feature map $\mathbf{F}_n^{TFF} \in \mathbb{R}^{H \times W \times C}$ of TFFEM is obtained by concatenation of the global and local feature maps as Equation (12)

$$\mathbf{F}_n^{TFF} = Cat(\hat{\mathbf{F}}_n^{Glob}, \hat{\mathbf{F}}_n^{Loc}) \tag{12}$$

The process of extracting global features is first to concatenate $\hat{\mathbf{F}}_n^{RGB}$ and $\hat{\mathbf{F}}_n^{HHA}$. Then a standard 3×3 convolution is used to obtain the global feature map. Since the standard convolution is employed, each channel in the output feature map is acquired by the joint calculation of all channels in the input feature map. So it can get the global features.

The local features are extracted by our proposed ACCF block. First, each channel in feature maps $\hat{\mathbf{F}}_n^{RGB}$ and $\hat{\mathbf{F}}_n^{HHA}$ with scale $W \times H \times \frac{C}{2}$ is cross-arranged to generate a new feature map with size $W \times H \times C$. The new feature map is then processed by group convolution with a kernel size of 1×1. The number of groups and output channels are all $\frac{C}{2}$. After group convolution, the local feature map $\hat{\mathbf{F}}_n^{Loc}$ is obtained. Each channel of $\hat{\mathbf{F}}_n^{Loc}$ is computed from the channels corresponding to $\hat{\mathbf{F}}_n^{RGB}$ and $\hat{\mathbf{F}}_n^{HHA}$ without involving other ones. So ACCF can extract the local features. Furthermore, the group convolution can be seen as assigning a weight to each channel of both feature maps, which is more

flexible than the direct add. ACCF is also better than the hard threshold adding operation proposed in [48], where all channels are multiplied by the same weight. Obviously, it is more reasonable to make the network learn the weights of each channel by itself.

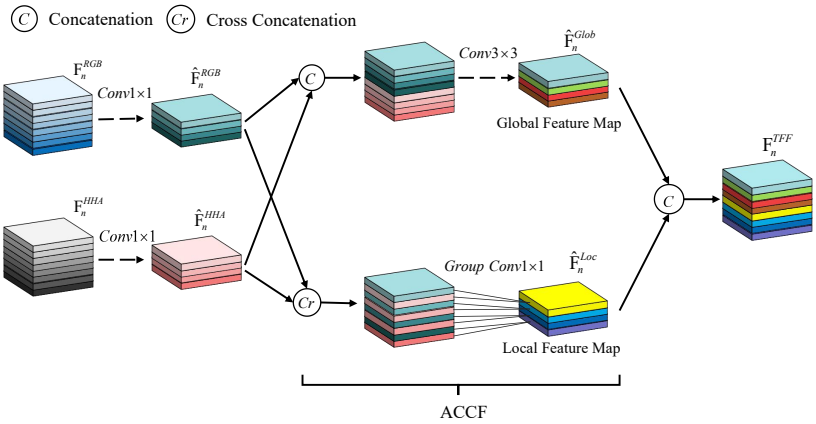


Figure 8. The detail of the proposed TFFEM. It contains two branches: (1) concatenating two modalities and learning the global channel features, and (2) learning local channel features by the ACCF block proposed in this paper.

3.2.3. Attention Module

The fused feature map obtained by TFFEM contains both global and local features between two modalities, and this feature map can continue to be fed into the attention module. It is different from the CW or ACW fusion module proposed in [4,13]. They refer to the idea of the attention module to calculate the weights of each channel.

The convolutional block attention module (CBAM) [49] is chosen in our network, which includes channel attention and spatial attention. In addition, a standard convolution module is added before the CBAM for preprocessing the input feature maps. Finally, the output feature map $\mathbf{F}_n^{Fused}$ by the MFAM is computed according to Equation (13)

$$\mathbf{F}_n^{Fused} = \mathbf{W}_s \odot (\mathbf{W}_c \odot Conv(\mathbf{F}_n^{TFF})) + \mathbf{F}_n^{TFF} \tag{13}$$

where $\mathbf{W}_c$, $\mathbf{W}_s$ represents the weights of channel and spatial attention module. $\odot$ represents the element-wise multiplication. *Conv* denotes a standard convolution, including the 3×3 convolution, batch normalization (BN), and activation function ReLU. $\mathbf{F}_n^{TFF}$ is the feature map obtained by the TFFEM.

4. Experiments

The experiments consist of four parts. The first two parts evaluate the performance of our proposed pedestrian detection method on two publicly-available datasets. The third part compares our improved HHA encoding method with the original one from the metrics of accuracy and speed of the gravity directions estimation. The last part is the ablation study, which first verifies the superiority of improved HHA encoding in RGB-D data based pedestrian detection and then demonstrates the advantages of our proposed fusion module TFFEM.

4.1. Evaluation of Pedestrian Detection on KITTI Dataset

4.1.1. The Dataset and Evaluation Metrics

We first evaluate our pedestrian detection algorithm on the KITTI [15], a widely used and challenging dataset for autonomous driving. KITTI contains 7481 training samples and 7518 test samples. Each sample includes up to 30 pedestrians. The metrics we evaluated are consistent with the official comparision, which uses the average precision (AP) with an *IOU* threshold of 0.5 for pedestrians. The AP also contains three difficulty levels, i.e., easy, moderate and hard levels, based on the target height, occlusion and truncation.

4.1.2. Implementation Detail

In experiments, we divide all samples into a training set (3712 samples) and a validation set (3769 samples) according to [50]. The KITTI dataset does not provide depth images, so we convert the LiDAR point cloud to the camera coordinate system and generate depth images. Afterward, depth images are recovered according to [51] and encoded to HHA images. Our network uses the FasterRCNN detection head and replaces its ROIPooling with ROIAlign [52]. Except for the TFFEM, the network is initialized with pretrained weights on the CrowdHuman [53] and Citypersons [54] datasets. Furthermore, the TFFEM cannot be pretrained only on RGB data, so its weights are initialized as [55]. The network is trained for 17.6 K iterations with a batch size of 2. The SGD gradient descent algorithm is used. The initial learning rate (LR) is set to 0.01, the momentum to 0.9, and the weight decay to 0.0001. The LR increases from 0.001 to 0.01 in the beginning of 0.5 K iterations and decreases to 10% at the 15.4 K and 16.5 K iterations. The input image resolution is kept constant while the image is padded to ensure that the width and height are multiples of 32. The entire training process takes about 5.5 h on a single GTX1080Ti GPU.

When training RGB and HHA images, the data augmentation is specially designed. Color augmentation, such as illumination, transparency and gamma, is only for RGB because pixel values of the HHA image contain geometric information. Moreover, all images are randomly translated by only $-5\sim5\%$, rotated by $0\sim5$ degrees, and scale transformed by $0.9\sim1.1$ times. Furthermore, the random horizontal flip is adopted.

We also utilize some tricks to improve the detection performance. For example, the pixel value of the DontCare region on RGB and HHA images is set to 0 according to [35]. Moreover, the width of each prediction bounding box is extended by 0.1 times. It is particularly useful for AP with an IOU threshold of 0.5. Besides, many samples of the KITTI dataset do not contain pedestrians, e.g., only 955 of 3712 training samples have pedestrians. So, training all the samples would reduce the network's ability to discriminate pedestrian targets. Therefore, for each epoch, we randomly select 100 samples from the pedestrian-free samples and together with all samples containing pedestrians to training.

4.1.3. Comparison with State-of-the-Art Approaches

Table 1 shows the pedestrian detection results of our method and other state-of-the-art methods on the KITTI test dataset. The input of these methods includes single-modal and multimodal. In Table 1, D represents the depth image, P represents the LiDAR point cloud, and F represents the optical flow. It can be seen that our method outperforms other methods on moderate and hard metrics and has reached the state-of-the-art on the easy metric. In addition, although we use two-modal data, the inference speed of our network is faster than that of many single-modal methods.

We select the model that performs best on the validation set to process the test set. The results are submitted to the official benchmark (http://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=2d, (accessed on 18 December 2021)) for 2D pedestrian detection and ranked third among all published methods. Figure 9 shows the detection results in several challenging scenarios on the KITTI test set. These scenarios include insufficient light, occlusion, small-scale objects and dense crowds.

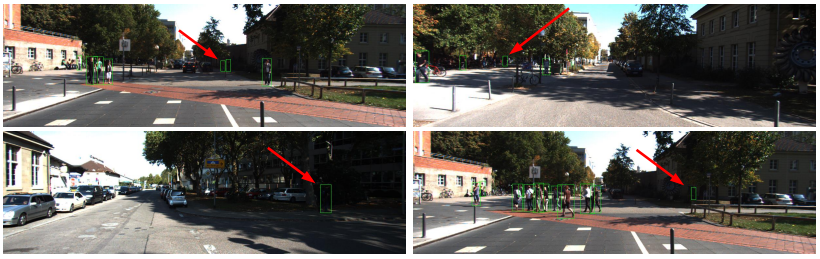
4.2. Evaluation of Pedestrian Detection on the EPFL Dataset

4.2.1. The Dataset and Evaluation Metrics

To further verify the robustness of our proposed pedestrian detection method, we also evaluate pedestrian detection performance on the EPFL dataset. This dataset employs Kinect V2 to capture RGB-D data in two scenes, i.e., a laboratory (EPFL-LAB) and a corridor (EPFL-CORRIDOR) in a university building. A total of 5140 samples were collected. EPFL-CORRIDOR is recognized as one of the most complex datasets [23]. EPFL contains various degrees of occlusion, and the distance between people is very close. We divide the EPFL dataset into a training set (1650 samples), a validation set (970 samples) and a test set (1570 samples) according to [23].

Table 1. Comparison with the state-of-the-art approaches on the KITTI test set for pedestrian detection.

Method	Easy	Moderate	Hard	Input	Time (s)
MM-MRFC (2017, [56])	83.79	70.76	64.81	RGB-D-F	0.05
SubCNN (2017, [57])	84.88	72.27	66.82	RGB	2
RRC (2017, [58])	**85.98**	76.61	71.47	RGB	3.6
ECP (2018, [59])	85.96	76.25	70.55	RGB	0.25
FRCNN+Or (2018, [60])	71.64	56.68	51.53	RGB	0.09
TAFT (2018, [36])	67.62	53.15	47.08	RGB	0.2
F-ConvNet (2019, [61])	83.63	72.91	67.18	RGB-P	0.47
VMVS (2019, [62])	82.80	71.82	66.85	RGB-P	0.25
HotSpotNet (2020, [63])	71.43	62.31	59.24	RGB	**0.04**
FII-CenterNet (2021, [64])	81.32	67.31	61.29	RGB	0.09
WSSN (2021, [5])	84.91	76.42	71.86	RGB-D	0.37
HHA-TFFEM (Proposed)	85.32	**77.12**	**72.69**	RGB-D	0.14



(a) Insufficient Light

(b) Occlusion

(c) Small Scale

(d) Dense Crowds

Figure 9. Visualization of our proposed detection result in some challenging scenarios on the KITTI test dataset.

The metrics are adopted with the AP. The EPFL data set does not distinguish the difficulty level of targets, so we only calculate the AP_{50}, i.e., the IOU threshold is 0.5. In addition, we also compute the AP_{75}, i.e., the IOU threshold is 0.75 and AP_{COCO} [65] to verify our method comprehensively. In addition, Ophoff et al. relabeled the EPFL dataset

to add the bounding box of severely obscured pedestrians that are not included in original annotations. Thus, we use their annotation information.

4.2.2. Comparison with State-of-the-Art Approaches

We compared the detection performance of our proposed pedestrian detection algorithm with nine state-of-the-art approaches with different strategies. They included four classical RGB-based target detection algorithms [24,30,31,66] and five of the newest RGB-D-based pedestrian detection algorithms [1–5]. FasterRCNN, SSD and YOLOV3 are trained and tested utilizing the algorithms provided in the MMDetection toolbox [67], and YOLOV5 utilizes the officially-offered code. The five methods of Ophoff [1], Zhang [2], Linder [3], AAFTSNet [4], and WSSN [5], have inconsistent settings for training and testing sets (e.g., Ophoff uses 70% of the data for training and 20% for testing). Therefore, we cannot directly use the test results provided in their papers. Finally, we reimplemented these methods according to their paper and recorded the test results.

Table 2 shows the detection performance of all methods on the EPFL dataset. It can be seen that our method outperforms other methods in all three metrics. In the AP_{50} metric, our method is the only one that exceeds 90%. In addition, the SSD-based method [2,31] has better performance on the AP_{50} metric but is lower on AP_{75} and AP_{COCO} metrics, while the FasterRCNN has much lower scores in AP_{50} than SSD, YOLOV3, and YOLOV5 but exceeds them in the AP_{75}. Figure 10 shows the Precision-Recall curves of AP_{50}, where the dashed line indicates the method using only RGB data.

Table 2. Comparison with the state-of-the-art approaches on the EPFL dataset for pedestrian detection.

Method	AP_{50}	AP_{75}	AP_{COCO}	Input
FasterRCNN (2015, [24])	78.1	59.1	50.2	RGB
SSD (2016, [31])	80.0	45.8	44.6	
YOLOV3 (2018, [30])	82.3	52.7	47.8	
YOLOV5 (2021, [66])	86.8	55.5	51.5	
Ophoff (2019, [1])	84.0	51.6	49.0	RGB-D
Zhang (2020, [2])	86.7	54.2	51.2	
Linder (2020, [3])	86.5	65.4	57.2	
AAFTSNet (2021, [4])	87.7	61.4	55.3	
WSSN (2021, [5])	88.4	64.1	55.8	
HHA-TFFEM (Proposed)	**90.2**	**66.0**	**57.4**	

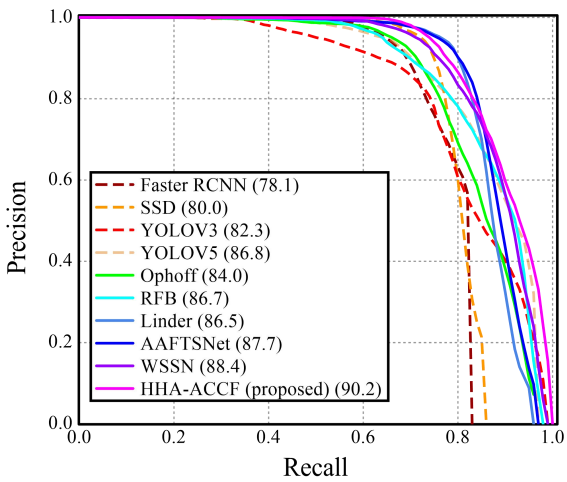


Figure 10. Precision-Recall curves of each approach for pedestrian detection on the EPFL dataset, where the dashed line indicates that the approach uses only RGB information.

4.3. Evaluation of Improved HHA Encoding

4.3.1. Datasets and Evaluation Metrics

We validated our proposed improved HHA encoding method under four commonly used RGB-D detection datasets, including UNIHALL [38,68], KTP [69,70], and two subsets of EPFL, i.e., EPFL-LAB and EPFL-COR (captured in different scenarios). To improve the validation efficiency, we selected the data captured by the first camera in UNIHALL, the first subset of EPFLCOR, and the subset of KTP named Still. Another reason for this choice is that each subset is captured with a fixed camera. Thus, the ground truth for gravity orientation is also fixed. It is worth mentioning that we did not choose the KITTI dataset for validation because the camera position of each sample on the KITTI is changed. So it is hard to obtain the ground-truth gravity orientation of all samples.

The validation metrics refer to the angle judgment proposed in [46] which judges the angle between the estimated and ground-truth gravity direction. A smaller angle indicates a better estimation. However, only the EPFL dataset provides the ground-truth gravity direction, So we manually calibrate the other two datasets to obtain the ground-truth gravity direction. The manual estimation process is first to select a depth image with the largest ground area. Then five ground points are manually marked and fitted by the least-squares algorithm to obtain the gravity direction and calculate the fitting error. Lastly, this process is repeated five times and selects the parameters with the smallest error as the ground truth.

4.3.2. Comparison of Gravity Direction Estimation

Figure 11 shows the angle of gravity direction for our improved (blue) and the original (red) methods on four datasets. In each subplot, the estimated angles of all images are listed, where the horizontal coordinates indicate the image number and the vertical coordinates indicate the angle.

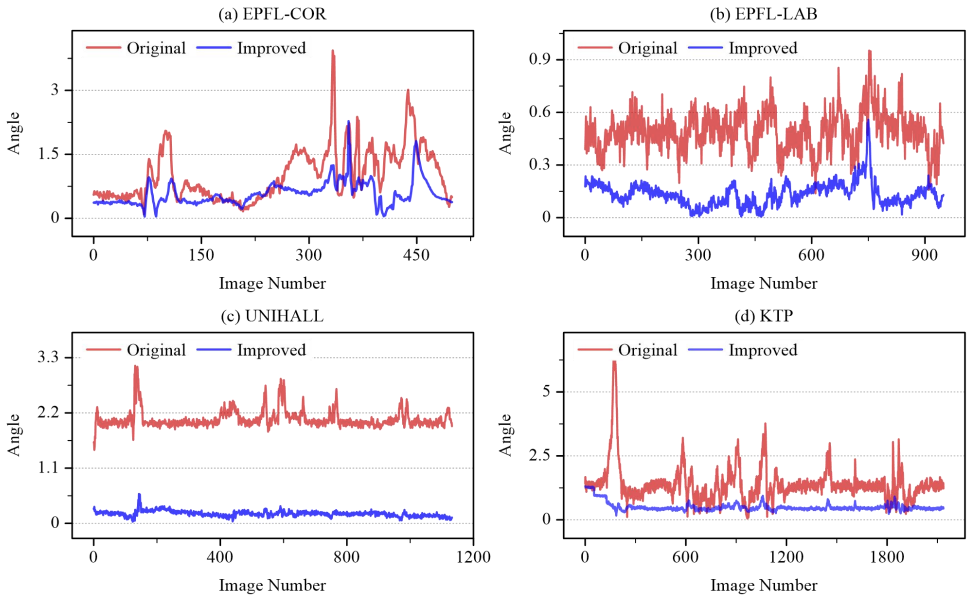


Figure 11. Comparison of the gravity direction estimation accuracy between our improved and original methods on four RGB-D datasets.

As can be seen, the overall angle of our improved method is smaller than the original method, especially on the UNIHALL. The scene of UNIHALL contains a staircase whose points are miss-classified to the vertical set, thus seriously affecting the gravity estimation of the original method. In contrast, the improved method is less affected because many irrelevant points are removed at each iteration.

In addition, both methods estimate larger angles on the UNIHALL and KTP than on the EPFL. This is because these two datasets are captured by Kinect V1, and their depth information quality is lower than that of the EPFL captured with Kinect V2.

4.3.3. Comparison of Encoding Speed

Since the improved HHA encoding method reduces the number of iterations from ten to three and removes a part of the points in each iteration when estimating the gravity direction, the running speed is significantly promoted. Table 3 shows the HHA encoding times of our improved and original methods on five datasets. The time for each dataset is averaged over all images, and the unit is in seconds. As can be seen, our improved method is more than twice as fast as the original method. The statistics experiment uses an ordinary desktop computer with Intel Core I7-409 8700K, 16 GB of RAM and Win10. The software used is the MatLab platform, which is consistent with the official HHA encoding code. It is worth mentioning that this time it will run faster on platforms such as C++.

Table 3. HHA encoding times of our improved and original methods on five datasets.

		KTP [70]	UNIHALL [38]	EPFL-LAB [16]	EPFL-COR [16]	KITTI [15]	Avg.
Runtimes (s)	Original	0.630	0.618	0.438	0.433	1.102	0.644
	Improved	0.267	0.263	0.182	0.172	0.471	0.271
Image Resolution		640 × 480	640 × 480	512 × 424	512 × 424	1242 × 375	

4.4. Ablation Study

In this section, we execute comparative experiments to demonstrate the effectiveness of these two parts.

4.4.1. Study on Different Depth Encoding Methods

Table 4 shows the detection performance of four types of depth information encoded images and our improved HHA images on the KITTI val dataset with RGB-D as input. These encoding methods include grayscale, surface normals, jet colormap and the original HHA. There are two main aspects to our improvement of HHA encoding. One is more efficient ground parammeter extraction (GE) and the other is DEM. Therefore, Table 4 lists two of our detection results, which are HHA + GE, i.e., only the effect after the GE is employed, and HHA + GE + DEM, when the two improvements are contained. Moreover, except for the original metrics provided in the KITTI dataset, we calculated the AP_{50} and AP_{COCO} results for a more comprehensive comparison. As can be seen, our encoding method outperforms other encoding methods in all metrics.

Table 4. Comparison of depth encoding methods on the KITTI val dataset for RGB-D pedestrian detection.

Depth Encoding Method	Easy	Moderate	Hard	AP_{50}	AP_{COCO}	Input
Grayscale	85.78	78.41	71.39	66.9	31.7	RGB-D
Surface Normals	87.86	79.42	72.53	69.7	34.4	
Colormap Jet	87.58	79.72	73.05	68.5	33.4	
HHA Orginal	87.26	80.25	72.96	68.9	32.9	
HHA + GE (Proposed)	87.44	80.84	74.00	70.6	33.5	
HHA + GE + DEM (Proposed)	**88.90**	**82.14**	**75.33**	**71.5**	**34.5**	

4.4.2. Study on Different Fusion Methods

We first compared our multimodal fusion module with common fusion methods on the KITTI val dataset as shown in Table 5. The compared methods include element summation and concatenation. Since concatenation increases the channel number, we reduced the channel number using 1 × 1 convolution after concatenation. We also compare

the detection results after using the attention module CBAM for each fusion method. From Table 5, we can see that the detection results of our fusion method are better than the other approaches in all metrics, both in direct fusion and followed by the attention module.

Table 5. Comparison of fusion methods on the KITTI val dataset for RGB-D pedestrian detection.

Fusion Method	Easy	Moderate	Hard	AP_{50}	AP_{COCO}	Input
Summation	86.76	79.02	72.83	68.4	31.6	RGB-D
Concatenation	87.79	80.33	73.19	69.5	33.7	
TFFEM (proposed)	**88.00**	**80.76**	**74.58**	**70.1**	**34.0**	
Summation + CBAM	87.42	79.86	73.83	70.3	33.9	
Concatenation + CBAM	87.74	80.62	73.74	71.1	33.4	
TFFEM + CBAM (proposed)	**88.90**	**82.14**	**75.33**	**71.5**	**34.5**	

5. Discussion

The above experiments fully demonstrate the effectiveness of the proposed method. In this section, the proposed method will be further analyzed.

(1) The improved HHA encoding method obtains more accurate gravity directions, thus improving the precision of the encoded height information. At the same time, height information is a very discriminated but noise-sensitive feature to the pedestrian detection task compared to other tasks such as scene understanding [9] and salient object detection [10]. Therefore, the HHA images obtained by the original encoding method perform poorly in the pedestrian detection task but well in other tasks [11,12].

(2) The result in Table 4 shows that using the DEM is better than mean mapping. DEM allows depth regions containing more targets to be mapped to a wider color space, which increases the dissimilarity between targets and thus can improve detection performance.

(3) The proposed TFFEM extracts local and global features of two modalities, which increases the richness of the network features and thus improves the pedestrian detection performance. In addition, the attention model is also beneficial to RGB-D pedestrian detection.

(4) The improved HHA encoding method is twice as fast as the original method but still does not run in real-time. In addition, the complexity of the proposed RGB-D pedestrian detection network is high. Therefore, our subsequent work will involve improving the detection speed to satisfy the application requirements.

6. Conclusions

In this paper, we address two key issues in the RGB-D pedestrian detection task, i.e., utilizing depth information and fusing the two modalities' information. Firstly, we analyze the reasons for the low detection performance of HHA encoded images carrying more information and improving the HHA encoding. Secondly, we propose the TFFEM and design a new RGB-D detection network based on this module.

Through the experiments, we first validate the effectiveness of the proposed RGB-D detection network on two datasets, followed by the advantages of our improved HHA encoding on four datasets. Finally, the superiority of the proposed improved HHA encoding over other encoding methods and the excellence of the proposed TFFEM are demonstrated.

Author Contributions: Methodology, F.T. and Z.X.; software, F.T. and Y.M.; validation, F.T.; investigation, X.F.; writing—original draft preparation, F.T.; writing—review and editing, Z.X.; visualization, F.T.; supervision, X.F.; project administration, X.F. and Z.X.; funding acquisition, X.F. and Z.X. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partly supported by the Key Research and Development Program of Shaanxi (Program Nos. 2021ZDLGY15-01, 2021ZDLGY09-04, 2021GY-004 and 2020GY-050), and the International Science and Technology Cooperation Research Project of Shenzhen (GJHZ20200731095204013).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available in [15,16,38,69].

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

1. Ophoff, T.; Van Beeck, K.; Goedemé, T. Exploring RGB+ Depth fusion for real-time object detection. *Sensors* **2019**, *19*, 866. [CrossRef]
2. Zhang, W.; Wang, J.; Guo, X.; Chen, K.; Wang, N. Two-Stream RGB-D Human Detection Algorithm Based on RFB Network. *IEEE Access* **2020**, *8*, 123175–123181. [CrossRef]
3. Linder, T.; Pfeiffer, K.Y.; Vaskevicius, N.; Schirmer, R.; Arras, K.O. Accurate detection and 3D localization of humans using a novel YOLO-based RGB-D fusion approach and synthetic training data. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1000–1006.
4. Zhang, W.; Guo, X.; Wang, J.; Wang, N.; Chen, K. Asymmetric Adaptive Fusion in a Two-Stream Network for RGB-D Human Detection. *Sensors* **2021**, *21*, 916. [CrossRef] [PubMed]
5. Guo, Z.; Liao, W.; Xiao, Y.; Veelaert, P.; Philips, W. Weak Segmentation Supervised Deep Neural Networks for Pedestrian Detection. *Pattern Recognit.* **2021**, *119*, 108063. [CrossRef]
6. Nebiker, S.; Meyer, J.; Blaser, S.; Ammann, M.; Rhyner, S. Outdoor Mobile Mapping and AI-Based 3D Object Detection with Low-Cost RGB-D Cameras: The Use Case of On-Street Parking Statistics. *Remote Sens.* **2021**, *13*, 3099. [CrossRef]
7. Gupta, S.; Girshick, R.; Arbeláez, P.; Malik, J. Learning rich features from RGB-D images for object detection and segmentation. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 345–360.
8. Cao, J.; Leng, H.; Lischinski, D.; Cohen-Or, D.; Tu, C.; Li, Y. ShapeConv: Shape-aware Convolutional Layer for Indoor RGB-D Semantic Segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Nashville, TN, USA, 20–25 June 2021; pp. 7088–7097.
9. Ferreri, A.; Bucci, S.; Tommasi, T. Multi-Modal RGB-D Scene Recognition across Domains. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Nashville, TN, USA, 20–25 June 2021; pp. 2199–2208.
10. Huang, Z.; Chen, H.X.; Zhou, T.; Yang, Y.Z.; Liu, B.Y. Multi-level cross-modal interaction network for RGB-D salient object detection. *Neurocomputing* **2021**, *452*, 200–211. [CrossRef]
11. Eitel, A.; Springenberg, J.T.; Spinello, L.; Riedmiller, M.; Burgard, W. Multimodal deep learning for robust RGB-D object recognition. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–3 October 2015; pp. 681–687. [CrossRef]
12. Ren, X.; Du, S.; Zheng, Y. Parallel RCNN: A deep learning method for people detection using RGB-D images. In Proceedings of the 2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Shanghai, China, 14–16 October 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–6.
13. Zhang, Q.; Xiao, T.; Huang, N.; Zhang, D.; Han, J. Revisiting feature fusion for rgb-t salient object detection. *IEEE Trans. Circuits Syst. Video Technol.* **2020**, *31*, 1804–1818. [CrossRef]
14. Zhang, Q.; Zhao, S.; Luo, Y.; Zhang, D.; Huang, N.; Han, J. ABMDRNet: Adaptive-weighted Bi-directional Modality Difference Reduction Network for RGB-T Semantic Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 2633–2642.
15. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012.
16. Bagautdinov, T.; Fleuret, F.; Fua, P. Probability occupancy maps for occluded depth images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 2829–2837.
17. Hu, T.; Zhang, H.; Zhu, X.; Clunis, J.; Yang, G. Depth sensor based human detection for indoor surveillance. *Future Gener. Comput. Syst.* **2018**, *88*, 540–551. [CrossRef]
18. Luna, C.A.; Losada-Gutiérrez, C.; Fuentes-Jiménez, D.; Mazo, M. Fast heuristic method to detect people in frontal depth images. *Expert Syst. Appl.* **2021**, *168*, 114483. [CrossRef]
19. Tian, L.; Li, M.; Hao, Y.; Liu, J.; Zhang, G.; Chen, Y.Q. Robust 3-d human detection in complex environments with a depth camera. *IEEE Trans. Multimed.* **2018**, *20*, 2249–2261. [CrossRef]
20. Xia, L.; Chen, C.C.; Aggarwal, J.K. Human detection using depth information by kinect. In Proceedings of the CVPR 2011 Workshops, Colorado Springs, CO, USA, 20–25 June 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 15–22.
21. Hacinecipoglu, A.; Konukseven, E.I.; Koku, A.B. Fast head detection in arbitrary poses using depth information. *Sens. Rev.* **2020**, *40*, 175–182. . [CrossRef]
22. Fuentes-Jimenez, D.; Martin-Lopez, R.; Losada-Gutierrez, C.; Casillas-Perez, D.; Macias-Guarasa, J.; Luna, C.A.; Pizarro, D. DPDnet: A robust people detector using deep learning with an overhead depth camera. *Expert Syst. Appl.* **2020**, *146*, 113168. [CrossRef]
23. Fuentes-Jimenez, D.; Losada-Gutierrez, C.; Casillas-Perez, D.; Macias-Guarasa, J.; Pizarro, D.; Martin-Lopez, R.; Luna, C.A. Towards dense people detection with deep learning and depth images. *Eng. Appl. Artif. Intell.* **2021**, *106*, 104484. [CrossRef]

24. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [CrossRef] [PubMed]
25. Xie, J.; Pang, Y.; Cholakkal, H.; Anwer, R.; Khan, F.; Shao, L. PSC-net: Learning part spatial co-occurrence for occluded pedestrian detection. *Sci. China Inf. Sci.* **2021**, *64*, 1–13. [CrossRef]
26. Wang, C.C.R.; Lien, J.J.J. AdaBoost learning for human detection based on histograms of oriented gradients. In Proceedings of the Asian Conference on Computer Vision, Venice, Italy, 22–29 October 2007; Springer: Berlin/Heidelberg, Germany, 2007; pp. 885–895.
27. Mu, Y.; Yan, S.; Liu, Y.; Huang, T.; Zhou, B. Discriminative local binary patterns for human detection in personal album. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, Alaska, 23–28 June 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 1–8.
28. Huang, S.S.; Tsai, H.M.; Hsiao, P.Y.; Tu, M.Q.; Jian, E.L. Combining histograms of oriented gradients with global feature for human detection. In Proceedings of the International Conference on Multimedia Modeling, Taipei, Taiwan, 5–7 January 2011; Springer: Berlin/Heidelberg, Germany, 2011; pp. 208–218.
29. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
30. Farhadi, A.; Redmon, J. Yolov3: An incremental improvement. In Proceedings of the Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 1804–2767.
31. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
32. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 10781–10790.
33. Zhou, X.; Wang, D.; Krähenbühl, P. Objects as points. *arXiv* **2019**, arXiv:1904.07850.
34. Tian, Z.; Shen, C.; Chen, H.; He, T. Fcos: Fully convolutional one-stage object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 9627–9636.
35. Liu, W.; Liao, S.; Ren, W.; Hu, W.; Yu, Y. High-level semantic feature detection: A new perspective for pedestrian detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5187–5196.
36. Shen, J.; Zuo, X.; Yang, W.; Prokhorov, D.; Mei, X.; Ling, H. Differential features for pedestrian detection: A Taylor series perspective. *IEEE Trans. Intell. Transp. Syst.* **2018**, *20*, 2913–2922. [CrossRef]
37. Luo, Y.; Zhang, C.; Zhao, M.; Zhou, H.; Sun, J. Where, What, Whether: Multi-modal learning meets pedestrian detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 14065–14073.
38. Spinello, L.; Arras, K.O. People detection in RGB-D data. In Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, 25–30 September 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 3838–3843.
39. Mees, O.; Eitel, A.; Burgard, W. Choosing smartly: Adaptive multimodal fusion for object detection in changing environments. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 151–156.
40. Zhou, K.; Paiement, A.; Mirmehdi, M. Detecting humans in RGB-D data with CNNs. In Proceedings of the 2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA), Nagoya, Japan, 8–12 May 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 306–309.
41. Chen, X.; Kundu, K.; Zhu, Y.; Ma, H.; Fidler, S.; Urtasun, R. 3d object proposals using stereo imagery for accurate object class detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 1259–1272. [CrossRef]
42. Ophoff, T.; Van Beeck, K.; Goedemé, T. Improving Real-Time Pedestrian Detectors with RGB+ Depth Fusion. In Proceedings of the 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Auckland, New Zealand, 27–30 November 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–6.
43. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.
44. Kollmitz, M.; Eitel, A.; Vasquez, A.; Burgard, W. Deep 3D perception of people and their mobility aids. *Robot. Auton. Syst.* **2019**, *114*, 29–40. [CrossRef]
45. Seichter, D.; Lewandowski, B.; Höchemer, D.; Wengefeld, T.; Gross, H.M. Multi-task deep learning for depth-based person perception in mobile robotics. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 25–29 October 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 10497–10504.
46. Gupta, S.; Arbelaez, P.; Malik, J. Perceptual organization and recognition of indoor scenes from RGB-D images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 564–571.
47. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
48. Guan, D.; Cao, Y.; Yang, J.; Cao, Y.; Yang, M.Y. Fusion of multispectral data through illumination-aware deep neural networks for pedestrian detection. *Inf. Fusion* **2019**, *50*, 148–157. [CrossRef]

49. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany,8–14 September 2018; pp. 3–19.
50. Yan, Y.; Mao, Y.; Li, B. Second: Sparsely embedded convolutional detection. *Sensors* **2018**, *18*, 3337. [CrossRef] [PubMed]
51. Park, J.; Joo, K.; Hu, Z.; Liu, C.K.; Kweon, I.S. Non-Local Spatial Propagation Network for Depth Completion. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020.
52. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
53. Shao, S.; Zhao, Z.; Li, B.; Xiao, T.; Yu, G.; Zhang, X.; Sun, J. Crowdhuman: A benchmark for detecting human in a crowd. *arXiv* **2018**, arXiv:1805.00123.
54. Zhang, S.; Benenson, R.; Schiele, B. Citypersons: A diverse dataset for pedestrian detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3213–3221.
55. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1026–1034.
56. Daniel Costea, A.; Varga, R.; Nedevschi, S. Fast boosting based detection using scale invariant multimodal multiresolution filtered features. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6674–6683.
57. Xiang, Y.; Choi, W.; Lin, Y.; Savarese, S. Subcategory-aware convolutional neural networks for object proposals and detection. In Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, USA, 24–31 March 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 924–933.
58. Ren, J.; Chen, X.; Liu, J.; Sun, W.; Pang, J.; Yan, Q.; Tai, Y.W.; Xu, L. Accurate single stage detector using recurrent rolling convolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5420–5428.
59. Braun, M.; Krebs, S.; Flohr, F.; Gavrila, D.M. The eurocity persons dataset: A novel benchmark for object detection. *arXiv* **2018**, arXiv:1805.07193.
60. Guindel, C.; Martin, D.; Armingol, J.M. Fast joint object detection and viewpoint estimation for traffic scene understanding. *IEEE Intell. Transp. Syst. Mag.* **2018**, *10*, 74–86. [CrossRef]
61. Wang, Z.; Jia, K. Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal 3d object detection. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1742–1749.
62. Ku, J.; Pon, A.D.; Walsh, S.; Waslander, S.L. Improving 3d object detection for pedestrians with virtual multi-view synthesis orientation estimation. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 3459–3466.
63. Chen, Q.; Sun, L.; Wang, Z.; Jia, K.; Yuille, A. object as hotspots. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020.
64. Fan, S.; Zhu, F.; Chen, S.; Zhang, H.; Tian, B.; Lv, Y.; Wang, F.Y. FII-CenterNet: An Anchor-Free Detector With Foreground Attention for Traffic Object Detection. *IEEE Trans. Veh. Technol.* **2021**, *70*, 121–132. [CrossRef]
65. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755.
66. Jocher, G.; Kwon, Y.; guigarfr; perry0418; Veitch-Michaelis, J.; Ttayu; Suess, D.; Baltacı, F.; Bianconi, G.; IlyaOvodov; et al. Ultralytics/yolov3: v9.5.0—YOLOv5 v5.0 Release Compatibility Update for YOLOv3.2021. Available online: https://zenodo.org/record/4681234#.YfP42OrMKUk (accessed on 18 December 2021). [CrossRef]
67. Chen, K.; Wang, J.; Pang, J.; Cao, Y.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Xu, J.; et al. MMDetection: Open MMLab Detection Toolbox and Benchmark. *arXiv* **2019**, arXiv:1906.07155.
68. Luber, M.; Spinello, L.; Arras, K.O. People tracking in rgb-d data with on-line boosted target models. In Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, 25–30 September 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 3844–3849.
69. Munaro, M.; Basso, F.; Menegatti, E. Tracking people within groups with RGB-D data. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, Portugal, 7–12 October 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 2101–2107.
70. Munaro, M.; Menegatti, E. Fast RGB-D people tracking for service robots. *Auton. Robot.* **2014**, *37*, 227–242. [CrossRef]