machines

MDPI

Article

Monocular Depth and Velocity Estimation Based on Multi-Cue Fusion

Chunyang Qi [1], Hongxiang Zhao [1], Chuanxue Song [2], Naifu Zhang [3], Sinxin Song [3], Haigang Xu [4] and Feng Xiao [1,*]

[1] State Key Laboratory of Automotive Simulation and Control, Jilin University, Changchun 130022, China; qicy20@mails.jlu.edu.cn (C.Q.); zhaohx19@mails.jlu.edu.cn (H.Z.)
[2] College of Automotive Engineering, Jilin University, Changchun 130022, China; scx@jlu.edu.cn
[3] School of Mechanical and Aerospace Engineering, Jilin University, Changchun 130022, China; zhangnf21@mails.jlu.edu.cn (N.Z.); ssx@jlu.edu.cn (S.S.)
[4] Shandong Shifeng (Group), Liaocheng 252000, China; tuchuanwen2@163.com
* Correspondence: xiaofengjl@jlu.edu.cn; Tel.: +86-178-0808-1932

Abstract: Many consumers and scholars currently focus on driving assistance systems (DAS) and intelligent transportation technologies. The distance and speed measurement technology of the vehicle ahead is an important part of the DAS. Existing vehicle distance and speed estimation algorithms based on monocular cameras still have limitations, such as ignoring the relationship between the underlying features of vehicle speed and distance. A multi-cue fusion monocular velocity and ranging framework is proposed to improve the accuracy of monocular ranging and velocity measurement. We use the attention mechanism to fuse different feature information. The training method is used to jointly train the network through the distance velocity regression loss function and the depth loss as an auxiliary loss function. Finally, experimental validation is performed on the Tusimple dataset and the KITTI dataset. On the Tusimple dataset, the average speed mean square error of the proposed method is less than 0.496 m^2/s^2, and the average mean square error of the distance is 5.695 m^2. On the KITTI dataset, the average velocity mean square error of our method is less than 0.40 m^2/s^2. In addition, we test in different scenarios and confirm the effectiveness of the network.

Keywords: monocular depth estimation; driver assistance systems; computer vision; attention mechanisms

Citation: Qi, C.; Zhao, H.; Song, C.; Zhang, N.; Song, S.; Xu, H.; Xiao, F. Monocular Depth and Velocity Estimation Based on Multi-Cue Fusion. *Machines* **2022**, *10*, 396. https://doi.org/10.3390/machines10050396

Academic Editors: Antonios Gasteratos and Ioannis Kostavelis

Received: 20 April 2022
Accepted: 16 May 2022
Published: 19 May 2022

1. Introduction

With the rapid economic growth, the global vehicle ownership increases rapidly, leading to more serious traffic safety problems. The application of advanced driver assistance systems allows the driver to be aware of possible hazards in advance, effectively increasing the comfort and safety of vehicle driving. Accurate calculation of the distance and speed between vehicles is a basic requirement for driver assistance systems.

Scene depth velocity information is a very important role in many contemporary topics and there are many typical algorithms in current research: single-radar sensor, camera sensor, stereo image, wireless sensing, multi-sensing fusion, etc.

Radar can achieve speed and range measurement of target vehicles, but it detects obstacles by transmitting optical fibers, and light reflection can also cause misjudgment in harsh environments, especially rain, snow, and water mist on foggy days [1]. In addition, the refresh rate of LIDAR is low, and it is difficult to perceive objects ahead quickly in a single scene at high speed. The camera sensor is another key part of a typical sensor configuration that can be used in normal rain and snow conditions, as it can obtain high-pixel environmental information as well as fine-texture structure information. Therefore, many researchers have started with a monocular sensor to explore the depth estimation algorithm.

In wireless sensors, many scholars have also conducted research. Ciccarese et al. [2] combined the potential of antenna array processing with cooperative strategies using vehicle-to-vehicle and vehicle-to-infrastructure communications and defined a novel algorithm with asynchronous updates triggered by beacon packet reception and capable of reaching the angle estimation goal. This algorithm achieves high localization accuracy even in sparse scenarios, outperforming competitors, while maintaining lightweight communication and low computational complexity. Shin et al. proposed [3] a prediction algorithm of vehicle speed based on a stochastic model using a Markov chain with speed constraints. The Markov chain generates the velocity trajectory stochastically within speed constraints. The constraints are estimated by an empirical model that takes into account the road geometry and is organized by the intuitive form of matrices. The experimental results show a root mean square error of 3.8041 km/h over a prediction range of up to 200 m. Stereo images take up too many computational resources compared to image sensors, even though the accuracy rate is improved.

Meanwhile, deep learning has recently achieved great success in many vision applications, such as object detection [4,5], optical flow estimation [6,7], and depth estimation [8,9]. Considering economic efficiency and environmental adaptability, many researchers have started with monocular cameras and explored their estimates of distance and speed using deep learning methods. In addition, some researchers explore the connection between the optical stream information and the monocular depth [10]. Ma [11] and Christoph [12] achieved impressive results by adding stereo video sequences to the optical flow algorithm. However, the use of stereoscopic video can lead to very high computational costs. Therefore, the use of monocular cameras with different cues has been proposed to estimate the speed and distance of vehicles in a real-time scene. For example, surveillance cameras can be used to analyze traffic flow and vehicle speed in real-time by fixing camera settings and road constraints [13]. However, the estimation becomes unstable in dynamic scenes. Thus, the actual depth of the vehicle as well as the speed mapped to reality through image time cues is difficult to obtain. Until the current study, studies on monocular velocity estimation using multiple complex cues, including vehicle target tracking, dense depth information, and optical flow information to regress the relative velocity of other vehicles, were few [14]. In the present study, we design the network systematically by combining the distance regression model and optical flow estimation network. To enable the network to accurately focus on the traffic prediction for each vehicle, a vehicle-centric vehicle bounding box extraction module is used to reduce the unbalanced motion caused between the stationary background and the moving vehicles. We focus more on the intrinsic connection between geometric cues and deep features.

The main contributions of this study are as follows:

1. The inter-vehicle distance and relative speed estimation network is systematically designed.
2. The intrinsic connection between geometric cues and deep features is investigated.
3. Geometric features are expanded and incorporated into the attention mechanism.
4. The results show that the speed and distance measurement results are significantly improved.

The remainder of this paper is structured as follows: Section 2 introduces the related work, Section 3 introduces the multi-cue fusion method and explores the relationship between deep features and the network, Section 4 introduces the proposed algorithm on two datasets, Tusimple and KITTI, and Section 5 presents the conclusion and future work.

2. Related Work

Traditional depth estimation uses binocular images for matching [15], but this method suffers from a slow computation speed and low accuracy. Deep neural network has become one of the most widely used depth estimation techniques. Generally, it can be roughly divided into the following categories: learning-based stereo-matching, supervised monocular depth estimation [16], and unsupervised monocular depth estimation. In Table 1,

we list the model structure and main contributions of some typical algorithms. Although they contribute significantly to the monocular depth velocity algorithm, they neglect the underlying vehicle geometry features.

Table 1. Algorithm model comparison.

Literature	Model Structure	Main Contribution
Eigen et al. [17]	CNN	Used deep learning models for the first time
Lee et al. [18]	CNN	Optimizing the frequency domain
Li et al. [19]	CNN	Used gradient information for optimization
Laina et al. [20]	FCN	Proposed a new sampling module
Hu et al. [21]	FCN	Used multiscale information to improve
Liu et al. [22]	CNN	Random field step-by-step optimization
Xu et al. [23]	FCN	Optimized with continuity condition

In traditional methods, the Markov random field model [24,25] is usually used to predict the monocular depth. Saxena et al. [26] trained, in a supervised manner, to model the relationship between the depth features of the image and the image target to predict the image depth from monocular images. Karsch et al. [27] proposed the use of nonparametric depth to estimate the depth of monocular images and videos, and it can also realize the transformation from stereo images to 3D images. Meanwhile, the structure of motion (SFM) algorithm [28–30] was commonly used to estimate the depth information of objects in monocular images.

In 2014, Eigen et al. [17] used two deep convolutional neural networks to estimate the depth of monocular images. Subsequently, Eigen et al. [31] used a multiscale approach to obtain the pixel set features of the image for depth prediction, which can improve the accuracy of the network. In addition, Atapour et al. [32] used a joint training of pixel-level semantic information and depth information to estimate the depth of objects in the scene. Moukari et al. [33] studied four different depth networks, where the depth map can be obtained using multiscale features in the network. Qi et al. [34] applied the uncertainty method to monocular depth. Zhe et al. [35] applied 3D detection to monocular depth estimation to achieve distance recovery.

Supervised monocular depth estimation requires the use of a large amount of manually labeled data to train the model, leading to a high cost of true depth acquisition. In 2016, Garge et al. [36] proposed an unsupervised framework based on deep convolutional neural networks using stereo image pairs for training, without pre-training. Godard et al. [37] proposed a consistency loss for left- and right-image parallax using polar line geometric constraints on binocular images to improve the accuracy and robustness of monocular depth estimation. Zhou et al. [38] established a visual correspondence between different instances and used the inter-instance consistency relationships as supervised signals to train convolutional neural networks. Subsequently, Zhou et al. [39] addressed the problem of new view synthesis by synthesizing the same scene obtained from any viewpoint to obtain a new image based on the highly correlated appearance of the same instance in different views. Inspired by these approaches, an unsupervised learning framework based on binocular images and image reconstruction loss [40] is widely used in monocular depth estimation.

Relevant studies on monocular velocity estimation algorithms are relatively few. Most of them rely on the distance information of the target ahead and then estimate the velocity by the rate of change of the distance. However, the existence of distance errors causes the superposition of speed estimation errors, thereby obtaining inaccurate speed information. To obtain the relative velocity between the self-vehicle and the vehicle in front, Christoph et al. [12] regressed the velocity of the vehicle directly from monocular sequences that exploited several cues, such as the motion features of Flownet [41] and the depth features of Monodepth. In addition, the authors of [42] used geometric constraints and optical flow features to jointly predict the velocity and distance of the vehicle. Although

these works achieved the expected performance, they predicted the state of each vehicle separately and neglected to explore the relationship between neighboring vehicles. The authors of [43] proposed a global relative constraint loss that requires the states between vehicles to reduce the error.

These research results show that the use of monocular cameras for distance and speed measurement work still has many unresolved problems. The two main factors are as follows: one is the difficulty of obtaining distance through monocular cameras, and the other is that the current ranging algorithms are imperfect, resulting in less accurate monocular ranging and speed measurement than expected. To solve these problems, a multi-clue fusion distance and speed model is proposed to estimate the distance and speed of the vehicle ahead.

3. Method

The coordinate system of the camera is defined as follows: the z-axis is forward along the optical axis of the camera, the x-axis is parallel to the image to the right, and the y-axis is parallel to the bottom of the image. The specific perspective view is shown in Figure 1.

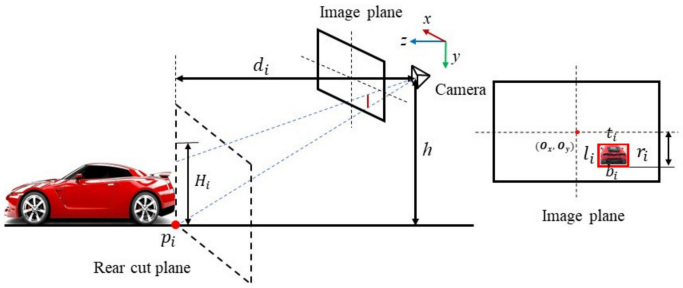

Figure 1. Perspective projection of the vehicle. In the above picture, p is the vehicle's pickup point, d_i denotes the distance to the previous vehicle, h denotes the camera height and l_i, t_i, r_i, b_i represent the size of the vehicle frame.

$$b_i = (l_i, t_i, r_i, b_i) \in \mathbb{R}^4, \tag{1}$$

The cropped vehicle target bounding box, $\{b_i \mid i = 1, \ldots, n\}$, is used as the input of the ranging and speed measurement network, and each bounding box consists of four image coordinates: left, top, right, and bottom. The overall algorithm flowchart is shown in Figure 2.

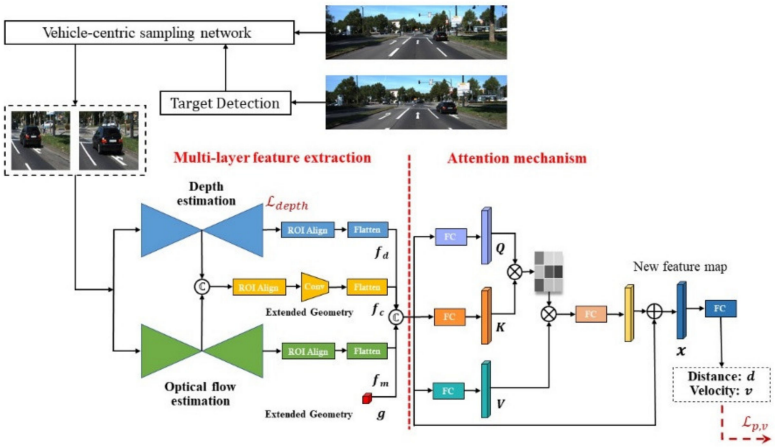

Figure 2. Algorithm flowchart. The target detection part is used to extract the vehicle detection frame in the video and input it into the speed measurement and ranging network. The algorithm used for target detection is the general Yolo3 [44] algorithm.

3.1. Geometric Cues and Odometry Models

Many current ranging algorithms rely on additional information, such as vanishing points and lane lines, but vanishing points and lane lines are susceptible to the influence of road quality and surrounding references. Therefore, we explore the relationship between geometric cues and ranging models to estimate the distance to the vehicle ahead.

According to the pinhole camera model, the distance of the vehicle ahead can be solved in two ways: one is solved assuming that the height or width of the vehicle is known, and the other is solved based on the vehicle's grounding point.

The distance based on the vehicle height and width is shown in Equation (2):

$$d_{w_i} = \frac{f_y \times H_i}{b_i - t_i} = \frac{f_x \times W_i}{r_i - l_i}, \tag{2}$$

and the distance based on the vehicle pickup point is shown in Equation (3):

$$d_{p_i} = \frac{f_y \times H_c}{b_i - o_y}, \tag{3}$$

where f_x and f_y indicate the focal length of the camera, W_i and H_i are the actual vehicle height and width, respectively, $l_i, r_i, t_i,$ and b_i are the coordinates of the left, right, top, and bottom of the vehicle bounding box, respectively, and o_y is the coordinate of the camera optical axis in the y-axis direction under the camera plane.

Each approach has its limitations. The distance based on the vehicle height and width requires the actual width of the vehicle. In addition, the other approach needs to assume that the road surface is always level.

$$d_i = \alpha \cdot \frac{f_y \times H_i}{b_i - t_i} + \beta \cdot \frac{f_x \times W_i}{r_i - l_i} + \gamma \cdot \frac{f_y \times H_c}{b_i - o_y}, \tag{4}$$

Therefore, the two ranging algorithms can be fused and used to improve the accuracy and stability of vehicle distance estimation in Equation (4), where $\frac{f_y}{b_i - t_i}$, $\frac{f_x}{r_i - l_i}$, and $\frac{f \times H_c}{b_i - o_y}$ are obtained directly from the geometric features through the camera intrinsic parameters, bounding box parameters, and camera height, respectively. H_i and W_i are the actual height and width information of the vehicle, respectively, depending only on the characteristics of the vehicle. These vehicle features can be learned by a large number of training samples. Therefore, to learn these parameters and features, they are extracted using a deep neural network, and the distance geometric feature vector obtained is represented by g_d. α, β, and γ can be used to measure the confidence level of each partial distance estimation. g_d is shown in Equation (5):

$$g_d = \left[\frac{f_y}{b_i - t_i}, \frac{f_x}{r_i - l_i}, \frac{f_y \, H_c}{b_i - o_y} \right]. \tag{5}$$

In summary, the specific method for the forward vehicle distance, d_i, estimation is as follows: The depth features and other different sizes are unified to the same size by ROI align, and then spread to obtain the depth feature vector, f_d. Finally, the vehicle deep feature vector, $f_{c,d}$, together with the geometric cue, g_i, form the depth estimation network.

Thus, the model for distance regression can be expressed as follows:

$$f_d = Flatten(RoI(depth(I_{t-1}, I_t))), \tag{6}$$

$$f_{c,d} = Flatten(RoI(feature(I_{t-1}, I_t))), \tag{7}$$

$$d_i = FC_d(f_d, f_{c,d}, g_d), \tag{8}$$

where $depth$ is the depth network, $feature$ is the feature extraction network, RoI is the ROI align module, $Flatten$ is the spreading operation, f_d is the depth feature vector, $f_{c,d}$

is the deep feature vector, g_d is the distance geometry feature vector, and FC_d is the fully connected layer.

The loss function, $\mathcal{L}_{depth}$, of the depth network is as follows:

$$\mathcal{L}_{depth} = \alpha_{ap}\left(\mathcal{L}_{ap}^{t-1} + \mathcal{L}_{ap}^{t}\right) + \alpha_{ds}\left(\mathcal{L}_{ds}^{t-1} + \mathcal{L}_{ds}^{t}\right) + \alpha_{lr}\left(\mathcal{L}_{lr}^{t-1} + \mathcal{L}_{lr}^{t}\right), \tag{9}$$

$$\mathcal{L}_{ap} = \frac{1}{N}\sum_{i,j}\alpha\frac{1 - SSIM\left(I_{ij}^{t-1}, \widetilde{I}_{ij}^{t}\right)}{2} + (1-\alpha)\,||I_{ij}^{t-1}, -, \widetilde{I}_{ij}^{t-1}\,||, \tag{10}$$

$$\mathcal{L}_{ds} = \frac{1}{N}\sum_{i,j}\left|\partial_x d_{i,j}^{t-1}\right|e^{-||\partial_x, I_{i,j}^{t-1}||} + \left|\partial_y d_{i,j}^{t-1}\right|e^{-||\partial_y, I_{i,j}^{t-1}||}, \tag{11}$$

$$L_{lr} = \frac{1}{N}\sum_{i,j}\left|d_{ij}^{t-1} - d_{ij}^{t}\right|, \tag{12}$$

where $\mathcal{L}_{ap}$ denotes the loss function of image reconstruction, $\mathcal{L}_{ds}$ denotes the smoothness loss of parallax, $\mathcal{L}_{lr}$ denotes the front-to-back consistency loss, α_{ap}, α_{ds}, and b are the corresponding coefficients, α is the image reconstruction parameter, and *SSIM* is the image structure similarity formula.

3.2. Geometric Cues and Speed Models

Solving the speed directly by distance leads to the superposition of errors because the distance has errors, thereby resulting in inaccurate speed information. Therefore, the speed of the vehicle ahead is estimated directly using geometric cues. In addition, because distance and speed information are directly related, distance information is introduced as an aid to improve the accuracy and stability of speed estimation. According to the basic theory of relative velocity and distance of vehicles, is the following is obtained:

$$v_i = \frac{d_i^t - d_i^{t-1}}{\Delta t}, \tag{13}$$

where $d_i^t - d_i^{t-1}$ is the distance that the vehicle moves between the two frames.

To obtain the velocity component in the lateral direction of the vehicle ahead, the coordinates of the pixel point at the center of the vehicle target frame are used, as well as the inverse perspective projection of the camera.

$$u_i = \frac{(l_i + r_i)}{2}, \tag{14}$$

$$v_i = \frac{(t_i + b_i)}{2}, \tag{15}$$

$$\begin{bmatrix} v_{x_i} \\ v_{y_i} \end{bmatrix} = \frac{1}{\Delta t}\begin{bmatrix} \frac{u_i^t - c_x}{f_x}d_i^t - \frac{u_i^{t-1} - c_x}{f_x}d_i^{t-1} \\ \frac{v_i^t - c_y}{f_y}d_i^t - \frac{v_i^{t-1} - c_y}{f_y}d_i^{t-1} \end{bmatrix}, \tag{16}$$

where v_{x_i} is the lateral velocity of the vehicle ahead, and v_{y_i} is the longitudinal velocity of the vehicle ahead. c_x and c_y indicate the offset of the optical axis concerning the coordinate center of the projection plane, and $\left(u_i^t, v_i^t\right)$ and $\left(u_i^{t-1}, v_i^{t-1}\right)$ are the projection image coordinates of p_i^t and p_i^{t-1}, respectively.

The analysis results show that the speed information of the vehicle ahead is directly related to several parameters, such as the center point of the bounding box, the change of height, the offset of the camera optical axis, the focal length, and the position of the camera. These parameters can be directly obtained by the inherent parameters of the camera, the target bounding box information, and the height of the camera, so the distance

geometric vector, g_d, can be expanded to obtain the expanded geometric vector, g, for speed estimation, as follows:

$$g = [\ \frac{f_y}{b_i^{t-1} - t_i^{t-1}}, \frac{f_x}{r_i^{t-1} - l_i^{t-1}}, \frac{f_y H_c}{b_i^{t-1} - o_y}, \frac{u_i^{t-1} - c_x}{f_x}, \frac{v_i^{t-1} - c_y}{f_y},$$
$$\frac{f_y}{b_i^t - t_i^t}, \frac{f_x}{r_i^t - l_i^t}, \frac{f_y H_c}{b_i^t - o_y}, \frac{u_i^t - c_x}{f_x}, \frac{v_i^t - c_y}{f_y}], \tag{17}$$

where u_i and v_i are directly related to the coordinates, so the geometric cue g can be translated as follows:

$$g = [\ \frac{f_y}{b_i^{t-1} - t_i^{t-1}}, \frac{f_x}{r_i^{t-1} - l_i^{t-1}}, \frac{f_y H_c}{b_i^{t-1} - o_y},$$
$$\frac{b_i^{t-1} - c_x}{f_x}, \frac{r_i^{t-1} - c_x}{f_x}, \frac{t_i^{t-1} - c_y}{f_y}, \frac{b_i^{t-1} - c_y}{f_y}, \frac{f_y}{b_i^t - t_i^t}, \frac{f_x}{r_i^t - l_i^t}, \frac{f_y H_c}{b_i^t - o_y},$$
$$\frac{b_i^{t-1} - c_x}{f_x}, \frac{r_i^{t-1} - c_x}{f_x}, \frac{t_i^{t-1} - c_y}{f_y}, \frac{b_i^{t-1} - c_y}{f_y}\ ]. \tag{18}$$

The motion of the vehicle is analyzed from the pixel perspective, indicating that the relative velocity information of the vehicle is the displacement of each pixel at Δt time interval. This displacement information can be obtained through the optical flow network. The feature vector f_m is used to represent the extracted optical flow information.

In summary, the final model for velocity regression can be expressed as follows:

$$f_{c,m} = Flatten\Big(RoI\Big(E_{depth}(I_{t-1}, I_t) \bigotimes E_{flow}(I_{t-1}, I_t)\Big)\Big),$$
$$f_m = Flatten(RoI(flow(I_{t-1}, I_t))),$$
$$v_i = FC_v\Big(f_d \bigotimes f_{c,m} \bigotimes f_m \bigotimes g\Big), \tag{19}$$

where E_{depth} is the encoder of the deep network, E_{flow} is the encoder of the optical flow network, $f_{c,m}$ is the deep feature vector of the vehicle, f_m is the optical flow feature vector, g is the expanded geometric feature vector, and FC_v is the fully connected layer.

The regression model for velocity contains the parameters required for the distance regression model, as follows:

$$d_i = FC_d\Big(f_d \bigotimes f_{c,d} \bigotimes g_d\Big). \tag{20}$$

The attention mechanism can be viewed as a resource allocation system that reallocates the original equally allocated features according to the importance of the features, which are achieved in neural networks by assigning different weights. Thus, the self-attention mechanism is improved based on the proposed framework, as shown in Figure 3. The self-attention mechanism adjusts the previously obtained depth features, deep features, optical flow features, and geometric features, and it generates an attention map. It forces the model to focus on stable and geometrically meaningful features and can self-adjust without any manual settings to capture the long-term correlation and global correlation, thereby generating better attention-guided maps from a wide range of spatial region features as well as features with different information for joint vehicle speed and distance estimation.

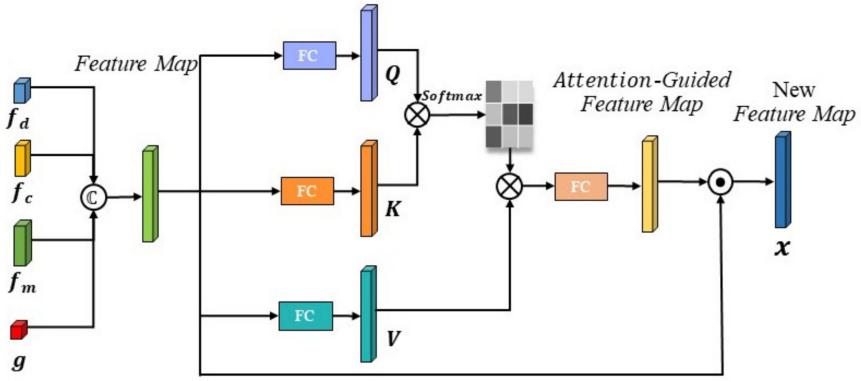

Figure 3. Attention fusion module.

The final regression model for distance velocity is obtained as follows:

$$(d_i, v_i) = FC\left(f_d \bigotimes f_c f_m \bigotimes g\right), \tag{21}$$

The fusion of depth features, f_d, deep features, f_c, optical flow features, f_m, and geometric features, g, by this attention fusion module is shown in Figure 3. Firstly, using the obtained features, x_0, the vectors Q and K are obtained by linear transformations W_Q and W_K, respectively, and the similarity of the inner product of the vectors Q and K is calculated as follows:

$$Q = W_Q(x_0) \tag{22}$$

$$K = W_K(x_0) \tag{23}$$

$$S = softmax\left(Q^T K\right) \tag{24}$$

Then, for the obtained feature x_0, the vector V is obtained by linear transformation again, and the inner product is calculated with the vector S to obtain the associated feature vector F:

$$V = W_V(x_0) \tag{25}$$

$$F = SV \tag{26}$$

Finally, the associated feature vector F is fused with the original feature vector x_0 through the fully connected layer W_F, to obtain the final feature vector x:

$$x = W_F(F) + x_0 \tag{27}$$

The loss function, $\mathcal{L}_{pv}$, used in the distance and velocity regression is regressed on distance and velocity using *MSE* loss, as follows:

$$\mathcal{L}_{p,v} = \alpha\mathcal{L}_p + \beta\mathcal{L}_v = \frac{1}{N}\left(\alpha\sum_i^N\left(d_i - \hat{d}_i\right)^2 + \beta\sum_i^N(v_i - \hat{v}_i)^2\right), \tag{28}$$

where $\alpha = 0.1$, $\beta = 1$, $\mathcal{L}_p$ is the loss function of distance, and $\mathcal{L}_v$ is the loss function of velocity.

4. Experimental Validation of Distance–Velocity Estimation Network

In this section, experiments are conducted on the proposed distance–velocity estimation network. We evaluate the performance metrics of the proposed network on the Tusimple velocity dataset and the original KITTI dataset. Some evaluation metrics for distance and speed estimation are presented, as well as experiments comparing the performance of this proposed network with previous networks.

Evaluation metrics:

1. Abs relative difference (*AbsRel*):

$$AbsRel = \frac{1}{N}\sum_{i=1}^{N}\frac{\left|D_i - D_i^*\right|}{D_i^*} \tag{29}$$

2. Squared relative difference (*SqRel*):

$$SqRel = \frac{1}{N}\sum_{i=1}^{N}\frac{\left|D_i - D_i^*\right|^2}{D_i^*} \tag{30}$$

3. Root mean square error (*RMSE*):

$$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left|D_i - D_i^*\right|^2} \tag{31}$$

4. Root mean square logarithmic error (*RMSlog*):

$$RMSlog = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left|log^{D_i} - log^{D_i^*}\right|^2} \tag{32}$$

5. Accuracy:

$$\max\left(\frac{D_i}{D_i^*}, \frac{D_i^*}{D_i}\right) = \delta < threshold \tag{33}$$

The three different thresholds $\left(1.25,\ 1.25^2, \text{and } 1.25^3\right)$ are generally used in the accuracy metrics to measure the accuracy of the network.

6. Mean squared error (*MSE*):

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(D_i - D_i^*)^2 \tag{34}$$

The overall *MSE* of the three distances was used as the final metric:

$$E_v = \frac{E_{v,far} + E_{v,mid} + E_{v,near}}{3}, \tag{35}$$

$$E_d = \frac{E_{d,far} + E_{d,mid} + E_{d,near}}{3}, \tag{36}$$

where D_i represents the distance of the vehicle ahead, and D_i^* represents the distance of the vehicle in the next frame.

4.1. Experimental Validation of the Tusimple Dataset

In the Tusimple speed estimation challenge rule, the vehicles are initially divided into three groups according to their relative distances. The data distribution of the Tusimple dataset is statistically distributed to obtain the distribution of samples at different distances: near distance (d < 20 m), about 12% of the samples, medium distance (20 m < d < 45 m) about 65% of the samples, and long distance (d > 45 m), about 23% of the samples. Related information is shown in Table 2.

Table 2. Distribution of the number of vehicles labeled in the Tusimple dataset.

Distance (m)	Close Range d < 20 m	Middle Range 20 m < d < 45 m	Long Range d > 45 m	Total
Training Set	166	943	333	1442
Test Set	29	247	99	375

The results of the distance estimation were not provided in the Tusimple speed challenge. Thus, the focus is placed on the comparison of the speed results. The evaluation results are shown in Table 3.

Table 3. Quantitative results of distance velocity estimation on the Tusimple dataset.

Method	Distance	Velocity			
	MSE_P	$MSE_{V,near}$	$MSE_{V,mid}$	$MSE_{V,far}$	MSE_V
Rank1 [17]	-	0.18	0.66	3.07	1.30
Rank2	-	0.25	0.75	3.50	1.50
Rank3	-	0.55	2.21	5.94	2.90
Song et al. [34] (org)	9.72	0.23	0.99	3.27	1.50
Song et al. [34] (full)	10.23	0.15	0.34	2.09	0.86
Huang et al. [27]	7.56	0.10	0.26	1.58	0.65
Our Method	**5.659**	**0.077**	**0.196**	**1.217**	**0.496**

The comparison results of different networks on speed metrics at different distances are shown in Table 3. The table shows that the proposed network achieved the best results in terms of the *MSE* of speed at each distance. Though the target frame of long-distance vehicles is insufficiently rich in deep information, leading to a significant increase in distance and speed estimation errors, the proposed network still achieved good results.

The proposed distance–velocity estimation network on the Tusimple test set yielded a mean velocity *MSE* of 0.496 m^2/s^2, a 42% reduction compared with [42] (full) and a 23% reduction compared with [35]. In terms of distance, the mean distance *MSE* obtained was 5.695 m^2, which is 44% lower than that in [34] (full) and 23% lower than that in [27]. The distance verification results are shown in Table 4.

Table 4. Quantitative results of distance estimation for the Tusimple dataset.

Index	Song et al. [34] (org)	Song et al. [34] (full)	Huang et al. [27]	Our Method
AbsRel ↓	0.037	0.041	**0.034**	0.038
SqRel	0.132	0.152	0.105	**0.076**
RMS	2.700	2.894	2.416	**1.993**
RMSlog	0.059	0.062	0.050	**0.038**
$\delta < 1.25^1$	0.989	0.987	**0.997**	**0.997**
$\delta < 1.25^2$	**1.000**	**1.000**	**1.000**	**1.000**
$\delta < 1.25^3$	**1.000**	**1.000**	**1.000**	**1.000**

The inference was performed on the test set using the network trained on the Tusimple dataset to visualize the prediction effect of the proposed network. Figure 4 and Table 5 show the results of the predicted values of the proposed network in terms of cross-longitudinal distance and velocity compared with the labels.

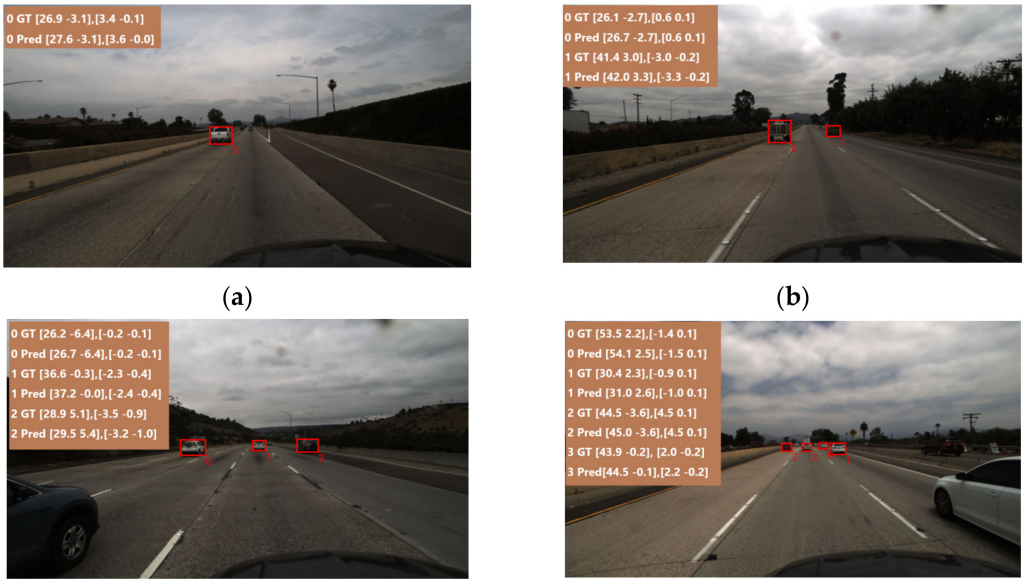

Figure 4. Qualitative visualization results for the Tusimple dataset. Figures (**a**–**d**) are typical feature maps extracted from the Tusimple dataset.

Table 5. Predicted distance velocity estimates obtained corresponding to Figure 4.

Vehicle ID	Distance (m)		Relative Velocity (m/s)	
	True Value	Predicted Value	True Value	Predicted Value
A-0	(26.9, −3.1)	(27.6, −3.1)	(3.4, −0.1)	(3.6, −0.0)
B-0	(26.1, −2.7)	(26.7, −2.7)	(0.6, 0.1)	(0.6, 0.1)
B-1	(41.4, 3.0)	(42.0, 3.3)	(−3.0, −0.2)	(−3.3, −0.2)
C-0	(26.2, −6.4)	(26.7, −6.4)	(−0.2, −0.1)	(−0.2, −0.1)
C-1	(36.6, −0.3)	(37.2, −0.0)	(−2.3, −0.4)	(−2.4, −0.4)
C-2	(28.9, 5.1)	(29.5, 5.4)	(−3.5, −0.9)	(−3.2, −1.0)
D-0	(53.5, 2.2)	(54.1, 2.5)	(−1.4, 0.1)	(−1.5, 0.1)
D-1	(30.4, 2.3)	(31.0, 2.6)	(−0.9, 0.1)	(−1.0, 0.1)
D-2	(44.5, −3.6)	(45.0, −3.6)	(4.6, 0.0)	(4.5, 0.1)
D-3	(43.9, −0.2)	(44.5, −0.1)	(2.0, 0.0)	(2.2, −0.2)

4.2. Experimental Validation of KITTI Dataset

4.2.1. Analysis of Performance Indicators

The speed estimation results of the network on the KITTI dataset are shown in Table 6.

Table 6. Quantitative results of speed estimation for the KITTI dataset.

	$MSE_{V,near}\downarrow$	$MSE_{V,mid}$	$MSE_{V,far}$	MSE_V
Song et al. [34]	0.29	0.93	1.57	0.94
Huang et al. [27]	0.23	0.67	0.96	0.62
Our Method	**0.16**	**0.27**	**0.78**	**0.40**

Table 6 shows that the *MSE* of the proposed network for medium distance velocity was reduced by 59.7% compared with [27], proving the effectiveness of multiple features for distance and velocity estimation.

Table 7 shows the quantitative results of different networks on each metric, and the proposed network had a 13% difference in error on the AbsRel metric compared with the method in [34], with only a 0.9% difference with the DORN method on the RMS metric. However, a 15.5% and a 13% improvement were found in the SqRel and RMSlog metrics, respectively. The accuracy was almost the same as other excellent networks. The proposed network achieved excellent results in distance estimation.

Table 7. Quantitative results of distance estimation for the KITTI dataset.

Indicators	3Dbbox [45]	DORN [9]	Unsfm [8]	Song et al. [34]	Huang et al. [27]	Our Method
AbsRel	0.222	0.078	0.219	0.075	0.098	0.085
SqRel	1.863	0.505	1.924	0.474	0.444	0.375
RMS	7.696	4.078	7.873	4.639	4.240	4.114
RMSlog	0.228	0.179	0.338	0.124	0.127	0.108
$\delta < 1.25^1$	0.659	0.927	0.710	0.912	0.930	0.974
$\delta < 1.25^2$	0.966	0.985	0.886	0.996	0.998	0.997
$\delta < 1.25^3$	0.994	0.995	0.933	1.000	1.000	1.000

4.2.2. Qualitative Visualization Analysis

The results were also visualized on the test set of the KITTI dataset to visualize the prediction effect of the proposed network. Figure 5 and Table 8 compare the prediction results of the proposed network with the labeling results in terms of horizontal and vertical distance and speed.

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

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

Figure 5. Qualitative visualization results on the KITTI dataset. Figures (**a**–**d**) respectively represent the vehicle ahead, the oncoming vehicle, the multi-vehicle in front, and the multi-vehicle in the opposite direction.

As shown in the table, the distance and speed in the longitudinal direction are shown on the left side of the brackets, and the distance and speed in the lateral direction are shown on the right side. After the statistical Table 8, the network can reach an average relative error of 2.1% in terms of distance and 2.6% in terms of relative speed obtained from the KITTI dataset. The proposed network can maintain high accuracy and stability for multiple targets, as well as for the prediction of the oncoming traffic situation.

Table 8. Qualitative results obtained from predictions corresponding to Figure 5.

Vehicle ID	Distance (m)		Relative Speed (m/s)	
	True Value	Predicted Value	True Value	Predicted Value
A-0	(32.3, −0.1)	(32.3, −0.2)	(−0.8, 0.1)	(−0.8, 0.1)
B-0	(42.9, −21.4)	(41.9, −22.4)	(−14.3, −0.2)	(−14.6, −0.2)
B-1	(30.1, −0.6)	(30.4, −0.9)	(−1.7, 0.0)	(−1.7, 0.0)
C-0	(31.8, −3.8)	(30.5, −4.0)	(−24.4, 0.3)	(−23.5, 0.3)
C-1	(53.1, −3.7)	(51.3, −4.7)	(−24.4, 0.1)	(−23.8, −0.1)
C-2	(28.9, 0.1)	(29.3, −0.1)	(1.5, −0.1)	(1.4, −0.1)
D-0	(28.8, −20.8)	(29.2, −21.6)	(−10.3, 0.4)	(−10.3, 0.4)
D-1	(34.0, −20.0)	(33.9, 20.9)	(−10.2, 0.3)	(−9.9, 0.3)
D-2	(25.9, −3.7)	(25.2, −3.8)	(−22.4, 0.0)	(−22.7, 0.1)
D-3	(61.2, −4.2)	(58.4, −5.5)	(−24.0, 0.6)	(−23.4, 0.6)
D-4	(27.3, −0.0)	(27.9, −0.2)	(1.3, 0.2)	(1.2, 0.2)

4.2.3. Visualization Analysis under Different Working Conditions

This section visualizes and analyzes different working conditions separately to clearly show the effect of the range and speed measurement network. The selected video clip scenes are as follows: forward following scene, containing 291 frames of images, lateral incoming scene, containing three targets with 56 frames of images, and opposite incoming scene, containing 17 frames of images.

Figure 6 shows the prediction results of the forward-following scenario. The red box is the obtained bounding box of the target vehicle, and the prediction is performed for each frame to obtain the real-time distance and speed variation of the target vehicle in the forward-following scenario, as shown in Figure 7.

Frame 38

Frame 39

Frame 40

Frame 41

Figure 6. Prediction results of the target vehicle in the forward-following scenario. The green font represents the true values, which are the longitudinal lateral distance and longitudinal lateral velocity; the blue represents the predicted values, which are the longitudinal lateral distance and longitudinal lateral velocity.

Figure 7a,b show that the predicted value and the actual value of the longitudinal distance and the transverse distance are offset at some moments. The maximum offset on the longitudinal distance is around 1 m, and the offset on the transverse distance is around 0.2 m. In addition, the predicted result has the same trend as the actual value, and the

change is relatively smooth, indicating that the network has high accuracy and robustness in the prediction result on the distance. Figure 7c,d show that the predicted value always fluctuates up and down near the actual value in the transverse and longitudinal velocities, and a better result can be obtained for the case of violent vehicle movement. Although the predicted velocity curve has some fluctuations, the magnitude of the fluctuations is maintained within a small range.

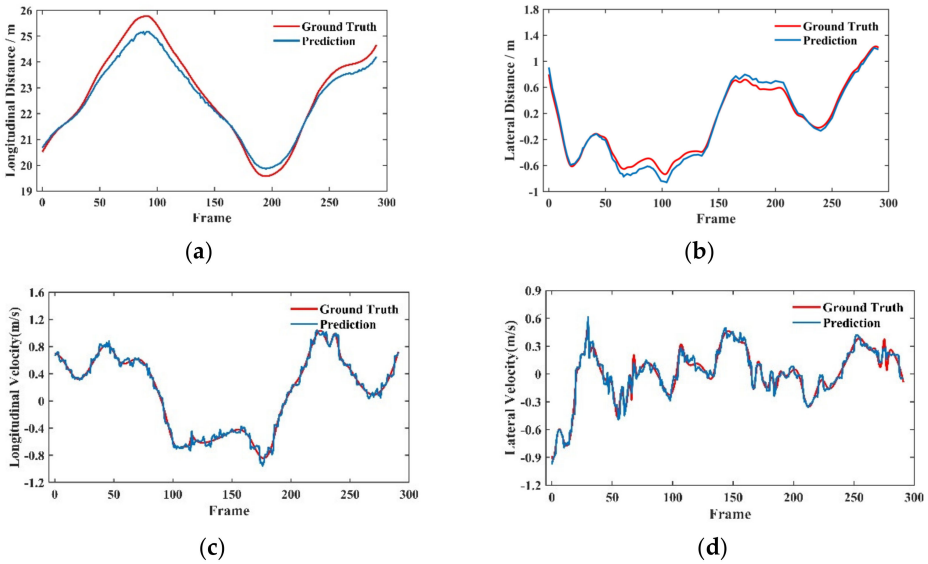

Figure 7. Real–time variation curve of distance and speed of the target vehicle in the forward-following scenario. (**a**) Longitudinal distance, (**b**) Lateral distance, (**c**) Longitudinal velocity, (**d**) Lateral velocity.

Figure 8 shows the prediction results of the lateral incoming vehicle scene. Figure 9 shows the real-time variation curves of the distance and speed of the target vehicle under the lateral incoming traffic scenario.

Figure 8. Prediction results of the target vehicle in the lateral incoming vehicle scenario.

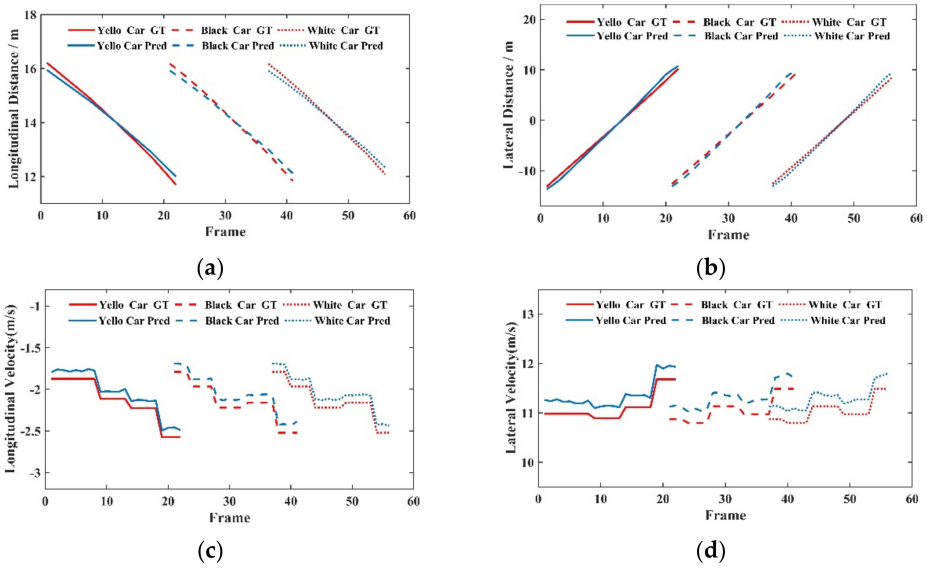

Figure 9. Real-time variation curves of the distance and speed of the target vehicle under the lateral incoming traffic scenario. (**a**) Longitudinal distance, (**b**) Lateral distance, (**c**) Longitudinal velocity, (**d**) Lateral velocity.

Figure 9a,b show that the prediction effect of the network on vehicles with farther lateral distances decreases in the longitudinal and lateral distances. Figure 9c,d show the predicted effect of the vehicle on longitudinal and lateral speeds. The error gradually decreases as the vehicle moves to the front of the self-propelled vehicle. In the transverse longitudinal distance, the slope of the distance prediction curve is different from that of the actual value curve, resulting in a bias between the predicted and actual results. The two reasons for the error in the lateral speed are as follows: on the one hand, the lateral incoming car belongs to a more difficult scene than the following car, which cannot be easily learned by the; on the other hand, the amount of data trained for this scene is relatively small, thereby increasing the error. Through the calculation, the error on the lateral speed is around 2% and that on the longitudinal speed is around 4%, which is still in a very low range.

Figure 10 shows the prediction results of the lateral incoming car scenario. The red box is the boundary box of the obtained target vehicle. Figure 11 shows the real-time distance and speed change of the target vehicle in the lateral incoming car scenario.

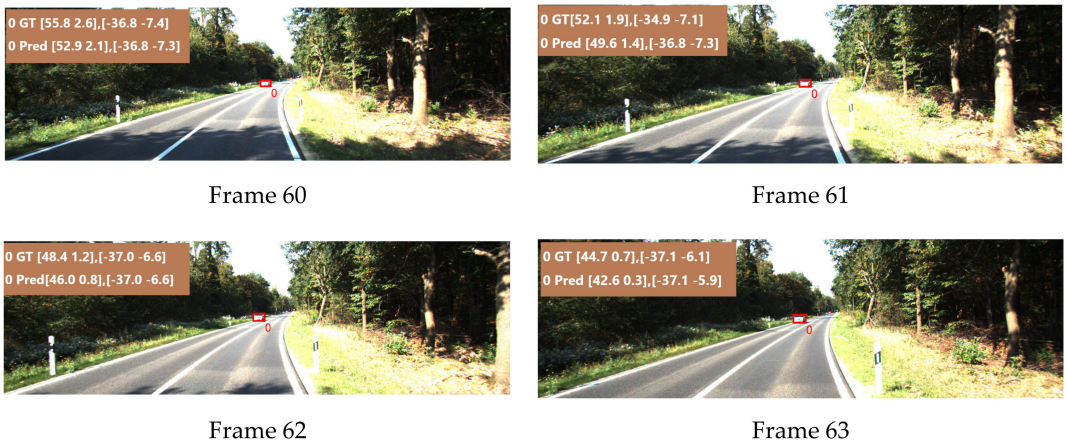

Figure 10. Prediction results of the target vehicle in the opposite direction traffic scenario. The green font represents the true values, which are the longitudinal lateral distance and longitudinal lateral velocity; the blue represents the predicted values, which are the longitudinal lateral distance and longitudinal lateral velocity.

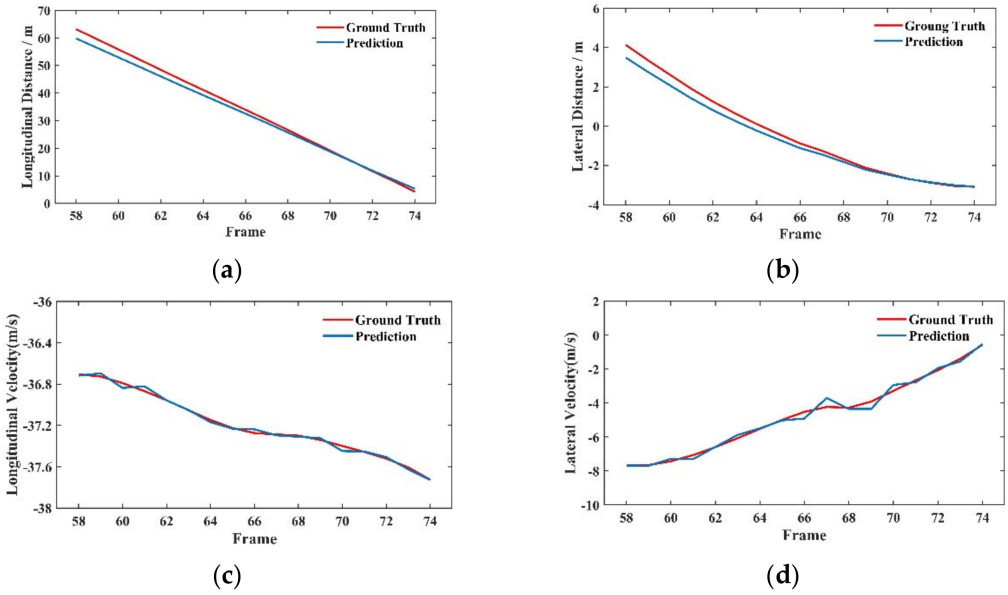

Figure 11. Real–time variation curve of the distance and speed of the target vehicle in the opposite direction of incoming traffic scenario. The red lines represent the true values, (**a**–**d**) are the longitudinal transverse distance and longitudinal transverse velocity.

Figure 11a,b show that the network's prediction effectiveness decreases for long-distance vehicles at longitudinal and lateral distances. This finding is due to the small target frame obtained from the long-range vehicle volume and the insignificant parameter variation, which leads to the reduced effectiveness of the network for long-range target prediction. However, the error gradually decreases as the target vehicle continues to approach the self-vehicle. Figure 11c,d show that the predicted value fluctuates up and down around the actual value in the transverse longitudinal velocity. This finding is due to the fast speed of incoming traffic in the opposite direction, and the fluctuations are elevated compared with the following scenario, but still remain small.

The analysis of the prediction effect of the network under different scenarios indicates that the network has a high prediction accuracy for medium- and close-range targets. Although the prediction accuracy for long-range and lateral farther targets is reduced, it still has a good prediction effect.

In addition to this, the model in this paper is also capable of running in real time, with each vehicle-centric patch running on a single TITAN XP with an inference time of 38 ms. The time consumption results for the different layers are shown in Table 9.

Table 9. Time consumption statistics for different layers.

	Deep Network	Optical Flow Network	Attention Mechanism	Fully Connected Layer	Total Time Consumption
Hardware	GPU	GPU	GPU	GPU	GPU
Time	8 ms	20 ms	0.4 ms	0.3 ms	30 ms

5. Discussion and Conclusions

The main focus of this study was a monocular camera-based distance and speed measurement method for forwarding vehicles in autonomous driving scenarios. The proposed algorithm in this paper enables end-to-end training of a monocular ranging and speed measurement model for ADAS systems. The main work was divided into the following parts.

1. Training process and dataset preparation: Firstly, the target frame parameters of the vehicle were obtained using a typical target detection algorithm. Then, the detected target frame was expanded so that some background information around the target frame could be preserved. Next, the previous frame was sampled and cropped using a vehicle-centric sampling strategy to deal with unbalanced motion distributions and perspective effects to obtain the image pairs for network input. Finally, the extracted image pairs from the Tusimple dataset and the KITTI dataset were used to train the network, and the distance and speed labels of the dataset were extracted and transformed to obtain the distance and speed labels of the targets.

2. We proposed a neural network-based multi-feature fusion distance and speed regression model. Firstly, by deriving the geometric relationship between the target frame information and the information between distance and speed, the information required in the distance and speed estimation was obtained to solve the problems existing in the current distance and speed measurement algorithm. By introducing depth features of images, optical flow features, deep features of vehicles, and geometric features obtained from target frame parameters and camera parameters, the fusion of multiple features was achieved, the accuracy of distance and speed estimation was improved, and a multi-feature distance and speed regression model based on neural network was presented. The attention mechanism was used to fuse different feature cues. The distance and speed information of the vehicle ahead was obtained by constructing distance and speed loss and adding depth loss as an auxiliary loss to regress the distance and speed.

3. On the Tusimple dataset, the mean squared error of the mean velocity of this method was less than 0.496 m^2/s^2, and the mean squared error of the distance was 5.695 m^2. The relative velocity estimation performance of this method was better than the existing techniques in all distance ranges. On the KITTI dataset, the mean speed mean squared error of this method was less than 0.40 m^2/s^2, and the method achieved the best results in most of the metrics and obtained fewer outliers in terms of distance. In addition, the prediction effect plots of the KITTI test set were visualized and the robustness of the model in the KITTI test set was verified. Figures 6–9 show the lateral incoming traffic scenario, and the comparison curve with the true value shows that the error was also within a small range. Figures 10 and 11 show the opposite direction incoming vehicle scenario in the KITTI dataset. From the real-time true value curves of the distance and speed of the target vehicle, it can be seen that the distance error became smaller and smaller as the vehicle came closer and closer, and the speed fluctuation in the horizontal and vertical directions also became smaller and smaller. In summary, the algorithm in this paper has a good effect at medium and long distance, and also has a good performance in other working conditions, which directly proves that the model has a certain generalization ability and is suitable for multi-scene working conditions.

Author Contributions: Conceptualization, S.S. and H.Z.; methodology, C.Q., N.Z. and C.S.; software, C.Q., S.S. and F.X.; validation, C.Q., F.X. and S.S.; formal analysis, C.S., H.Z. and H.X.; investigation, N.Z., S.S. and C.Q.; resources, H.Z. and F.X.; data curation, C.Q., H.Z. and N.Z.; H.X. provided suggestions for the revision. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National key research and development program under 2021YFB2500704, in part the by Grant Science and Technology Development Plan Program of Jilin Province under Grant 20200401112GX, in part by the Industry Independent Innovation Ability Special Fund Project of Jilin Province under Grant 2020C021-3.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to express their gratitude to the editors and the anonymous reviewers for their insightful and constructive comments and suggestions, which have led to this improved version of the paper.

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

References

1. Hasirlioglu, S.; Riener, A.; Huber, W.; Wintersberger, P. Effects of Exhaust Gases on Laser Scanner Data Quality at Low Ambient Temperatures. In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA, 11–14 June 2017.
2. Fascista, A.; Ciccarese, G.; Coluccia, A.; Ricci, G. Angle of Arrival-Based Cooperative Positioning for Smart Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2017**, *19*, 2880–2892. [CrossRef]
3. Shin, J.; Sunwoo, M. Vehicle Speed Prediction Using a Markov Chain With Speed Constraints. *IEEE Trans. Intell. Transp. Syst.* **2018**, *20*, 3201–3211. [CrossRef]
4. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
5. 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*, 28,1137–1149. [CrossRef] [PubMed]
6. Fischer, P.; Dosovitskiy, A.; Ilg, E.; Häusser, P.; Hazırbaş, C.; Golkov, V.; van der Smagt, P.; Cremers, D.; Brox, T. FlowNet: Learning Optical Flow with Convolutional Networks. *arXiv* **2015**, arXiv:1504.06852.
7. Sun, D.; Yang, X.; Liu, M.Y.; Kautz, J. PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
8. Bian, J.; Li, Z.; Wang, N.; Zhan, H.; Shen, C.; Cheng, M.M.; Reid, I. Unsupervised Scale-consistent Depth and Ego-motion Learning from Monocular Video. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 49–65.
9. Fu, H.; Gong, M.; Wang, C.; Batmanghelich, K.; Tao, D. Deep Ordinal Regression Network for Monocular Depth Estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
10. Menze, M.; Geiger, A. Object scene flow for autonomous vehicles in Computer Vision & Pattern Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015.
11. Ma, W.C.; Wang, S.; Hu, R.; Xiong, Y.; Urtasun, R. Deep Rigid Instance Scene Flow. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
12. Vogel, C.; Schindler, K.; Roth, S. 3D Scene Flow Estimation with a Piecewise Rigid Scene Model. *Int. J. Comput. Vis.* **2015**, *115*, 1–28. [CrossRef]
13. Tran, M.T.; Dinh-Duy, T.; Truong, T.D.; Ton-That, V.; Do, T.N.; Luong, Q.A.; Nguyen, T.A.; Nguyen, V.T.; Do, M.N. Traffic Flow Analysis with Multiple Adaptive Vehicle Detectors and Velocity Estimation with Landmark-Based Scanlines. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–22 June 2018.
14. Kampelmühler, M.; Müller, M.G.; Feichtenhofer, C. Camera-based vehicle velocity estimation from monocular video. *arXiv* **2018**, arXiv:1802.07094.
15. Hirschm, H. Stereo Processing by Semiglobal Matching and Mutual Information. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *30*, 328–341. [CrossRef] [PubMed]
16. Li, B.; Shen, C.; Dai, Y.; Van Den Hengel, A.; He, M. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical CRFs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015.
17. Eigen, D.; Puhrsch, C.; Fergus, R. Depth Map Prediction from a Single Image using a Multi-Scale Deep Network. In Proceedings of the 28th Conference on Neural Information Processing Systems (NIPS), Montreal, Canada, 8–13 December 2014.
18. Lee, J.H.; Heo, M.; Kim, K.R.; Kim, C.S. Single-Image Depth Estimation Based on Fourier Domain Analysis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
19. Li, J.; Klein, R.; Yao, A. Two-Streamed Network for Estimating Fine-Scaled Depth Maps from Single RGB Images. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.
20. Laina, I.; Rupprecht, C.; Belagiannis, V.; Tombari, F.; Navab, N. Deeper Depth Prediction with Fully Convolutional Residual Networks. In Proceedings of the 2016 Fourth international conference on 3D vision (3DV), Stanford, CA, USA, 25–28 October 2016.
21. Hu, J.; Ozay, M.; Zhang, Y.; Okatani, T. Revisiting Single Image Depth Estimation: Toward Higher Resolution Maps with Accurate Object Boundaries. In Proceedings of the 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa Village, HI, USA, 7–11 January 2019.
22. Liu, F.; Shen, C.; Lin, G. Deep Convolutional Neural Fields for Depth Estimation from a Single Image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015.
23. Xu, D.; Ricci, E.; Ouyang, W.; Wang, X.; Sebe, N. Multi-Scale Continuous CRFs as Sequential Deep Networks for Monocular Depth Estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
24. Rajagopalan, A.N.; Chaudhuri, S.; Mudenagudi, U. Depth Estimation and Image Restoration Using Defocused Stereo Pairs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 1521–1525. [CrossRef] [PubMed]
25. Saxena, A.; Chung, S.; Ng, A. Learning Depth from Single Monocular Images. *Adv. Neural Inf. Process. Syst.* **2005**, *18*, 1161–1168.
26. Saxena, A.; Sun, M.; Ng, A.Y. Make3D: Learning 3D Scene Structure from a Single Still Image. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *31*, 824–840. [CrossRef] [PubMed]

27. Karsch, K.; Liu, C.; Kang, S.B. Depth Extraction from Video Using Non-parametric Sampling. In Proceedings of the European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; Springer: Berlin/Heidelberg, Germany, 2012.
28. Song, S.; Chandraker, M.; Guest, C.C. High Accuracy Monocular SFM and Scale Correction for Autonomous Driving. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 730–743. [CrossRef] [PubMed]
29. Song, S.; Chandraker, M. Robust Scale Estimation in Real-Time Monocular SFM for Autonomous Driving. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014.
30. Szeliski, R. *Computer Vision: Algorithms and Applications (Texts in Computer Science)*; Springer Science and Business Media: New York, NY, USA, 2010.
31. Eigen, D.; Fergus, R. Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-Scale Convolutional Architecture. In Proceedings of the International Conference on Computer Vision (ICCV), Santiago, Chile, 7-13 December 2015.
32. Atapour-Abarghouei, A.; Breckon, T.P. Monocular Segment-Wise Depth: Monocular Depth Estimation Based on a Semantic Segmentation Prior. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019.
33. Moukari, M.; Picard, S.; Simon, L.; Jurie, F. Deep multi-scale architectures for monocular depth estimation. In Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018.
34. Song, C.; Qi, C.; Song, S.; Xiao, F. Unsupervised Monocular Depth Estimation Method Based on Uncertainty Analysis and Retinex Algorithm. *Sensors* **2020**, *20*, 5389. [CrossRef] [PubMed]
35. Zhe, T.; Huang, L.; Wu, Q.; Zhang, J.; Pei, C.; Li, L. Inter-Vehicle Distance Estimation Method Based on Monocular Vision Using 3D Detection. *IEEE Trans. Veh. Technol.* **2020**, *69*, 4907–4919. [CrossRef]
36. Garg, R.; Bg, V.K.; Carneiro, G.; Reid, I. *Unsupervised CNN for Single View Depth Estimation: Geometry to the Rescue*; Springer: Cham, Switzerland, 2016.
37. Godard, C.; Mac Aodha, O.; Brostow, G.J. Unsupervised Monocular Depth Estimation with Left-Right Consistency. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
38. Zhou, T.; Krahenbuhl, P.; Aubry, M.; Huang, Q.; Efros, A.A. Learning Dense Correspondence via 3D-guided Cycle Consistency. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
39. Zhou, T.; Tulsiani, S.; Sun, W.; Malik, J.; Efros, A.A. View Synthesis by Appearance Flow. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Cham, Switzerland, 2016.
40. Yin, Z.; Shi, J. GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
41. Ilg, E.; Mayer, N.; Saikia, T.; Keuper, M.; Dosovitskiy, A.; Brox, T. FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
42. Song, Z.; Lu, J.; Zhang, T.; Li, H. End-to-end Learning for Inter-Vehicle Distance and Relative Velocity Estimation in ADAS with a Monocular Camera. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020.
43. Huang, K.C.; Huang, Y.K.; Hsu, W.H. Multi-Stream Attention Learning for Monocular Vehicle Velocity and Inter-Vehicle Distance Estimation. *arXiv* **2021**, arXiv:2110.11608.
44. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
45. Mousavian, A.; Anguelov, D.; Flynn, J.; Kosecka, J. 3D Bounding Box Estimation Using Deep Learning and Geometry. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.