Visual Attention-based Self-supervised Absolute Depth Estimation using Geometric Priors in Autonomous Driving

Jie Xiang\textsuperscript{1}, Yun Wang\textsuperscript{2}, Lifeng An\textsuperscript{2}, Haiyang Liu\textsuperscript{2}, Zijun Wang\textsuperscript{2} and Jian Liu\textsuperscript{2}

Abstract—Although existing monocular depth estimation methods have made great progress, predicting an accurate absolute depth map from a single image is still challenging due to the limited modeling capacity of networks and the scale ambiguity issue. In this paper, we introduce a fully Visual Attention-based Depth (VAD) network, where spatial attention and channel attention are applied to all stages. By continuously extracting the dependencies of features along the spatial and channel dimensions over a long distance, VAD network can effectively preserve important details and suppress interfering features to better perceive the scene structure for more accurate depth estimates. In addition, we utilize geometric priors to form scale constraints for scale-aware model training. Specifically, we construct a novel scale-aware loss using the distance between the camera and a plane fitted by the ground points corresponding to the pixels of the rectangular area in the bottom middle of the image. Experimental results on the KITTI dataset show that this architecture achieves the state-of-the-art performance and our method can directly output absolute depth without post-processing. Moreover, our experiments on the SeasonDepth dataset also demonstrate the robustness of our model to multiple unseen environments.

Index Terms—Deep learning for visual perception, computer vision for transportation, range sensing.

I. INTRODUCTION

MONOCULAR Depth Estimation (MDE) is a fundamental problem in the field of computer vision, which refers to predicting the corresponding depth map from a single image. Depth Estimation has broad application prospects in autonomous driving, such as 3D object detection\cite{1}, scene understanding\cite{2}, and obstacle avoidance\cite{3}. Compared to other ways of measuring depth\cite{4,5}, monocular depth estimation has unique advantages in obtaining dense depth maps at low cost. Therefore, MDE has aroused the interest of many researchers. In the past few years, many deep learning-based MDE methods\cite{6–23} have emerged and continual improvements\cite{7–23} have been made. These methods can be divided into supervised methods\cite{6–9} and self-supervised methods\cite{10–23}. Unlike supervised methods, self-supervised methods do not need to rely on ground truth depth for training. Furthermore, existing self-supervised MDE methods can be divided into two categories: self-supervised stereo training methods\cite{10–12} and monocular training methods\cite{13–23}. The former use stereo image pairs while the latter use monocular videos as training data. In contrast to stereo training, monocular training is a more general form of self-supervised methods and easier to obtain training data.

However, existing self-supervised monocular training methods still have difficulties in predicting accurate metric depth. This unsatisfied performance may come from two reasons: the limited modeling power of the model and the limited supervision of loss functions. Many existing self-supervised depth estimation networks\cite{16–18,23,24} adopt residual networks\cite{25} based on an encoder-decoder architecture\cite{26}. Such a depth network shows a powerful capacity for capturing local information and representing hierarchical abstract features, but is insufficient to distinguish important details from interfering information. Due to these characteristics of the model, it is difficult for these methods to predict accurate depth estimation results on patterned surfaces or tiny structures. Thus, we introduce a fully Visual Attention-based Depth (VAD) network. Specifically, we build a decoder based on the designed visual attention blocks and use a visual attention network followed by a dual attention module as the encoder. Benefiting from the ability to dynamically handle long-range dependencies brought by combining channel attention and spatial attention at all stages of the encoder-decoder architecture, our model not only continuously preserves important details but also effectively filters out noise.

On the other hand, most depth networks\cite{13–22} trained on monocular videos only output relative depth due to lacking scale constraints. The commonly used scale recovery method called median scaling\cite{13} requires ground truth depth data that is unavailable at test time in many practical applications. Training a model that directly produces absolute depth in metric units can omit the scale recovery operation like median scaling. An intuitive idea to this end is to use object size\cite{27} but it suffers from seeking ubiquitous fixed-size objects and detecting the appearance size of detected objects. A simple but effective approach is to use camera height with respect
II. RELATED WORK

A. Monocular Depth Estimation

Early deep learning-based works [6], [7] on monocular depth estimation mainly focused on supervised methods and dramatically improved the state-of-the-art performance. However, ground truth depth maps for supervised methods are hard to obtain. To avoid the heavy work of collecting ground truth depth, Garg et al. [10] proposed a self-supervised method inspired by view synthesis task [30], which leverages stereo image pairs to construct the reconstruction loss. Compared to stereo image pairs, monocular videos are much easier to obtain. Therefore, SMILearner [13] leveraged monocular videos to train MDE models. Unlike stereo training approaches [10]–[12], monocular training methods [13], [16] need to simultaneously estimate ego-motion and depth at training time.

Recently, monocular training methods have been extensively studied. Some works [14], [15], [17], [20], [24], [31] explored to design more effective loss functions based on the photometric reconstruction loss. For example, edge-aware depth-normal consistency loss was used in [14], [15], Godard et al. [24] proposed a per-pixel minimum reprojection loss to solve the occlusion problem, a multi-scale loss for alleviating the local gradient problem, and the auto-masking method to filter out stationary pixels. Based on Monodepth2 [24], the use of feature-metric loss [31], temporal geometric consistency [17] and semantic-depth consistency [20] further improved monocular depth estimation accuracy. Our method also follows the general pipeline of Monodepth2 [24]. Besides, some works [22], [32] proposed more powerful convolutional networks to enhance the accuracy of depth estimation. Beyond ResNet-based U-Net, Packnet-sfm [22] leveraged 3D convolutions to improve representational capacity of the network. HR-Depth [32] introduced more dense skip-connections to better fuse multi-scale feature maps. Because of the improvement of loss functions and network architectures, the performance of self-supervised monocular depth estimation has been improved a lot. However, these methods still face challenges in predicting metrically accurate depth due to the scale ambiguity issue and lacking the ability to distinguish important features from noise.

B. Scale Ambiguity

Because of lacking scale constraints, monocular vision suffers from the scale ambiguity issue. In order to produce absolute depth at test time, some MDE works [13]–[22] leveraged post-processing for scale recovery. The typical post-processing technique of median scaling proposed in [13] scales the relative depth map by the ratio of the median of the ground truth depth values to the median of the estimated depth values. Median scaling has been adopted by many subsequent works [17]–[22], [24], [31]. Nevertheless, ground truth depth maps required by median scaling are not always available at test time in many cases. As the use of camera height in classical monocular visual odometry [33], DNet [16] leveraged a dense geometrical constraints module to determine the scale factor by the estimated camera height from every ground point.

Compared to post-processing techniques, training a scale-aware model is more attractive due to its simplicity of testing. To learn metrically accurate depth estimation, [34] pretrained depth network on stereo data of one dataset and then fine-tuned the model on monocular videos of another dataset to preserve the absolute scale, and Packnet-sfm [22] leveraged the camera’s velocity and [35] utilized 4-beam Lidar data to construct scale-aware loss. All of these works require additional sensor data for training. In order to avoid using other sensor data, [23] proposed a camera height-based loss function to enforce metrically-scaled depth during training. Similar to [23], we also leverage camera height to construct scale constraints. However, our method directly determines the pixels corresponding to the ground plane based on predicted depth instead of relying on an extra pretrained ground plane segmentation model as in [23]. Thus, we simplify the training procedure and improve depth estimation accuracy by mitigating the effects of the limited accuracy of the ground plane segmentation model and the uneven ground surface.

C. Attention Mechanism

Attention mechanism is a dynamic process that diverts attention to important features [36]. The pioneering work of visual attention is the spatial attention network called RAM [37]. After that, more attention mechanisms [38]–[41] were proposed, such as self-attention [42] and channel attention [38]. Nowadays, attention mechanisms have been applied in
many visual tasks [37]–[41], [43], [44]. As for monocular depth estimation, the attention-based network has been applied not only to supervised methods [48], [49] but also to self-supervised methods [19]–[21]. In [19], an attention module was used to better perceive contextual information. FSRE-Depth [20] utilized an attention module to fuse depth features and semantic features. CADEPTH-Net [21] applied channel attention for structure perception as well as skip-connection. However, these networks [48], [49], [19]–[21] just added several attention modules to the original network, and the body of the depth network is still the usual localized convolutional network. Therefore, we introduce the visual attention mechanism throughout the processing of the network, so that our model can continuously and adaptively extract important features and remove noise interference at all stages for more accurate depth predictions.

III. METHOD

In this section, we first introduce the basic paradigm for self-supervised monocular depth estimation. Then, we describe the details of our VADEPTH network. Finally, we introduce a simple method for detecting coplanar points and construct a scale loss based on the planar geometric prior.

A. Self-Supervised Monocular Depth Estimation

During the self-supervised MDE model training, the model is optimized by minimizing the difference between the target image and the image synthesized by the source image and the predicted target depth map. In the training process using monocular videos, the source frame is chosen from the same 3D point set of the same 3D points $X^s_t$ as in [45]. Thus, we can synthesize the target image:

$$I_{s \rightarrow t}(u, v) = I_s \langle (u_s, v_s) \rangle,$$

where $\langle \cdot \rangle$ denotes bilinear sampling operation. Following [11], we combine the L1 loss and structural similarity index measure (SSIM) [46] to formulate the photometric error:

$$PE(I_a, I_b) = \alpha ||I_a - I_b||_1 + (1 - \alpha) \frac{1 - SSIM(I_a, I_b)}{2}.$$  \hspace{1cm} (4)

To avoid the false photometric error of occluded pixels in the source frame, we apply per-pixel minimum photometric loss $L_{ph}$, i.e.

$$L_{ph} = \min_{s \in \{t-1, t+1\}} PE(I_t, I_{s \rightarrow t}).$$  \hspace{1cm} (5)

Besides, we adopt the auto-masking method [24] to filter out pixels that do not change the appearance between $I_s$ and $I_t$. The binary mask $\mu$ is computed as

$$\mu = \left[ \min_s PE(I_t, I_{s \rightarrow t}) < \min_s PE(I_t, I_s) \right].$$  \hspace{1cm} (6)

where $[ \cdot ]$ is the Inverse bracket. Like [11], we also use the edge-aware smoothness loss:

$$L_{sm} = ||\partial_u d^*_t e^{-\partial_u I_t} + |\partial_v d^*_t| e^{-|\partial_v I_t}|.$$  \hspace{1cm} (7)

where $d^*_t = d_t/\partial I_t$ is the mean-normalized inverse depth. Considering the gradient locality of the bilinear sampler, we predict multi-scale depth maps in the decoder and compute the individual losses at each scale following [24]. Thus, the final baseline loss with a hyperparameter $\lambda_{sm}$ is defined as

$$L_{baseinc} = \frac{1}{S} \sum_{i=0}^{S-1} \mu L_{ph} + \lambda_{sm} L_{sm},$$  \hspace{1cm} (8)

where $S$ refers to the number of multi-scale depth maps.
B. VADEPTH Network Architecture

As shown in Fig. 1, VADEPTH adopts the improved U-shape encoder-decoder architecture, which takes a single RGB image as input and outputs multi-scale depth maps. Unlike U-net architecture [26], the encoder and decoder in VADEPTH are not completely symmetrical, and an additional Dual Attention Module (DAM) is used to adaptively suppress the unimportant or noisy regions as in [39]. In the decoder, we design a novel Visual Attention Block (VAB). There are five successive VABs in the decoder.

For the encoder, we employ a visual attention network (VAN) [44] to extract multi-scale feature maps $X^e_i, i = 1, 2, 3, 4$. A VAN has four stages, where spatial adaptability and channel adaptability are efficiently implemented by the large kernel attention.

Then the internal features of the encoder are fed into the decoder with skip connections, and the lowest resolution features are taken as the input of the DAM. The DAM adaptively selects the discriminative features as the initial input of the decoder by learning spatial and channel interdependencies of features in parallel.

As for the decoder, we design a novel Visual Attention Block (VAB). There are five successive VABs in the decoder. As illustrated in Fig. 2, the $i$-th VAB has two inputs, the internal features $X^e_{i-1}$ from the encoder and the features $X^d_i$ from the previous VAB. $X^d_i$ is fed into a Spatial Attention Module (SAM), which emphasizes the important regions and suppresses the unimportant or noisy regions as in [39]. In the SAM, the initial spatial attention maps are formed by max pooling and average pooling along the channel axis. Then the concatenated initial attention maps are passed through a convolutional layer with a sigmoid function to obtain the final spatial attention map. The output of the SAM is the element-wise multiplication of the broadcasted final attention map and $X^e_{i-1}$. At the same time, $X^d_i$ is accordingly passed through a convolutional layer and a nearest upsampling layer with a factor of 2 to recover the resolution. Then a Channel Attention Module (CAM) is used to concatenate and fuse the low-level feature from the SAM and the high-level feature from the upsampling operation. For the CAM, as in [38], the concatenated features are processed through the global average pooling layer, fully-connected (FC) layer with ReLU, and FC layer with sigmoid function to generate the channel attention map. Similar to the SAM, the output of the CAM is obtained by element-wise multiplication of the attention map and the original features. After the process of CAM, a convolutional layer is used to output $X^d_{i-1}$. Finally, a convolutional layer with sigmoid is required to process $X^d_{i-1}$ when necessary, and the sigmoid output $\sigma$ can be converted to the depth map $D_i$ with $1/(\alpha\sigma + b)$, which is omitted in Fig. 2 for simplicity. Note that the SAM is removed from VAB when the internal feature from the encoder are unavailable.

C. Co-planar Points Detection on Ground Surface

The depth estimation model obtained by minimizing the baseline loss defined in [45] can only directly output the relative depth. In order to train a depth estimation model that can output absolute depth maps, we need to utilize the geometric information that contains absolute scale. In most self-driving scenarios, the ubiquitous camera height can be considered as a known constant value. Therefore, the prior knowledge about camera height is suitable for constructing scale constraints by computing the camera height errors derived from the estimated depth.

To estimate the camera height from the predicted depth map, we need to detect the co-planar points on the ground surface. We observe that the middle-lower region of the image is part of the ground surface in almost all self-driving cases. Based on such geometric prior knowledge, we suppose that a small stationary rectangular region in image $I_t$ is the ground plane. We use two hyperparameters, $\alpha_u = 0.075$ and $\alpha_v = 0.875$ to predefine an $H \times W$ binary mask $M_{\text{rect}}$ (as shown in Fig. 3) to distinguish whether a pixel is in the rectangular region.

$$M_{\text{rect}}(u, v) = \begin{cases} 1 & \text{if } |0.5 - \frac{u}{W}| < \alpha_u \text{ and } \frac{v}{H} > \alpha_v, \\ 0 & \text{others.} \end{cases}$$

Let $P_{\text{rect}} \in R^{N_{\text{rect}} \times 3}$ be a matrix for a set of 3D points $\{P_t(u, v)\} M_p(u, v) = 1$ in the predefined rectangular region, where $N_{\text{rect}} = 2\alpha_u W (1 - \alpha_v) H$ and $P_t(u, v)$ is calculated from the estimated depth according to (9). Based on these 3D points, we can fit a plane $P_{\text{rect}}$ that does not pass through the origin by solving the equation: $P_{\text{rect}} n = 1$. We take the Moore-Penrose inverse of $P_{\text{rect}}$ to obtain the least-squares solution of the contradictory equations:

$$n = P_{\text{rect}}^+ 1.$$  \hspace{1cm} (10)

After plane fitting, we need to detect all 3D points co-planar with the plane $P_{\text{rect}}$ in the target view using a binary mask:

$$M_p(u, v) = \begin{cases} 1 & \text{if } |P_t(u, v) n - 1| < \delta, \\ 0 & \text{others.} \end{cases}$$

In $M_p$, an element whose value is equal to one indicates that the corresponding point is on the plane $P_{\text{rect}}$. The threshold
\( \delta \) is set to 0.01, which is used to judge whether the points are coplanar or not.

**D. Absolute Scale Loss**

After co-planar points detection on the ground surface, we construct an absolute scale loss term using the camera height. To form a scale constraint, we need to compute the camera height from the estimated depth results:

\[
H_{cam}(u, v) = P_i(u, v)n_e,
\]

where \( n_e = \frac{n}{||n||} \) is the normalized vector of \( n \). Using the estimated heights \( H_{cam} \) and the ground truth height \( h_{gt} \), we define the absolute scale loss as follows:

\[
L_{as} = \frac{1}{||M_p||}||M_p \circ (H_{cam} - h_{gt})||_1,
\]

where \( \circ \) is Hadamard product operator. Combining \( L_{as} \) with \( L_{baseline} \), we have the final objective function:

\[
L = \frac{1}{S} \sum_{i=0}^{S-1} \mu L_{ph} + \lambda_{sm}L_{sm} + \lambda_{as}L_{as}.
\]

In our setting, \( \lambda_{sm} \), \( \lambda_{as} \) and \( S \) are set to 0.001, 0.01 and 4, respectively.

**IV. EXPERIMENTAL RESULTS**

**A. Implementation details**

We use a single Nvidia GeForce RTX 2080 Ti to implement our models in Pytorch. In our setting, we use the VADepth network as our depth network with an input/output resolution of 192 \( \times \) 640. For pose estimation, we use a modified ResNet18 \([25]\) to predict a single 6-DoF ego-motion of two adjacent frames following \([24]\). The pose network adopts the same input resolution as the depth network. Both the encoder of the depth network and the encoder of the pose network are initialized with weights pretrained on ImageNet \([47]\). The depth network and pose network are jointly trained with Adam optimizer for 20 epochs. The learning rate is set to \( 5.0 \times 10^{-5} \) for the first 15 epochs and then drops by half for the remaining epochs. The source code and models are available at [https://github.com/xjixzz/vadepth-net](https://github.com/xjixzz/vadepth-net).

**B. Evaluation on KITTI**

KITTI dataset \([28]\) is the commonly used dataset for autonomous driving. We use the Eigen split \([6]\) of KITTI Stereo dataset to evaluate our method with the metrics proposed in \([6]\). Following \([13]\), we also remove the static frames before training, which results in 39810 monocular triplets for training, 4424 images for validation and 697 images for test. The resolution of these images is approximately 375 \( \times \) 1242. All images are required to resize to 192 \( \times \) 640 for training and evaluation. At test time, the depth network only outputs the depth map with the resolution of 192 \( \times \) 640, which is then resized to the full resolution of the original RGB image for evaluation.

To evaluate the performance of our depth network, we first train VADepth network only with the baseline loss function. We report the results of applying median scaling at the top of Table I where we can see that our model achieves the best scores in the metrics of Abs Rel, RMSE log, \( \delta_1 \), and \( \delta_2 \), and ranks second or third in the remaining metrics. Among these methods, CADepth \([21]\) achieves close performance to VADepth but requires much more parameters, due to the underutilization of visual attention, especially ignoring the channel and spatial interdependencies of low-level features.

Besides, we also combine our absolute scale loss to train our model so that the depth network can directly output metrically accurate depth without post-processing that rely on the ground
Fig. 3. Quantitative results on the Eigen split of KITTI dataset. Compared to other methods [20], [21], [24], our model not only outputs smoother depth in patterned planar regions, but also preserves better details for small objects. White boxes highlight the difference. Best viewed in color and zoom in.

### TABLE II
Generalization performance on SeasonDepth

| Method               | Train | Resolution | KITTI Eigen test | SeasonDepth test | Relative Range |
|----------------------|-------|------------|------------------|------------------|----------------|
|                      |       |            | AbsRel ↓ | a1 ↑ |          | AbsRel ↓ | a1 ↑ | Variance(10^{-2}) | AbsRel ↓ | a1 ↑ | 1 - a1 ↓ |
| Eigen et al. [6]‡    | D     |             | 0.203   | 0.702 | 1.093  | 0.340  | 0.346 | 0.0170          | 0.206    | 0.0746 |
| VNL(ResNet101) [8]‡ | D     |             | 0.072   | 0.938 | 0.306  | 0.527  | 0.126 | 0.166           | 0.400    | 0.290  |
| BTS(DenseNet161) [9]‡| D     |             | 0.060   | 0.955 | 0.676  | 0.209  | 0.545 | 0.0650          | 0.405    | 0.129  |
| SfMLearner [13]‡     | M     | 128 × 416  | 0.181   | 0.733 | 0.360  | 0.495  | 0.0801 | 0.0628          | 0.269    | 0.182  |
| Monodepth2 [24]      | M     | 192 × 640  | 0.115   | 0.877 | 0.266  | 0.611  | 0.0410 | 0.0457          | 0.266    | 0.202  |
| FSRE-Depth [20]      | M     | 192 × 640  | 0.105   | 0.886 | 0.256  | 0.624  | 0.0288 | 0.0283          | 0.227    | 0.158  |
| CADepth-Net [21]     | M     | 192 × 640  | 0.105   | 0.892 | 0.257  | 0.625  | 0.0447 | 0.0725          | 0.265    | 0.278  |
| VADepth (Ours)       | M     | 192 × 640  | 0.104   | 0.892 | 0.230  | 0.667  | 0.0158 | 0.0215          | 0.205    | 0.179  |

In the Train Column, D refers to ground truth depth supervision and M refers to self-supervised monocular supervision. ↓ means the lower the better while ↑ means the higher the better. Methods marked with ‡ indicate that the corresponding test results are taken from [29].

truth depth maps. As listed at the bottom of Table I we compare our methods with existing self-supervised absolute depth estimation methods that do not rely on the ground truth depth. Among these methods, our method achieves the best results on all metrics without requiring extra pretrained road segmentation network [23] or velocity supervision [22] at training time, or post-processing [16] at test time.

The qualitative results are presented in Fig. 3. Compared with [20], [21], [24], our model predicts smoother and more accurate depth for patterned surfaces (e.g. the traffic sign in row 1, the banner in row 3, and the bus in row 5), which validates the effectiveness of our model to suppress unnecessary information. Meanwhile, our model also works well for thin things, such as the thin poles in row 2 and the traffic sign in row 4, which reflects the ability to highlight the important details. Taken together, these results illustrate that incorporating the attention mechanism throughout the whole depth network can help to improve the ability of the model to automatically distinguish important contextual information from redundant information for better depth estimation.

### C. Generalization performance

SeasonDepth dataset [29] is an MDE dataset that contains the same multi-traverse routes under 12 different environmental conditions. The test set of SeasonDepth contains 17225 RGB images and corresponding ground truth depth maps with the resolution of 768 × 1024. We use the model trained on KITTI to evaluate the generalization ability of our method on the test set of SeasonDepth without fine-tuning. Results in Table II show that our model generalizes better than existing supervised methods and self-supervised monocular training methods, which proves that our model is more capable of learning transferable feature representation.

We also provide the quantitative comparison in Fig. 4. The inferior results of CADepth [21] and FSRE [20] suggest that applying channel attention in the decoder is not enough to obtain high-level features that are robust to lighting and weather changes. Despite extreme light conditions or challenging weather conditions, the output of our model still correctly reflects the scene structure. This further verifies that robust low-level feature representations extracted from our encoder and SAM are also important for forming robust high-level features.

### D. Ablation Study

To verify the effectiveness of each component in our method, we perform an ablation study on the KITTI dataset. Ablations of the VADepth network are reported in Table III where all models are trained with the loss in (14). We use
Besides, we also list the results of selecting different values between important features and noises. Thus, we empirically select $\alpha_u$ and $\alpha_v$ as 0.075 and 0.875, respectively.

Moreover, we find that absolute scale loss helps alleviate the impact of lane lines on depth estimation. As shown in Fig. 5 although these networks can not predict smooth depth at the solid white line unless absolute scale loss is used, both VADepth and Monodepth2 trained with the absolute scale loss output more accurate depths at the lane line, which suggests that adopting supervision of camera height can lead to smooth depth estimates of the ground surface, in addition to absolute scaled depth. This further demonstrates that our network can effectively learn the geometric knowledge of autonomous driving scenarios with the proposed loss.

V. CONCLUSION

In this work, we presented a novel deep learning architecture for self-supervised monocular depth estimation and proposed an absolute scale loss to supervise the scale of estimated depth maps. Experimental results show that our method achieves state-of-the-art performance with or without median scaling on theKITTI dataset and generalizes well on the unseen changing environments. In the future, we plan to further improve the generalization capacity and interpretability of the depth estimation network by visualizing the attention map and using the results of semantic segmentation to guide the network to strengthen the responses of different channels to specific semantic information or improve the distinction between important features and noises.
