# Depth Estimation Using Feature Pyramid U-Net and Polarized Self-Attention for Road Scenes

**Bo Tao** 1,2,*, Yunfei Shen 1,3, Xiliang Tong 3, Du Jiang 4 and Baojia Chen 5

1. Key Laboratory of Metallurgical Equipment and Its Control, Ministry of Education, Wuhan University of Science and Technology, Wuhan 430081, China; syf154772872@wust.edu.cn
2. Hubei Key Laboratory of Mechanical Transmission and Manufacturing Engineering, Wuhan University of Science and Technology, Wuhan 430081, China
3. Precision Manufacturing Institute, Wuhan University of Science and Technology, Wuhan 430081, China; tongxiliang@wust.edu.cn
4. Research Center for Biomimetic Robot and Intelligent Measurement and Control, Wuhan University of Science and Technology, Wuhan 430081, China; jiangdu@wust.edu.cn
5. Hubei Key Laboratory of Hydroelectric Machinery Design and Maintenance, China Three Gorges University, Yichang 443005, China; cbjia@163.com

*Correspondence: taoboq@wust.edu.cn

**Abstract:** Studies have shown that the observed image texture details and semantic information are of great significance for the depth estimation on the road scenes. However, there are ambiguities and inaccuracies in the boundary information of observed objects in previous methods. For this reason, we hope to design a new depth estimation method that can obtain higher accuracy and more accurate boundary information of the detected object. Based on polarized self-attention (PSA) and feature pyramid U-net, we proposed a new self-supervised monocular depth estimation model to extract more accurate texture details and semantic information. Firstly, we add a PSA module at the end of the depth encoder and pose encoder so that the network can extract more accurate semantic information. Then, based on the U-net, we put the multi-scale image obtained by the object detection module FPN (Feature Pyramid network) directly into the decoder. It can guide the model to learn semantic information, thus enhancing the boundary of the image. We evaluated our method on KITTI 2015 datasets and Make3D datasets, and our model achieved better results than previous studies. In order to verify the generalization of the model, we have done monocular, stereo, monocular plus stereo experiments. The experimental results show that our model has achieved better results in several main evaluation indexes and clearer boundary information. In order to compare different forms of PSA mechanism, we did ablation experiments. Compared with no PSA module, after adding the PSA module, better results in evaluating indicators were achieved. We also found that our model is better in monocular training than stereo training and monocular plus stereo training.

**Keywords:** monocular depth estimation; self-supervision; attention mechanism; target detection module

---

## 1. Introduction

Depth estimation is a fundamental problem in computer vision. It can be used in many fields, such as 3D model reconstruction, scene understanding, autonomous driving, computational photography, etc. Generally, depth information can be collected by numerous hardware such as Kinect V1 and Kinect V2 (using the stereo matching and time-of-flight methods, respectively [1]), laser sweep tracers, and structured light sensors. Among them, depth estimation based on a monocular camera is the cheapest scheme, and a monocular camera is the most commonly used camera.

Training strategies for monocular depth estimation are widely used in real-world applications such as shadow detection and removal [2], 3D reconstruction [3,4], and augmented reality [5]. The training methods of monocular depth estimation can be classified into supervised, semi-supervised, unsupervised, and self-supervised methods.
Methods for supervised monocular depth estimation [4–7] can be trained with ground truth and produce good results. However, the ground truth is difficult to obtain in most scenarios. Semi-supervised monocular depth estimation uses large amounts of relatively inexpensive no-labeled data to improve learning performance effectively. The method introduces additional information, such as synthetic data, surface texture, and LiDAR. As a semi-supervised learning method, it reduces the reliance of the model on the ground truth depth map, enhances the consistency of the scale, and improves the accuracy of the estimated depth map.

By contrast, self-supervised monocular depth estimation [8–12], which relies only on stereo image pairs or monocular video for supervised training, has attracted more attention from the industry to academic community. The state-of-the-art (SOTA) self-supervised monocular depth estimation methods [8–11] can successfully estimate the relative depth. However, the existing methods are weak for image edges. The edge contour detail estimation still needs to be improved. To address this problem, We propose a new polarized self-attention (PSA) and feature pyramid U-net self-supervised monocular depth estimation method, which can estimate the depth of the image accurately and preserve the contour lines of the image. The PSA mechanism combines the characteristics of channel self-attention mechanism and spatial self-attention mechanism and connect them in parallel and serial ways novelty; we add it after the encoder 512 layer directly, so we can plug and play without changing the main structure of the network. With this special structure, the model can learn pixel-level semantic features by convolution without significantly increasing the size of the model. Inspired by the U-net model and the object detection module FPN, we pass the original image processed by the maximum pool operation to the decoder. The experiment shows that our method has achieved good results in the task of depth estimation.

The main contributions of this research are as follows:

1. PSA is used in the monocular self-supervised depth estimation model. It can guide the model to learn pixel-level semantic information, so it can get the depth map with more accurate boundaries.
2. We design a new decoder splicing method by combining the skip connection of U-net and FPN. This approach can get better results without significantly increasing the amount of calculation.

2. Related Work

In this section, we describe self-supervised monocular depth estimation methods, the network combining FPN and U-net, and the application of the self-attention mechanism in the task of depth estimation.

2.1. Self-Supervised Monocular Depth Estimation

By combining deep learning [13–16] with depth estimation, an increasing number of strategies are available for monocular depth estimation. Among them, self-supervised monocular depth estimation [17–23] has become a hot research topic in industry and academia because it utilizes the network’s learning ability. The datasets of monocular depth estimation model are left-right image pair [19,21–23] or video sequence [17,18,20]. They do not have real-depth information. The joint loss function guides the convergence of the model. When predicting, they use the trained model and camera matrix to calculate a depth map.

In [24], Xie, et al. proposed discrete parallax estimation of stereo image pairs and used images to estimate training losses. From 3D movies, they extracted stereo image pair datasets. In [8] (MonoDepth), Godard, et al. adds a left–right consistency objective function; the left and right images can be predicted with better depth consistency. Chen, et al. [25] constructed a loss function by using the relative depth relationship, and predicted the pixel-level depth through a multi-scale neural network directly. In addition, they think it is not reasonable to predict left-right parallax maps from a single input image. Zhou, et al. [26]
combined a depth network with a pose network and used the depth network to predict the depth of the object image. The transformation matrix of the camera is predicted using a pose network. In addition, a motion interpretation mask [20, 27] is predicted to encourage the exclusion of nonrigid scene motion regions. Godard, et al. [9] (monodepth2) proposed a pixel-by-pixel minimization of reprojection error to remove occlusions. Masking losses were designed to ignore training pixels that violate the camera motion assumption. At the same time, the multi-scale estimation method [10, 28] was used to upsample and project the input image of the depth decoder to the size of the original image by the network. Finally, the projected images are used to calculate the loss. Although the above methods have achieved good results, their predicted depth does not provide accurate semantic information, and it is even difficult to judge what the detected objects are from the color depth map. That is because the model can not learn the relationship between object boundary information and different textures effectively.

2.2. The Network Combining FPN and U-Net

Object detection module FPN [29] can guide the model to learn more semantic information. Researchers found that the different scale images obtained by maximum pool operation can improve the accuracy of object detection after convolution layer, upsampling and cat operation, respectively. In [30], Song, et al. proposed a new method of monocular depth estimation. With this method, the decoding process can be broken down into different components in order to maximize the benefits of good coding features. In [31], Lai, et al. proposed a densely connected pyramid network for monocular depth estimation. By using dense connection modules, they not only integrate the features of adjacent floors, but also integrate the features of non-adjacent floors. It is different from the traditional pyramid structure neural network that only fuses features between adjacent floors of the pyramid. In monodepth [26], Zhou, et al. built the model with U-net. It has achieved better results than previous studies. So we thought of building a new model by combining FPN with U-net.

2.3. Self-Attention Mechanism

When dealing with semantic segmentation tasks, researchers found that inserting a self-attention mechanism [32–38] into the model can effectively improve the performance of the model. Because of its plug-and-play characteristics, this method is used in various tasks. In CBAM [39], Woo, et al. developed a channel-plus-space self-attention mechanism, which achieved better results than channel-only self-attention. They found that spatial attention mechanism plays a key role in making attention "where". In [40], Huang, et al. used the self-attention mechanism and boundary consistency to build a model, which improved the performance of the depth estimation task. The self-attention mechanism allows the network to improve the depth boundary and image quality through more accurate value and boundary consistency, resulting in a clearer structure. In [41], a self-attention-based depth and motion network was proposed. This framework could capture long-distance context information, resulting in a clearer depth map. We chose the PSA [42] because it achieved good results in the recent semantic segmentation task. We think that the self-attention mechanism and object detection module can guide the model to learn semantic information.

3. Methods

In this section, we describe the self-supervised monocular depth estimation method in detail; it is based on the PSA and object detection module. We chose three forms of data: monocular continuous images, stereo image pairs, monocular continuous images, plus stereo image pairs.

We construct two end-to-end networks based on U-net, a depth network, and a pose network. The two networks are both encoder and decoder structures. The depth encoder extracts the features from a single colorful image, and we can learn the semantic information
from the network. We use the object detection module and depth encoder in parallel. It can pass the extracted boundary information of the detected object to depth decoder, then construct a clearer depth image. At the same time, the pose encoder is also designed to extract pose information from successive frames and calculate the camera matrix by learning the pose relationships between successive images. Finally, we add a PSA module to the end of depth encoder and pose encoder. It is helpful for extracting pixel-level semantic information.

3.1. Network Architecture

The depth estimation task is highly relevant to the target detection and semantic segmentation tasks. We can see the outline of the appearance of the detected object clearly from the depth map generated by the model. For this purpose, we parallelize the object detection module and the depth encoder. After that, pass the result of the object detection module into the deep decoder, as shown in Figure 1.

![Figure 1. Self-supervised monocular depth estimation U-net model with the FPN and the attention mechanism PSA.](image)

After the original image passes through FPN, four pictures of different scales are obtained. They are 2, 4, 8, 16 times of down-sampling size, respectively. They are exactly the same size as the pictures of the skip connection. On this basis, we concatenate them with the up-sampled image. We use it to extract multi-scale features. We do not need real semantic information, labels, and anchors of detected objects. We just need the model to learn the different details from the detected objects and the environment. This is beneficial to generate more accurate boundary information of objects.

For self-attention, we have chosen a lighter PSA mechanism. PSA has two characteristics: (1) filtering: completely collapsing features in one direction while maintaining high resolution in its orthogonal direction; (2) HDR: performing a softmax normalization on the $H \times W \times 1 \times 1$ feature (the smallest feature tensor in the attention block) and then using a sigmoid function for projecting mapping. It increases the dynamic attentional range. Formally, the PSA mechanism can be instantiated as the following two modules (PSA_p, PSA_s).

3.1.1. Channel-Only Branch

In Figures 2 and 3, the content inside the red dotted box is Channel-only Self-Attention. $H \times W$ is the height and width of the picture, $C$ is the number of channels. From the shape
of the cube, we can see how features change inside the PSA module intuitively. The green solid block $A^{ch}$ can be obtained by the following formula:

$$A^{ch}(X) = F_{SC}[W_z](\sigma_1(W_v(X)) \times F_{SM}(\sigma_2(W_q(X))))$$  \hspace{1cm} (1)

where $A^{ch}(X) \in \mathbb{R}^{C \times 1 \times 1}$. $W_q$, $W_v$ and $W_z$ are $1 \times 1$ convolutional layers. $\sigma_1$ and $\sigma_2$ are two tensors reshape operators which change the dimension of tensor. $F_{SM}(X) = \sum_{j=1}^{N_p} \sum_{m=1}^{N_p} e^x_{mj} x_j$, where $F_{SM}(\cdot)$ is a sigmoid function.

**Figure 2.** The attention mechanism PSA_p module. Channel-only Self-attention in the left half and Spatial-only Self-attention in the right half are parallelized and summed. LN is layer normalization. $\otimes$ is matrix multiplication.

**Figure 3.** The attention mechanism PSA_s module. Channel-only Self-attention in the left half and Spatial-only Self-attention in the right half are connected in series. LN is layer normalization. $\otimes$ is matrix multiplication.
\( \odot_{ch} \) denotes the channel multiplication operator, the output \( Z_{ch} \) of the channel-only branch is:

\[
Z_{ch} = A_{ch}(X) \odot_{ch} X \in R^{C \times H \times W}
\]  

(2)

3.1.2. Spatial-Only Branch

The yellow solid block \( A_{sp} \) can be obtained by the following formula:

\[
A_{sp}(X) = F_{SG}\left[\sigma_3\left(F_{SM}\left(\sigma_1\left(F_{GP}\left(W_q(X)\right)\right)\right)\right) \times \left(\sigma_2(W_v(X))\right)\right]
\]

(3)

where \( A_{sp}(X) \in R^{1 \times H \times W} \). \( W_q \) and \( W_v \) are 1 \times 1 convolutional layer. \( \sigma_1, \sigma_2 \) and \( \sigma_3 \) are three tensors reshape operator which change the dimension of tensor. \( F_{GP}(\cdot) \) is the global pooling. Denoting the space multiplication operator by \( \odot_{sp} \), the output of the Spatial-only branch is:

\[
Z_{sp} = A_{sp}(X) \odot_{sp} X \in R^{C \times H \times W}
\]

(4)

From Equations (1)–(4), two forms of PSA modules can be generated. We use the self-supervised monocular depth estimation network monodepth2 [9] as our baseline. It is a U-net model based on an encoder–decoder architecture, allowing for end-to-end output. We use consecutive 3-frame images as input. Different scale images from the depth decoder are used to calculate the loss. In Figure 1, we show the basic framework of the model. We use the ResNet18 [43] network as the encoder backbone for depth network and pose network. To better facilitate the acquisition of accurate semantic information, we insert the PSA module behind the 512 layers. It is used for depth encoder and pose encoder at the same time. Inspired by residual networks and U-net, a parallel object detection module is designed. It scales the image to the corresponding depth solution through the maximum pooling downsample operation. Then stitch the two together. It allows the network to learn the boundary information and the complex texture information effectively.

3.2. Loss Function

As in [44], we upsampled the small-scale images using a bilinear sampling method, which is locally sub-differentiable. The photometric error function \( pe \) of pair images was obtained by using \( L_1 \) norm [45,46] and SSIM [9].

\[
pe(I_a, I_b) = \alpha \left( 1 - \text{SSIM}(I_a, I_b) \right) + (1 - \alpha) \| I_a - I_b \|_1
\]

(5)

where, \( \alpha = 0.85, I_a, I_b \) are two consecutive frames.

Inspired by monodepth2 [9], we also use a pixel-by-pixel minimization of the reprojection error, resulting in a loss of luminosity at each pixel is:

\[
L_p = \min_{\hat{t}} pe(I_t, I_{t \rightarrow \hat{t}})
\]

(6)

As described in [46], we use edge-aware smoothing loss:

\[
L_s = |\partial_x d_{\hat{t}}^i| e^{-|\partial_x h_t|} + |\partial_y d_{\hat{t}}^i| e^{-|\partial_y h_t|}
\]

(7)

where, \( d_{\hat{t}}^i = d_i / d_t \) is the average normalized inverse depth, \( d_i \) is the true depth and \( d_t \) is the predicted depth. The estimation can be stopped counting the depth of contraction.

The final loss function is a composite of Equations (6) and (7):

\[
L = \mu L_p + \lambda L_s
\]

(8)

\( \lambda = 0.001, \) and \( \mu \) is a per-pixel mask. Set \( \mu \) to 0 or 1 to selectively weight each pixel. When the reprojection error of the pixels on the transformed image is smaller than the reprojection error on the original image, it is only brought into the calculation, \( \mu = 1. \)
Our models were implemented in PyTorch [47]. In monocular training, we used Adam [48] for 25 iterations with a batch size of 12 and an input/output resolution of $640 \times 192$. For the first 15 iterations, we used a learning rate of $10^{-5}$ and then the learning rate dropped to $10^{-6}$. We trained on an AMD Ryzen 5 5600X 6-Core, NVIDIA GeForce RTX 3060 (12 G). Our monocular model took 13 h, the stereo model took 14 h, and the monocular plus stereo took 17 h.

4. Results and Discussion

In this section, we discussed the results achieved by our model on publicly available datasets. And the results were compared with the results of SOTA monocular depth estimation models.

We trained our model on the KITTI dataset. It has continuous frames taken by cameras set up on moving vehicles. Then, we evaluated our models on KITTI and Make3D [49]. Finally, we did an ablation study of different PSA modules.

4.1. KITTI Results

The results of the evaluation of the KITTI dataset are shown in Table 1. KITTI 2015 Eigen split [50] was chosen as the evaluation set. We compared our results with other monocular depth estimation models. The models are trained by the self-supervised monocular, self-supervised stereo, and self-supervised monocular plus stereo, respectively.

<table>
<thead>
<tr>
<th>Method</th>
<th>Train</th>
<th>Abs Rel</th>
<th>Sq Rel</th>
<th>RMSE</th>
<th>RMSE log</th>
<th>$\delta &lt; 1.25$</th>
<th>$\delta &lt; 1.25^2$</th>
<th>$\delta &lt; 1.25^3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zhou, et al. [26]</td>
<td>M(K)</td>
<td>0.183</td>
<td>1.595</td>
<td>6.709</td>
<td>0.270</td>
<td>0.734</td>
<td>0.902</td>
<td>0.959</td>
</tr>
<tr>
<td>Yang, et al. [17]</td>
<td>M(K)</td>
<td>0.182</td>
<td>1.481</td>
<td>6.501</td>
<td>0.267</td>
<td>0.725</td>
<td>0.906</td>
<td>0.963</td>
</tr>
<tr>
<td>Mahjourian, et al. [18]</td>
<td>M(K)</td>
<td>0.163</td>
<td>1.240</td>
<td>6.220</td>
<td>0.250</td>
<td>0.762</td>
<td>0.916</td>
<td>0.968</td>
</tr>
<tr>
<td>GeoNet [19]</td>
<td>M(K)</td>
<td>0.149</td>
<td>1.060</td>
<td>5.567</td>
<td>0.226</td>
<td>0.796</td>
<td>0.935</td>
<td>0.975</td>
</tr>
<tr>
<td>DDVO [20]</td>
<td>M(K)</td>
<td>0.151</td>
<td>1.257</td>
<td>5.583</td>
<td>0.228</td>
<td>0.810</td>
<td>0.936</td>
<td>0.974</td>
</tr>
<tr>
<td>DF-Net [21]</td>
<td>M(K)</td>
<td>0.150</td>
<td>1.124</td>
<td>5.507</td>
<td>0.223</td>
<td>0.806</td>
<td>0.933</td>
<td>0.973</td>
</tr>
<tr>
<td>Ranjan, et al. [22]</td>
<td>M(K)</td>
<td>0.148</td>
<td>1.149</td>
<td>5.464</td>
<td>0.226</td>
<td>0.815</td>
<td>0.935</td>
<td>0.973</td>
</tr>
<tr>
<td>EPC++ [23]</td>
<td>M(K)</td>
<td>0.141</td>
<td>1.026</td>
<td>5.291</td>
<td>0.215</td>
<td>0.816</td>
<td>0.945</td>
<td>0.979</td>
</tr>
<tr>
<td>Struct2depth [51]</td>
<td>M(K)</td>
<td>0.141</td>
<td>1.026</td>
<td>5.291</td>
<td>0.215</td>
<td>0.816</td>
<td>0.945</td>
<td>0.979</td>
</tr>
<tr>
<td>Monodepth2 [9]</td>
<td>M(K)</td>
<td>0.115</td>
<td>0.903</td>
<td>4.863</td>
<td>0.193</td>
<td>0.877</td>
<td>0.959</td>
<td>0.980</td>
</tr>
<tr>
<td>Klingner, et al. [11]</td>
<td>M(K)</td>
<td>0.113</td>
<td>0.870</td>
<td>4.720</td>
<td>0.187</td>
<td>0.876</td>
<td>0.958</td>
<td>0.978</td>
</tr>
<tr>
<td>Johnston, et al. [37]</td>
<td>M(K)</td>
<td>0.110</td>
<td>0.872</td>
<td>4.714</td>
<td>0.189</td>
<td>0.878</td>
<td>0.958</td>
<td>0.980</td>
</tr>
<tr>
<td><strong>Current study</strong></td>
<td>M(K)</td>
<td><strong>0.110</strong></td>
<td><strong>0.838</strong></td>
<td><strong>4.706</strong></td>
<td><strong>0.180</strong></td>
<td><strong>0.878</strong></td>
<td><strong>0.960</strong></td>
<td><strong>0.982</strong></td>
</tr>
<tr>
<td>Garg, et al. [52]</td>
<td>S(K)</td>
<td>0.152</td>
<td>1.226</td>
<td>5.489</td>
<td>0.246</td>
<td>0.784</td>
<td>0.921</td>
<td>0.967</td>
</tr>
<tr>
<td>Monodepth R50 [53]</td>
<td>S(K)</td>
<td>0.133</td>
<td>1.142</td>
<td>5.533</td>
<td>0.230</td>
<td>0.830</td>
<td>0.936</td>
<td>0.970</td>
</tr>
<tr>
<td>StrAT [54]</td>
<td>S(K)</td>
<td>0.128</td>
<td>1.019</td>
<td>5.403</td>
<td>0.227</td>
<td>0.827</td>
<td>0.935</td>
<td>0.971</td>
</tr>
<tr>
<td>3Net(R50) [55]</td>
<td>S(K)</td>
<td>0.129</td>
<td>0.996</td>
<td>5.281</td>
<td>0.223</td>
<td>0.831</td>
<td>0.939</td>
<td>0.974</td>
</tr>
<tr>
<td>3Net(R18) [55]</td>
<td>S(K)</td>
<td>0.112</td>
<td>0.953</td>
<td>5.007</td>
<td>0.207</td>
<td>0.862</td>
<td>0.949</td>
<td>0.976</td>
</tr>
<tr>
<td>Monodepth2 [9]</td>
<td>S(K)</td>
<td><strong>0.109</strong></td>
<td>0.873</td>
<td>4.960</td>
<td>0.209</td>
<td>0.864</td>
<td>0.948</td>
<td>0.975</td>
</tr>
<tr>
<td>Hint-Monodepth [56]</td>
<td>S(K)</td>
<td>0.111</td>
<td>0.912</td>
<td>4.977</td>
<td>0.205</td>
<td>0.862</td>
<td>0.950</td>
<td>0.977</td>
</tr>
<tr>
<td><strong>Current study</strong></td>
<td>S(K)</td>
<td><strong>0.111</strong></td>
<td><strong>0.870</strong></td>
<td><strong>4.917</strong></td>
<td><strong>0.205</strong></td>
<td><strong>0.866</strong></td>
<td><strong>0.952</strong></td>
<td><strong>0.977</strong></td>
</tr>
<tr>
<td>UnDeepVO [57]</td>
<td>MS(K)</td>
<td>0.183</td>
<td>1.730</td>
<td>6.571</td>
<td>0.268</td>
<td>−</td>
<td>−</td>
<td>−</td>
</tr>
<tr>
<td>EPC++ [23]</td>
<td>MS(K)</td>
<td>0.128</td>
<td>0.936</td>
<td>5.011</td>
<td>0.209</td>
<td>0.831</td>
<td>0.945</td>
<td>0.979</td>
</tr>
<tr>
<td>Monodepth2 [9]</td>
<td>MS(K)</td>
<td><strong>0.107</strong></td>
<td><strong>0.819</strong></td>
<td><strong>4.751</strong></td>
<td>0.198</td>
<td>0.873</td>
<td>0.955</td>
<td>0.977</td>
</tr>
<tr>
<td><strong>Current study</strong></td>
<td>MS(K)</td>
<td><strong>0.110</strong></td>
<td>0.857</td>
<td><strong>4.741</strong></td>
<td><strong>0.190</strong></td>
<td><strong>0.882</strong></td>
<td><strong>0.960</strong></td>
<td><strong>0.981</strong></td>
</tr>
</tbody>
</table>

From Table 1, we find that our method achieves well when compared with the other self-supervised depth estimation models. Our model achieves the best result in the monocular training. For the stereo training, our model achieves the best result except for the
absolute relative error. For the monocular plus stereo training, our model achieves the best result except for the absolute relative error and square relative error.

The texture information of the important detectors can be found by direct observation that our method also achieves good results, as shown in Figure 4. Therefore, the model with the PSA module improves depth estimation map quality and gets a more accurate boundary by extracting pixel-level semantic information.

![Input monodepth2 GCNDepth Ours](image)

Figure 4. We train our model on the KITTI dataset and compare it with the experimental results of monodepth2 [9] and GCNDepth [58]. The main differences in the experiment are marked with red boxes, and our model achieves better results.

4.2. Make3D Results

Similar to monodepth2 [9], our model was also evaluated on the Make3D benchmark. Make3D has a single-frame RGB-D image without stereo image pairs or video sequences. Therefore, we trained our monocular depth model on the KITTI 2015 [59] datasets. We used the evaluation criterion proposed in monodepth [8], and the results of the comparison are shown in Table 2.

Table 2. Predicted results of our monocular depth estimation model and other current monocular depth estimation models on Make3D. M(K): a self-supervised monocular model trained on KITTI. The best results are identified in bold.

<table>
<thead>
<tr>
<th>Method</th>
<th>Train</th>
<th>Abs Rel</th>
<th>Sq Rel</th>
<th>RMSE</th>
<th>RMSE log</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zhou, et al. [26]</td>
<td>M(K)</td>
<td>0.386</td>
<td>5.328</td>
<td>10.472</td>
<td>0.478</td>
</tr>
<tr>
<td>monodepth2 [9]</td>
<td>M(K)</td>
<td>0.324</td>
<td>3.586</td>
<td>7.415</td>
<td>0.164</td>
</tr>
<tr>
<td>Johnston, et al. [37]</td>
<td>M(K)</td>
<td>0.306</td>
<td>3.100</td>
<td>7.126</td>
<td>0.160</td>
</tr>
<tr>
<td><strong>Current study</strong></td>
<td>M(K)</td>
<td><strong>0.284</strong></td>
<td><strong>2.903</strong></td>
<td><strong>7.011</strong></td>
<td><strong>0.149</strong></td>
</tr>
</tbody>
</table>

We used others’ models directly to derive their error evaluation metrics [60]. From Table 2, we find that our method achieves the best results in evaluating indicators when compared with the other self-supervised monocular depth estimation models.

4.3. Generalizing to Other Datasets

We have also used our method on other publicly available datasets of the road scenes. For example: cityscapes [45], DIODE [61], IDD [62]. We put images of datasets into our model for prediction. The results show that our model can be applied to the new scenes.

The predicted depth maps are shown in Figure 5. It is evident that our model can also achieve better results on other datasets. However, due to differences in lighting, environment, and objects, the displayed texture boundaries are not as accurate as those on the KITTI dataset.
Figure 5. We train our model on KITTI and then test our model by the images from cityscapes, DIODE, IDD. In the first row, the first three pictures are from cityscapes; the fourth picture is from DIODE. The pictures in the second row are from IDD.

4.4. Ablation Study

We conducted an ablation study to compare the effects of adding PSA_p, PSA_s, and both separately. They were added into the 512 layers locations of the depth encoder and the pose encoder [63].

We summarize the data in Table 3. Compared with no PSA module, after adding the PSA module, better results in the evaluating indicator have been achieved in monocular training. For stereo training, our model did not achieve best result in evaluating indicators of absolute relative error. In addition, For monocular plus stereo training, our model did not achieve best result in evaluating indicators of absolute relative error and square relative error.

Table 3. Ablation study. √: PSA module added, ×: no PSA module added. If two forms of PSA module are not added, we also do not add object detection module FPN. M(K): a self-supervised monocular model trained on KITTI, S(K): a self-supervised stereo model trained on KITTI, MS(K): a self-supervised monocular plus stereo model trained on KITTI. The best results are identified in bold.

<table>
<thead>
<tr>
<th>Train</th>
<th>PSA_p</th>
<th>PSA_s</th>
<th>Abs Rel</th>
<th>Sq Rel</th>
<th>RMSE</th>
<th>RMSE log</th>
<th>δ &lt; 1.25²</th>
<th>δ &lt; 1.25²</th>
<th>δ &lt; 1.25³</th>
</tr>
</thead>
<tbody>
<tr>
<td>M(K)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>√</td>
<td>×</td>
<td>0.116</td>
<td>0.865</td>
<td>4.816</td>
<td>0.194</td>
<td>0.874</td>
<td>0.958</td>
<td>0.980</td>
<td>0.980</td>
</tr>
<tr>
<td>×</td>
<td>√</td>
<td>0.117</td>
<td>0.908</td>
<td>4.878</td>
<td>0.192</td>
<td>0.870</td>
<td>0.959</td>
<td>0.982</td>
<td>0.982</td>
</tr>
<tr>
<td>√</td>
<td>√</td>
<td>0.110</td>
<td>0.838</td>
<td>4.706</td>
<td>0.180</td>
<td>0.878</td>
<td>0.960</td>
<td>0.982</td>
<td>0.982</td>
</tr>
<tr>
<td>×</td>
<td>×</td>
<td>0.115</td>
<td>0.903</td>
<td>4.863</td>
<td>0.193</td>
<td>0.877</td>
<td>0.959</td>
<td>0.980</td>
<td>0.980</td>
</tr>
<tr>
<td>S(K)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>×</td>
<td>×</td>
<td>0.110</td>
<td>0.919</td>
<td>5.000</td>
<td>0.207</td>
<td>0.865</td>
<td>0.950</td>
<td>0.976</td>
<td>0.976</td>
</tr>
<tr>
<td>×</td>
<td>√</td>
<td>0.110</td>
<td>0.893</td>
<td>4.958</td>
<td>0.206</td>
<td>0.886</td>
<td>0.950</td>
<td>0.977</td>
<td>0.977</td>
</tr>
<tr>
<td>√</td>
<td>√</td>
<td>0.111</td>
<td>0.870</td>
<td>4.917</td>
<td>0.205</td>
<td>0.866</td>
<td>0.952</td>
<td>0.977</td>
<td>0.977</td>
</tr>
<tr>
<td>×</td>
<td>×</td>
<td>0.109</td>
<td>0.873</td>
<td>4.960</td>
<td>0.209</td>
<td>0.864</td>
<td>0.948</td>
<td>0.975</td>
<td>0.975</td>
</tr>
<tr>
<td>MS(K)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>√</td>
<td>×</td>
<td>0.110</td>
<td>0.857</td>
<td>4.741</td>
<td>0.190</td>
<td>0.882</td>
<td>0.960</td>
<td>0.981</td>
<td>0.981</td>
</tr>
<tr>
<td>×</td>
<td>√</td>
<td>0.117</td>
<td>0.857</td>
<td>4.877</td>
<td>0.191</td>
<td>0.863</td>
<td>0.956</td>
<td>0.982</td>
<td>0.982</td>
</tr>
<tr>
<td>√</td>
<td>√</td>
<td>0.111</td>
<td>0.862</td>
<td>4.742</td>
<td>0.190</td>
<td>0.883</td>
<td>0.960</td>
<td>0.981</td>
<td>0.981</td>
</tr>
<tr>
<td>×</td>
<td>×</td>
<td>0.107</td>
<td>0.819</td>
<td>4.751</td>
<td>0.198</td>
<td>0.873</td>
<td>0.955</td>
<td>0.977</td>
<td>0.977</td>
</tr>
</tbody>
</table>
4.5. Discussion

The results of this research show that using PSA plus feature pyramid U-net is better than traditional methods in semantic information extraction. Our method does not only provide better results in the evaluating indicators but also better boundary information for the actual depth map prediction. Another keypoint compares monocular, stereo and monocular plus stereo training in different modules of PSA. In Table 3, it is evident that adding PSA_p and PSA_s together leads to better results, especially in monocular training. But our models did not achieve the best results on all the evaluating indicators. Therefore, PSA modules are better at monocular training. Finally, we have also verified our model on other datasets, and it can also achieve good results. It proves that the model has a certain generalization.

5. Conclusions

In this study, we proposed a new self-supervised monocular depth estimation method. This method is intended to get a more accurate depth estimation map and to understand a detected object from the predicted depth map intuitively. Compared with the previous models, its predicted depth map has achieved better results in detecting the boundary of the object and the accuracy of depth estimation. Finally, we have also verified our model on other datasets, and it can also achieve good results. It proves that the model has a certain generalization.

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

Funding: This research was funded by the National Natural Science Foundation of China, grant number (51505349, 51575407), Hubei Provincial Department of Education, grant number D20201106, and the Open Fund of Hubei Key Laboratory of Hydroelectric Machinery Design and Maintenance in China Three Gorges University (2021KJX13).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.


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

References


60. Tao, B.; Huang, L.; Zhao, H.; Li, G.; Tong, X. A time sequence images matching method based on the siamese network. Sensors 2021, 21, 5900. [CrossRef] [PubMed]

