Attention-Based Dense Decoding Network for Monocular Depth Estimation

JIANRONG WANG1,2, GE ZHANG1, MEI YU1,2, TIANYI XU1, AND TAO LUO1,

1College of Intelligence and Computing, Tianjin University, Tianjin 300350, China
2Tianjin Key Laboratory of Advanced Networking, Tianjin 300350, China

Corresponding author: Tao Luo (luo_tao@tju.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61977049, and in part by the Tianjin Key Laboratory of Advanced Networking.

ABSTRACT Depth estimation is a classic computer vision task and provides rich representation of objects and environment. In recent years, the performance of end-to-end depth estimation has been significantly improved. However, the stack of convolutions and pooling operations result in losing local detail spatial information, which is extremely important to monocular depth estimation. In order to overcome this problem, in this work, we propose an encoder-decoder framework with skip connections. Based on the self-attention mechanism, we apply the channel-spatial attention module as a transition layer, which captures the depth and spatial positional relationship and improves the presentation ability of channel and space. Then we propose a dense decoding module to make full use of the attention features of different scale ranges in the decoding process. It achieves a more massive and denser receptive field while obtaining multi-scale information. Finally, a novel distance-aware loss is introduced to predict more meticulous edges and local details in the distance. Experiments demonstrate that the proposed method outperforms the state-of-the-art on KITTI and NYU Depth V2 datasets.

INDEX TERMS Atrous spatial pyramid pooling, depth estimation, self-attention, dense decoding module.

I. INTRODUCTION

Depth information is vital for real-world 3D scenes, and depth estimation is an essential part of understanding the geometric relationship in the scenes. Generally, it can enhance many recognition tasks and have full applications in multimodal emotion recognition, 3D reconstruction, simultaneous localization and mapping (SLAM), 3D object detection and autonomous driving.

Monocular depth estimation refers to estimating the corresponding depth based on one rgb image. As early as the 1960s, scholars have studied the topic of estimating depth from the scene. Previous methods were generally based on optical geometric constraints or environmental assumptions [1], [2]. In recent years, deep convolutional neural networks have made great progress in image classification [3], semantic segmentation [4] and object detection [5]. At the same time, more and more scholars apply a deep convolutional neural network to monocular image depth estimation [6]–[9].

Despite the success of convolutional networks, there are severe challenges in depth estimation. Modern convolutional neural networks have multi-layer or even super multi-layer convolution operations. As the network goes deeper, the receptive field of the posterior layer neurons will increase, and different levels of features in the image region can be obtained. With a stack of several convolutions and pooling, the deep features of each layer change from generalized features to high-level semantic representations. Furthermore, operations such as pooling can integrate and abstract these features. Although consecutive convolutions and pooling reduce the spatial dimension and prevent overfitting to some extent, they make the local spatial information lost. In order to overcome this problem, some works [6], [10] used skip connections to add relatively shallow generalized features to relatively deep semantic features.

Another challenge comes from the image distortion in the distance. As shown in FIGURE 1, the picture frame far away from the camera is blurred. Some methods [12], [13] exploited atrous spatial pyramid pooling (ASPP) to capture objects of different sizes. Although ASPP effectively obtains different receptivity sizes without changing the image resolution, there are three problems with the dilated convolution:
first, the kernel of the dilated convolution is discontinuous, and not all pixels are involved in the operation; second, how to design a suitable dilated convolution kernel is the key to deal with objects of different sizes; third, this approach does not consider the relationship between different pixel feature.

The main goal of this work is to effectively alleviate the problems of blurred local and distant details as mentioned above. We propose an attention-dense decoding network, which pays attention to the distance in an Encoder-Decoder framework. The first part of the decoder is the Channel-Spatial Attention Module (CSAM). Since the convolution extracts feature by mixing cross channel and spatial information, there is a weak dependence in the feature map. However, the informational context is especially important for depth estimation. We apply CSAM as the intermediate transition layer to enhance the representation of information feature in channel and spatial dimensions. Here, channel attention is adopted to capture the dependence of objects in the depth map no matter what the distance of these objects are. The spatial attention selectively aggregates feature of each position through a weighted summation of feature at all locations. Following the attention module, we propose a dense decoding module (DDM), which cascades multiple dilated convolutional layers and adds all previous features as next inputs in the way of dense up-sampling. In this way, multiple dilated convolutions generate wider and denser features. Finally, we propose a distance-aware loss which enforces accurately depth prediction for far away objects. To summarize, our main contributions are three-fold:

- We cascade channel-spatial attention module and dense decoding module to explore the dependencies between arbitrary channel and spatial feature in depth estimation, and enhance the discrimination ability of attention feature representation in a more intensive way.
- We propose a novel distance-aware loss which predicts more meticulous edges and local details in the distance.
- Experimental results show that the proposed framework achieves state-of-the-art performance on KITTI [14] and NYU Depth V2 [15] datasets.

II. RELATED WORK

In this section, we mainly discuss the related works based on deep learning from two perspectives, which are unsupervised and supervised depth estimation.

A. SUPERVISED DEPTH ESTIMATION

In recent years, the deep convolutional network have been applied to depth estimation due to its robust feature representation capability [10], [13], [16]–[21]. Eigen et al. [16] first proposed to use a multi-scale convolutional neural network for depth estimation. They performed a coarse estimation of the scene structure in the rough estimation stage and then performed a more accurate depth prediction in the fine estimation stage. Based on this work, [22] proposed a three level framework by introducing depth estimation, surface normal prediction, and semantic annotation. Liu et al. [18] used a conditional random field (CRF) and deep convolutional neural network to explicitly simulate the complicated relationship between adjacent parts of the depth map. The model extracted the relevant features in the image through CNN and then used CRF to enhance the smoothness and edge preservation of adjacent superpixel blocks. In order to improve the output resolution, Laina et al. [10] proposed a new practical method for learning feature upsampling. For optimization, they introduced the reverse Huber loss function [23]. Lee et al. [24] used the depth network to predict the depth gradient and took it as local cues, and then further estimated the global and coarse depth. Finally they integrated the complementary prediction into the unified depth CNN framework to estimate the final depth image. These methods simulate depth estimation as a regression problem and train the regression network by minimizing the mean square error, which is affected by slow convergence and unsatisfactory local solutions. In order to eliminate or at least significantly reduce these problems, Fu et al. [13] used a multi-scale dilated convolution and spacing-increasing discretization (SID) strategy to avoid unnecessary spatial pooling and capture multi-scale information. Chen et al. [25] proposed an attention-based context aggregation network to solve grid artifacts in [13]. However, they only consider information unilaterally from channel attention and fail to combine pixel spatial level with channel level attention to obtain more detailed and comprehensive contextual information.

B. UNSUPERVISED DEPTH ESTIMATION

Without requiring the ground truth labels, unsupervised methods use photometric constraints from multiple views to predict depth. It was proposed in [17], [20] that utilized epipolar geometry to train the depth estimation network (DepthNet). SfMLearner [26] was the first framework to learn both depth and ego-motion using monocular videos. Following this framework, [27]–[30] have been proposed to solve the challenge of moving objects that violated the rigid scene assumption. Recent work [4], [31], [32] jointly learnt multiple tasks based on [26]. All unsupervised methods include a

FIGURE 1. (Top left) input. (Top right) ground truth. (Bottom left) results of [11]. (Bottom right) ours.
DepthNet. Therefore a reasonable DepthNet is crucial for the performance improvement of unsupervised methods.

III. METHOD
In this section, we first introduce the overall framework, then describe the CSAM and dense decoding module that captures the long-range context information from the channel and spatial level. Finally we introduce the distance-aware loss function in this paper.

A. OVERVIEW
Inspired by the success of the encoder-decoder framework, our work is based on the structure of encoder-decoder, as shown in FIGURE 2. Four intermediate features of the encoder are separately upsampled, convoluted and concatenated with the output features of decoder, and finally fine-tuned to obtain a depth map. SAM and CAM are the Spatial Attention Module, Channel Attention Module respectively.

B. SPATIAL ATTENTION MODULE
There is a relationship between adjacent pixels in the same feature map because there may be a shared area at the time of the upper convolution. When the distance between two-pixel points in the feature map gradually increases, the relationship reduce or even become irrelevant. This dependence is necessary for distinguishing objects of different depths. Based on the self-attention mechanism [33], [34], we introduce the spatial attention module to more accurately characterize the contours of objects of different depths by modeling the relationship between any point and global feature. Given a local feature \( f \in \mathbb{R}^{C \times H\times W} \), we first feed it into three independent \( 3 \times 3 \) convolution layers to generate three new features \( q^p, k^p \) and \( v^p \) (\( q^p \in \mathbb{R}^{WH \times C}, k^p \in \mathbb{R}^{C \times WH}, v^p \in \mathbb{R}^{WH \times C} \)), respectively. After that, multiplying \( q^p \) by \( k^p \) and calculating the softmax on the horizontal axis, we can obtain the spatial attention ‘map’ of a pixel to other pixel points. In order to get a more stable gradient, the ‘map’ is divided by \( \sqrt{d_k} \), where \( d_k \) is the number of channels. Then we perform a matrix multiplication between ‘map’ and \( v^p \) and restore to \( p \in \mathbb{R}^{C \times H \times W} \). Finally, we multiply it by a scale parameter \( \alpha \) and perform an element-wise sum operation with the feature \( f \) to obtain \( P \in \mathbb{R}^{C \times H \times W} \) as follows:

\[
P = \alpha \cdot p + f \tag{1}
\]

\[
p = \frac{\text{softmax}(q^p \cdot k^p)}{\sqrt{d_k}} \cdot v^p \tag{2}
\]

C. CHANNEL ATTENTION MODULE
We build a channel attention module by taking advantage of the relationship between the different channels of the feature map. Since each channel of the feature map is regarded as a feature detector [35], the feature map channels focus on what it is. In order to summarize spatial information, [36] adopted average pooling to calculate spatial statistics in their attention module. Besides, it was proposed in [37] that used global pooling and max-pooling to collect two important cues that highlight the unique object characteristics. [38] combined attention mechanism and dilated convolution to learn optical flow. Different from the above work, we describe how to represent complex channel context relationship in depth.
estimation tasks. Similar to spatial attention module, we first feed \( f \) into convolutions to generate three new features \( q^c, k^c \) and \( v^c \), and get channel attention ‘map’ through matrix multiplication, softmax and \( \sqrt{d_k} \), where \( d_k \) is the number of pixel points in one channel. Then we perform matrix multiplications between channel ‘map’ and \( v^c \) and restore to \( c \in \mathbb{R}^{C \times H \times W} \). Finally, we multiply it by a scale parameter and perform an element-wise sum operation with the feature \( f \) to obtain \( C \in \mathbb{R}^{C \times H \times W} \) as follows:

\[
C = \beta \cdot c + f
\]
\[
c = \frac{\text{softmax}(q^c \cdot k^c)}{\sqrt{d_k}} \cdot v^c
\]

D. DENSE DECODING MODULE

When depth estimation is considered as a regression problem, the low-resolution features are restored to the same resolution as the color map by multiple up-sampling. In order to recover the local detail information, the relatively shallow features are spliced to the relatively deep features by skip connection. Inspired by DenseNet [39] and AsppBlock [12], the dilated convolution rates of 3, 6, 12, 18 are combined into a dense cascade. As shown in FIGURE 3, the dense decoding module contains multiple up-projection [10] and AsppBlock. Given an input feature \( f_i \), the output of up-projection, is \( u_i \), and the output of AsppBlock is \( a_i \). Then \( a_i \) element-wise sum with \( u_i \) to get \( f_{i+1} \), which is regarded as the input of AsppBlock\(_{i+1} \) and a part of the up-projection\(_{i+1} \). The formula is as follows:

\[
f_{i+1} = a_i + u_i
\]
\[
u_i = U_i(u_{i-1} + f_i)
\]
\[
a_i = A_i(f_i)
\]

where \( U_i \) and \( A_i \) denote up-projection, and AsppBlock, respectively.

![FIGURE 3. The detailed structure of the dense decoding module. AsppBlock has 4 colors, each color corresponding to one dilation rate, and the rates from left to right are 3, 6, 12, 18, respectively.](image)

Stacking all the dilated convolutions in a more dense manner can produce multiscale features and cover a larger scale and a more dense range. All up-projection\(_i \) share parameters. The purpose is to get the same resolutions and channels of skip features when the feature passing through the AsppBlock\(_i \). For all AsppBlock\(_i \), the bilinear interpolation is firstly performed, and the interpolation size is consistent with \( e_{5-i} \) of the encoding layer. Then we choose to implement two convolutions. The first convolution kernel size is \( 1 \times 1 \), the input and output channels are the same, and the second size is \( 3 \times 3 \). The second kernel uses different dilation rates, and the number of output channels is half of the input.

E. LOSS FUNCTION

Many previous studies took depth estimation as a regression problem. \( l_1 \) and \( l_2 \) were used to minimize the difference between predicted depth values and ground truth. For \( l_1 \) loss, the formula is:

\[
l_1 = \frac{1}{n} \sum_{i=1}^{n} e_i
\]

where \( e_i = \| y_g^i - y_p^i \|_1 \), \( y_g \) is the ground truth and \( y_p \) is the predicted map. We find that most methods are realistic for near-field. However, the prediction of distant scenes are distorted or fuzzy. We believe that we should pay more attention to the distant scene while calculating the near scene. If \( l_1 \) loss function is used, the depth difference between the predicted map and the ground truth in the distance and near scene has the same contributions to the \( l_1 \) loss, which has the same penalty for near and far scenes. The key idea of distance-aware loss is that we give greater weight to the farther depth difference. Even if it is only 1cm, the total loss value is still large due to its weight. Through back propagation, network parameters are optimized to make the distant depth difference as small as possible. When the depth difference is small enough in the distance, the total loss function is still small even if multiplied by a relatively large weight. Based on the loss function in [11], we propose the distance-aware loss that pays more attention to the long-distance:

\[
l_d = \frac{1}{n} \sum_{i=1}^{n} F(e_i^*)
\]

where

\[
F(x) = \ln(x + \alpha)
\]
\[
e_i^* = (y_g^i + \lambda) \| y_g^i - y_p^i \|_1
\]

where \( \alpha \) and \( \lambda \) are hyper-parameters. In practical, \( \alpha = 0.5 \) and \( \lambda \) is calculated as:

\[
\lambda = \frac{y_g - \min(y_g)}{\max(y_g) - \min(y_g)}
\]

Similarly, it is necessary to punish more errors at the edge of the distant object, which makes the edge structure clearer.

\[
l_g = \frac{1}{n} \sum_{i=1}^{n} (F(\partial_x(e_i^*))) + F(\partial_y(e_i^*))
\]

where \( \partial_x(e_i^*) \) and \( \partial_y(e_i^*) \), respectively, are the derivative of \( e_i \) calculated relative to \( x \) and \( y \) at the \( i \)-th pixel.

Finally, the angle between the surface normals of the depth map \( n_i^p \) and the ground truth \( n_i^- \) is calculated to complement...
the two other kinds of loss and further improve the fine details of the depth map prediction.

\[
l_n = \frac{1}{n} \sum_{i=1}^{n} \left( 1 - \frac{n_i^g \cdot n_i^p}{\|n_i^g\| \cdot \|n_i^p\|} \right)
\]  

(14)

Therefore, our loss function is:

\[
L = l_d + \gamma l_k + \eta l_n
\]  

(15)

where \(\gamma, \eta \in \mathbb{R}\). In these experiments, the weights \(\gamma\) of \(l_k\) and \(\eta\) of \(l_n\) are set to be \(\gamma = 1\) and \(\eta = 1\).

For \(l_d\), taking derivative of \(\omega\), the gradient takes the following form:

\[
\frac{\partial l_d}{\partial \omega} = \frac{1}{n} \sum_{i=1}^{n} \frac{\partial l_d}{\partial y_p} \frac{\partial y_p}{\partial \omega} = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{y_p - y_R} \frac{\partial y_R}{\partial \omega}
\]  

(16)

For the same depth difference, according to formula \(\frac{1}{y_p - y_R} = \frac{1}{\sqrt{\frac{x}{x^2 + a^2}}}\), it has a relatively large gradient in the distance, while the near gradient is relatively small, so the distant error causes a larger weight adjustment. In this way, it achieves the goal of focusing on long distance.

IV. EXPERIMENTS
A. DATASET

Our experiments are mainly conducted on NYU v2 [15] and KITTI [14]. The NYU v2 consists of 240k RGB-D images of indoor scenes, captured with Microsoft Kinect. We use the official train\textbackslash test split, where 249 scenes for training, and 215 scenes for testing. All the NYU v2 images are reduced to the resolution of 228 × 304 from 480 × 640. In the test phase, we use the same test set with 654 samples. The KITTI dataset includes a full suite of raw outdoor data such as stereo videos and 3D point clouds. We use 39810 frames for training, about 4K for evaluation, and 697 images from the test split [16]. All the KITTI images are reduced to the resolution of 192 × 640 from 375 × 1241. Data enhancement such as horizontal flip, random rotation, center cropping, and data normalization are used to avoid overfitting and improve the generalization ability of the model.

B. IMPLEMENTATION DETAILS

We implement our proposed depth estimation network on the public deep learning platform pytorch [40] and train it on the 16GB Tesla V100, with a batch size of 8. The encoder of the entire network is initialized by ILSVRC [41], and the rest of the network are randomly initialized by Gaussian distribution. We adopt Adam optimizer with an initial learning rate of \(1e^{-4}\), \(\beta_1\), \(\beta_2\) and weight decay are set to be 0.9, 0.999 and 0.0001 respectively. Scale parameters \(\alpha\) and \(\beta\) are initialized as 0 and gradually updated as the model learns. We train the proposed model for ten epochs. When the model is trained to the fifth epochs, the performance tends to be stable and no longer increases.

C. EVALUATION METRICS

Following previous work [16], we use the following metrics to evaluate our method:

- Root mean squared error(RMS):
\[
\sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_R - y_p)^2}
\]

- Mean relative error(REL):
\[
\frac{1}{N} \sum_{i=1}^{N} \frac{y_R - y_p}{y_R}
\]

- Mean log10 error(log10):
\[
\frac{1}{N} \sum_{i=1}^{N} \left\| \log_{10} y_R - \log_{10} y_p \right\|
\]

- Thresholded accuracy(\(\delta\)): Percentage of \(d_i\), such that
\[
\max \left( \frac{y_p}{y_R}, \frac{y_R}{y_p}, \frac{y_R}{y_p}, \frac{y_p}{y_R} \right) = \delta < \text{thr}, \quad \text{thr} = 1.25, 1.25^2, 1.25^3
\]

D. COMPARISONS WITH THE STATE-OF-THE-ART

In this subsection, we evaluate our methods from both qualitative and quantitative points of views on the NYU V2 and the KITTI datasets. Results show that our proposed framework achieves state-of-the-art performance.

In TABLE 1, we compare the proposed method with the different previous state-of-the-art using the mentioned evaluation metrics on NYU v2 depth dataset. It is observed that
our model obtains the same score as [11] on rel, $\delta_2$ and $\delta_3$ metrics, and provides a second performance 0.519 on rms, only 0.01 less than the result in [13]. However, this difference is caused by different iterations and training samples. We use 50 K training data for 10 epochs. Other metrics results show that our method achieves significant gains over all existing state-of-the-art approaches.

Qualitative results are illustrated in FIGURE 4. We show some samples comparison between our method and previous state-of-the-art on NYU V2 dataset. It is observed that [10], [22] only predict the approximate contour of an object. For example, the computer on the table in (b) and the left four chairs in (a) cannot be accurately predicted. Although [11], [13] can produce relatively clear details and edge at close range, they focus on the boundary of a closer object such as the floor in (d), and weaken the local details, such as the table in (e). Our method not only considers the dense intermediate features but also focuses on long distance features from the image. Therefore, the predicted depth maps provide excellent edge and local details both in the near (e) and far (c).

In Table 2, we also evaluate our method’s performance on KITTI dataset. Following the evaluation protocol in [26], we only evaluate in central images where depth is less than 80 meters. Our result outperforms most existing state-of-the-art methods except DORN. We suspect that one reason is sparse 3D point clouds in KITTI raw data. The distance-aware loss does not converge well in sparse depth map. Although qualitative results might not be the best, the quality of the proposed method is much better than state-of-the-art as shown in FIGURE 5.

E. ABLATION STUDIES
In order to further analyze the proposed loss function, dense decoding module, and CSAM, we conduct three sets of
TABLE 2. Evaluation results of depth estimation on the KITTI test set. The methods trained on KITTI raw dataset are denoted by K, virtual KITTI dataset are denoted by vK, M, S and D* denote monocular video, stereo supervision and auxiliary depth supervision, respectively. D means depth supervision. The best results in each category are in bold, and the second best are underlined.

| Methods          | Dataset | Rel ↓ | Sq Rel ↓ | RMSE  | RMSE log | \( \delta_1 \) ↑ | \( \delta_2 \) ↑ | \( \delta_3 \) ↑ |
|------------------|---------|-------|----------|--------|----------|----------------|----------------|----------------|
| Zhou et al. [26] | K (M)   | 0.208 | 1.768    | 6.856  | 0.283    | 0.678          | 0.885          | 0.957          |
| Mahjourian et al. [48] | K (M)  | 0.163 | 1.240    | 6.220  | 0.250    | 0.762          | 0.916          | 0.968          |
| LEGO [49]        | K (M)   | 0.154 | 1.272    | 6.012  | 0.230    | -              | -              | -              |
| Monodepth2 [28] | K (M)   | 0.115 | 0.903    | 4.863  | 0.193    | 0.877          | 0.959          | 0.981          |
| Rian et al. [27] | K (M)   | 0.137 | 1.089    | 5.439  | 0.217    | 0.830          | 0.942          | 0.975          |
| SynDeMo [50]     | K+vK (MΔ*) | 0.116 | 0.746    | 4.627  | 0.194    | 0.858          | 0.952          | 0.977          |
| GLNet [31]       | K (M)   | 0.135 | 1.070    | 5.230  | 0.210    | 0.841          | 0.948          | 0.980          |
| Zhou et al. [51] | K (M)   | 0.121 | 0.837    | 4.945  | 0.197    | 0.853          | 0.955          | 0.982          |
| Patil et al. [52] | K (M)    | 0.111 | 0.821    | 4.650  | 0.187    | 0.883          | 0.961          | 0.982          |
| Eigen et al. [16] | K (D) | 0.203 | 1.548    | 6.307  | 0.282    | 0.702          | 0.890          | 0.958          |
| Liu et al. [18]  | K (D)   | 0.202 | 1.614    | 6.523  | 0.275    | 0.678          | 0.895          | 0.965          |
| Garg et al. [17] | K (S)   | 0.152 | 1.226    | 5.849  | 0.246    | 0.784          | 0.921          | 0.967          |
| Godard et al. [20] | K (S) | 0.124 | 1.076    | 5.311  | 0.219    | 0.847          | 0.942          | 0.973          |
| DUSO [53]        | K (DS)  | 0.097 | 0.734    | 4.442  | 0.187    | 0.888          | 0.958          | 0.980          |
| DORN [13]        | K (D)   | 0.072 | 0.307    | 2.727  | 0.120    | 0.932          | 0.984          | 0.994          |
| **Ours**         | K (D)   | 0.096 | 0.655    | 4.327  | 0.171    | 0.893          | 0.963          | 0.983          |

FIGURE 5. Qualitative results on the KITTI Eigen split. From top to bottom, the images are input, ground truth, results of Monodepth [20], SfmLearner [26], Monodepth2 [28], SC-SfmLearner [27], DORN [13] and our method. Our method provides clearer local details on the KITTI dataset.

ablation experiments on NYU Depth V2. Results are shown in TABLE 3.

1) LOSS FUNCTION
In this ablation experiment, the encoder is SENet, and the decoder are CSAM and DDM. We use three loss functions \( L_0, L_1, L_2 \), their \( e_t \) are \( e_t^0, e_t^1, e_t^2 \):

\[
e_t^0 = \| y_t^g - y_t^p \|_1
\]  
\[
e_t^1 = (y_t^g + \lambda) \| y_t^g - y_t^p \|_1
\]  
\[
e_t^2 = (y_t^2 + \lambda) \| y_t^2 - y_t^p \|_1
\]  

\( L_0 \) pays attention to the nearby points and less attention to distant points, while \( L_1 \) and \( L_2 \) are opposite to \( L_0 \). From TABLE 3, \( L_1 \) and \( L_2 \) are better than \( L_0 \). The difference between \( L_1 \) and \( L_2 \) is that \( y \) is used in \( L_1 \), and \( y^2 \) is used in \( L_2 \). The fuzziness of the predicted map does not increase linearly from nearby points to distant points. Compared with \( L_1, L_2 \)}
uses nonlinear weights from near to far to punish this non-linear fuzziness. Looking at the white boxes in FIGURE 6, the prediction of the $L_2$ in the distance is closer to ground-truth, while the prediction of the $L_0$ is inaccurate or even unpredictable.

### 2) DENSE DECODING MODULE

We conduct the ablation experiments to reveal the effectiveness of incorporating the DDM into the proposed modules. We control the same conditions and compare the effects of up-projection and dense decoding module. The experimental results are shown in TABLE 3. The results demonstrate that the DDM provides benefits in different perspectives. Analyze the decoder, instead of putting all input features of the AsppBlock$_j$ ($j<i$) through multiple interpolations and then adding them up as input to AsppBlock$_i$, we upsample the already upsampled features. Compared with the former, this method uses upsamplings four times during decoding, while the former may use ten times ($4+3+2+1$). Since upsampling consists of up-projection, this design significantly reduces the number of parameters. In addition, DDM has four more AsppBlocks than the baseline, which uses only four up-projections. The number of parameters in AsppBlock$_i$ is $C \times 1 \times 1 \times C + C \times 3 \times 3 \times \frac{C}{2}$, the parameters of the 4 AsppBlocks total about 1.9M.

In addition, we find that after the first epoch of training, the $\delta_1$ is as high as 82.4%. Compared with the ordinary decoding layer [11], dense decoding layers make full use of different intermediate features in the process of decoding. These features contain different scale ranges and achieve greater receptive fields while acquiring multi-scale information. At each iteration, the multi-scale information of different receptive fields is reused as much as possible to accelerate the convergence speed of the model. Therefore, the network can obtain relatively good results at the beginning of training.

### 3) CHANNEL-SPATIAL ATTENTION MODULE

We compare two sets of experiments with and without the CSAM. As shown in TABLE 3, the CSAM significantly improves performance. Looking at the red and green boxes of the first row in FIGURE 7, the CSAM can predict richer local details, such as table corners. The predicted map without the attention module may be blurred. Besides, observing the red box in the second row (b), the chair and the floor in the red box have almost the same color, that is, they have the same distance. While the distance is different in the color map (a). It may be that the ground truth has lost information during acquisition or data preprocessing. The CSAM can capture the relationship dependence between any pixel points from the color map. There are strong dependencies on chair internal pixels. Another dependencies between the chair and the ground make the chair appear darker in the predicted depth map, and the chair is different from the floor.

### 4) FEATURE VISUALIZATION

In order to further verify the benefits of the two modules and the loss function, we visualize the intermediate feature maps generated by the network as shown in FIGURE 8. It is observed that the baseline represents a low level of informational context and distant scenes, while our loss function

![FIGURE 6. Results of different loss functions for two images. From (a) to (d): input images, ground-truth depth maps, with $L_0$, with $L_2$.](image1)

![FIGURE 7. Results of different decoding modules for two images. From (a) to (d): input images, ground-truth depth maps, without CSAM, with CSAM.](image2)

![FIGURE 8. Feature visualization, the visualization result is obtained by superimposing the feature map and the color map. From (a) to (e): input, baseline, with DDM and $L_2$, with CSAM and $L_2$, with CSAM, DDM and $L_2$.](image3)
gets larger areas of interest in the distance. The network that does not use the CSAM as the transition layer has a weak ability to discriminate the depth features, although the dense decoding module is used to enhance the reuse of features. As the dependence between these features is inferior, it has similar features in the far and near, which results in dispersed features ranges in (c), (d) and (e) use the attention module to characterize objects of different depths and locations by modeling the relationship between any point and global features. Compared with (d), (e) reuses the attention features, so the depth layers are stronger.

5) TIMING ANALYSIS
In FIGURE 9, we compare the computational cost of the proposed modules during test phase. The CSAM and DDM do not significantly increase computing resources. Our approach outperforms Xu et al. [9], Hu et al. [11] both in terms of accuracy and running time. Furthermore, in comparison to Eigen et al. [16], Laina et al. [10] and Hao et al. [21], the proposed method provides a trade-off between accuracy and speed.

![Figure 9](image.png)

**Figure 9.** Accuracy and Speed Comparison. Comparison with previous methods: δ vs. running time.

### V. CONCLUSION
To solve the local depth details loss caused by convolution stacking, a novel encoder-decoder attention-dense decoding network is proposed. Our main idea is to take advantage of the channel-spatial attention module, which captures the dependence between different channels and spatial locations by self-attentions. Furthermore, a dense decoding module is introduced to capture more massive and denser attention features. Moreover, in order to solve the fuzzy phenomenon in the distant scene, we further propose a distance-aware loss function that pays more attention to the long-distance objects. Extensive experimental results demonstrate that our method outperforms state-of-the-art performance on the KITTI/NYU V2 dataset. For the future work, it is worthwhile to explore a more accurate unsupervised depth estimation based on these modules.

### VI. ACKNOWLEDGMENT
(Jianrong Wang and Ge Zhang contributed equally to this work.)

### REFERENCES
[1] C. Tang, C. Hou, and Z. Song, “Depth recovery and refinement from a single image using defocus cues,” J. Modern Opt., vol. 62, no. 6, pp. 441–448, 2015.
[2] J. Han, L. Shao, D. Xu, and J. Shotton, “Enhanced computer vision with microsoft kinect sensor: A review,” IEEE Trans. Cybern., vol. 43, no. 5, pp. 1318–1334, Oct. 2013.
[3] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, “Residual attention network for image classification,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 6450–6458.
[4] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, “Learning a discriminative feature network for semantic segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 1857–1866.
[5] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[6] S. Zhao, L. Zhang, Y. Shen, S. Zhao, and H. Zhang, “Super-resolution for monocular depth estimation with multi-scale sub-pixel convolutions and a smoothness constraint,” IEEE Access, vol. 7, pp. 16323–16335, 2019.
[7] J. Zhang, Q. Su, P. Liu, C. Xu, and Y. Chen, “Unsupervised learning of monocular depth and Large-Ego-Motion with multiple loop consistency losses,” IEEE Access, vol. 7, pp. 7793–7804, 2019.
[8] M. Song and W. Kim, “Depth estimation from a single image using guided deep network,” IEEE Access, vol. 7, pp. 142595–142606, 2019.
[9] D. Xu, W. Wang, H. Tang, H. Liu, N. Sebe, and E. Ricci, “Structured attention guided convolutional neural fields for monocular depth estimation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Salt Lake City, UT, USA, Jun. 2018, pp. 3917–3925.
[10] J. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, “Deeper depth prediction with fully convolutional residual networks,” in Proc. 4th Int. Conf. 3D Vis. (3DV), Stanford, CA, USA, Oct. 2016, pp. 239–248.
[11] J. Hu, M. Ozay, Y. Zhang, and T. Okatani, “Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries,” in Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV), Jan. 2019, pp. 1043–1051.
[12] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in Proc. 15th Eur. Conf. Comput. Vis. (ECCV), Munich, Germany, Sep. 2018, pp. 833–851.
[13] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, “Deep ordinal regression network for monocular depth estimation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 2002–2011.
[14] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? The KITTI vision benchmark suite,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2012, pp. 3354–3361.
[15] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, “Indoor segmentation and support inference from RGBD images,” in Proc. 12th Eur. Conf. Comput. Vis. (ECCV), Florence, Italy, Oct. 2012, pp. 746–760.
[16] D. Eigen, C. Puhrsch, and R. Fergus, “Depth map prediction from a single image using a multi-scale deep network,” in Proc. Adv. Neural Inf. Process. Syst., Montreal, QC, Canada, Dec. 2014, pp. 2366–2374.
[17] R. Garg, B. G. V. Kumar, G. Carneiro, and I. Reid, “Unsupervised CNN for single view depth estimation: Geometry to the rescue,” in Proc. 14th Eur. Conf. Comput. Vis. (ECCV), Amsterdam, The Netherlands, Oct. 2016, pp. 740–756.
[18] P. Liu, C. Shen, G. Lin, and I. Reid, “Learning depth from single monocular images using deep convolutional neural fields,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 10, pp. 2024–2039, Oct. 2016.
[19] D. Xu, E. Ricci, W. Ouyang, X. Wang, and N. Sebe, “Multi-scale continuous CRFs as sequential deep networks for monocular depth estimation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Honolulu, HI, USA, Jul. 2017, pp. 161–169.
[20] C. Godard, O. M. Aodha, and G. J. Brostow, “Unsupervised monocular depth estimation with left-right consistency,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Honolulu, HI, USA, Jul. 2017, pp. 6602–6611.
[21] Z. Hao, Y. Li, S. You, and F. Lu, “Detail preserving depth estimation from a single image using attention guided networks,” in Proc. Int. Conf. 3D Vis. (3DV), Verona, Italy, Sep. 2018, pp. 304–313.
[22] D. Eigen and R. Fergus, “Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 2650–2658.
J. Li, R. Klein, and A. Yao, “Single-image depth estimation based on Fourier domain analysis,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 330–339.

Y. Chen, H. Zhao, and Z. Hu, “Attention-based context aggregation network for monocular depth estimation,” 2019, arXiv:1901.10137. [Online]. Available: http://arxiv.org/abs/1901.10137

T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, “Unsupervised learning of depth and ego-motion from video,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 6612–6619.

J.-W. Bian, Z. Li, N. Wang, H. Zhan, C. Shen, M.-M. Cheng, and I. Reid, “Unsupervised scale-consistent depth and ego-motion learning from monocular video,” in Proc. NIPS, 2019, pp. 35–45.

C. Godard, O. M. Aodha, M. Firman, and G. Brostow, “Digging into self-supervised monocular depth estimation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 3828–3838.

A. Ranjan, V. Jampani, L. Balles, K. Kim, D. Sun, J. Wulff, and M. J. Black, “Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 12232–12241.

Y. Zou, Z. Luo, and J.-B. Huang, “DF-net: Unsupervised joint learning of depth and flow using cross-task consistency,” in Proc. ECCV, vol. 11209, 2018, pp. 38–55.

Y. Chen, C. Schmid, and C. Sminchisescu, “Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and cameras,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 7063–7072.

Z. Yin and J. Shi, “GeoNet: Unsupervised learning of dense depth, optical flow and camera pose,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 1983–1992.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in Proc. Adv. Neural Inf. Process. Syst., Long Beach, CA, USA, Dec. 2017, pp. 5998–6008.

M. Zhao, X. Xiang, R. Zhang, N. Lv, and A. El Saddik, “Optical flow estimation using dual self-attention pyramid networks,” IEEE Trans. Circuits Syst. Video Technol., early access, Sep. 23, 2019, doi: 10.1109/TCVSVT.2019.2943140.

D. M. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in Proc. 13th Eur. Conf. Comput. Vis. (ECCV), Zürich, Switzerland, Sep. 2014, pp. 818–833.

J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, “Squeeze-and-excitation networks,” IEEE Trans. Pattern Anal. Mach. Intell., early access, Apr. 29, 2019, doi: 10.1109/TPAMI.2019.2913372.

S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, “CBAM: Convolutional block attention module,” in Proc. 15th Eur. Conf. Comput. Vis. (ECCV), Munich, Germany, Sep. 2018, pp. 3–19.

M. Zhai, X. Xiang, R. Zhang, N. Lv, and A. E. Saddik, “Optical flow estimation using channel attention mechanism and dilated convolutional neural networks,” Neurocomputing, vol. 368, pp. 124–132, Nov. 2019.

G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 2261–2269.

A. Paszke et al., “Automatic differentiation in PyTorch,” in Proc. NIPS-W, 2017.

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet large scale visual recognition challenge,” Int. J. Comput. Vis., vol. 115, no. 3, pp. 211–252, Dec. 2015.

F. Liu, C. Shen, and G. Lin, “Deep convolutional neural fields for depth estimation from a single image,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 5162–5170.

Y. Cao, Z. Wu, and C. Shen, “Estimating depth from monocular images as classification using deep fully convolutional residual networks,” IEEE Trans. Circuits Syst. Video Technol., vol. 28, no. 11, pp. 3174–3182, Nov. 2018.

J. Li, R. Klein, and A. Yao, “A two-streamed network for estimating fine-scaled depth maps from single RGB images,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 3392–3400.

F. Ma and S. Karaman, “Sparse-to-dense: Depth prediction from sparse depth samples and a single image,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), May 2018, pp. 4796–4803.

X. Qi, R. Liao, Z. Liu, R. Urtasun, and J. Jia, “GeoNet: Geometric neural network for joint depth and surface normal estimation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 283–291.

H. Zhang, Y. Li, Y. Cao, Y. Liu, C. Shen, and Y. Yan, “Exploiting temporal consistency for real-time video depth estimation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 1725–1734.

R. Mahjourian, M. Wicke, and A. Angelova, “Unsupervised learning of depth and ego-motion from monocular video using 3D geometric constraints,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 5667–5675.

Z. Yang, P. Wang, Y. Wang, W. Xu, and R. Nevatia, “LEGO: Learning edge with geometry all at once by watching videos,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Salt Lake City, UT, USA, Jun. 2018, pp. 225–234.

B. Zhong and M. S. Rad, D. Mahapatra, and J.-P. Thiran, “SynDeMo: Synergistic deep feature alignment for joint learning of depth and ego-motion,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 4210–4219.

J. Zhou, Y. Wang, K. Qin, and W. Zeng, “Unsupervised high-resolution depth learning from videos with dual networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), May 2019, pp. 6872–6881.

V. Patil, W. V. Gansbeke, D. Dai, and L. V. Gool, “Don’t forget the past: Recurrent depth estimation from monocular video,” CoRR, abs/2001.02613, 2020. [Online]. Available: https://arxiv.org/abs/2001.02613.

N. Yang, R. Wang, J. Stückler, and D. Cremers, “Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry,” in Proc. ECCV, vol. 11212, 2018, pp. 835–852.

JIANRONG WANG was born in China, in 1976. He received the master’s degree and the master’s and Ph.D. degrees in computer application technology from Tianjin University, in 1998, 2004, and 2009, respectively. From 1998 to 2004, he was an Assistant Professor with the School of Computing, Tianjin University. Since 2009, he has been an Associate Professor with the School of Computer Science, Tianjin University. He has won many national and regional awards.

GE ZHANG received the B.S. degree in computer science from Tianjin Polytechnic University, in 2018, where he is currently pursuing the master’s degree in computer technology from Tianjin University. His research interests include 3D pose estimation, unsupervised depth estimation, scene understanding, and deep learning.

TIANYI XIU received the B.S. degree in automation and the M.S. degree in computer science and technology from Tianjin University, in 2012 and in 2015, respectively, where he is currently pursuing the Ph.D. degree. His research interests include data mining, the IoT, and blockchain.
MEI YU is currently an Associate Professor with the College of Intelligence and Computing, Tianjin University, China. Besides, she also serves as the Instructor of the IT innovation and entrepreneurship training base of Tianjin University, where she is responsible for the construction of the base. Her research interests include natural language processing and knowledge mapping.

TAO LUO received the M.S. degree from the School of Precision Instrument and Opto-Electronics, in 2006, and the Ph.D. degree from the School of Electronic Information Engineering, Tianjin University, in 2009. He is currently an Associate Professor with the College of Intelligence and Computing, Tianjin University. His research interests include intelligent vision processing, image sensors, and integrated circuits.