

# FADNet: A Fast and Accurate Network for Disparity Estimation

Qiang Wang<sup>1,\*</sup>, Shaohuai Shi<sup>1,\*</sup>, Shizhen Zheng<sup>1</sup>, Kaiyong Zhao<sup>1,†</sup>, Xiaowen Chu<sup>1,†</sup>

**Abstract**—Deep neural networks (DNNs) have achieved great success in the area of computer vision. The disparity estimation problem tends to be addressed by DNNs which achieve much better prediction accuracy in stereo matching than traditional hand-crafted feature based methods. On one hand, however, the designed DNNs require significant memory and computation resources to accurately predict the disparity, especially for those 3D convolution based networks, which makes it difficult for deployment in real-time applications. On the other hand, existing computation-efficient networks lack expression capability in large-scale datasets so that they cannot make an accurate prediction in many scenarios. To this end, we propose an efficient and accurate deep network for disparity estimation named FADNet with three main features: 1) It exploits efficient 2D based correlation layers with stacked blocks to preserve fast computation; 2) It combines the residual structures to make the deeper model easier to learn; 3) It contains multi-scale predictions so as to exploit a multi-scale weight scheduling training technique to improve the accuracy. We conduct experiments to demonstrate the effectiveness of FADNet on two popular datasets, Scene Flow and KITTI 2015. Experimental results show that FADNet achieves state-of-the-art prediction accuracy, and runs at a significant order of magnitude faster speed than existing 3D models. The codes of FADNet are available at <https://github.com/HKBU-HPML/FADNet>.

## I. INTRODUCTION

It has been seen that deep learning has been widely deployed in many computer vision tasks. Disparity estimation (also referred to as stereo matching) is a classical and important problem in computer vision applications, such as 3D scene reconstruction, robotics and autonomous driving. While traditional methods based on hand-crafted feature extraction and matching cost aggregation such as Semi-Global Matching (SGM) [1] tend to fail on those textureless and repetitive regions in the images, recent advanced deep neural network (DNN) techniques surpass them with decent generalization and robustness to those challenging patches, and achieve state-of-the-art performance in many public datasets [2][3][4][5][6][7]. The DNN-based methods for disparity estimation are end-to-end frameworks which take stereo images (left and right) as input to the neural network and predict the disparity directly. The architectures of DNN are very essential to achieve accurate estimation, and can be categorized into two classes, encoder-decoder network with 2D convolution (ED-Conv2D) and cost volume matching with 3D convolution (CVM-Conv3D). Besides, recent studies [8], [9] begin to reveal the potential of automated machine

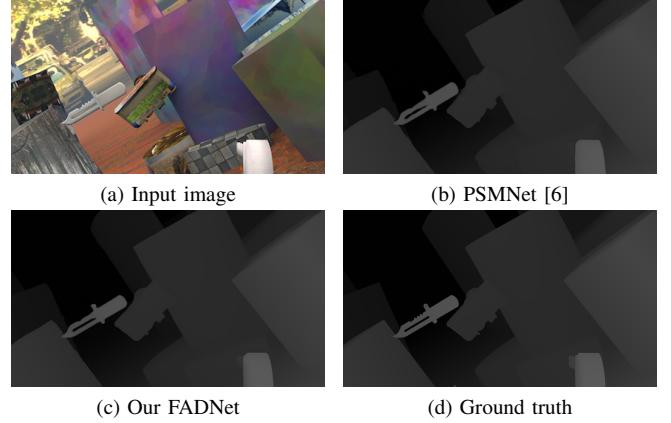


Fig. 1: Performance illustrations. (a) a challenging input image. (b) Result of PSMNet [6] which consumes 13.99 GB GPU memory and runs 399.3 ms for one stereo image pair on an Nvidia Tesla V100 GPU. (c) Result of our FADNet, which only consumes 1.62 GB GPU memory and runs 18.7 ms for one stereo image pair on the Nvidia Tesla V100 GPU.

learning (AutoML) for neural architecture search (NAS) on stereo matching. In practice, to measure whether a DNN model is good enough, we not only need to evaluate its accuracy on unseen stereo images (whether it can estimate the disparity correctly), but also need to evaluate its time efficiency (whether it can generate the results in real-time).

In ED-Conv2D methods, stereo matching neural networks [2][3][5] are first proposed for end-to-end disparity estimation by exploiting an encoder-decoder structure. The encoder part extracts the features from the input images, and the decoder part predicts the disparity with the generated features. The disparity prediction is optimized as a regression or classification problem using large-scale datasets (e.g., Scene Flow [5]) with disparity ground truth. The correlation layer [10][5] is then proposed to increase the learning capability of DNNs in disparity estimation, and it has been proved to be successful in learning strong features at multiple levels of scales [10][5][11][12][13]. To further improve the capability of the models, residual networks [14][15][16] are introduced into the architecture of disparity estimation networks since the residual structure enables much deeper network to be easier to train [17]. The ED-Conv2D methods have been proved computing efficient, but they cannot achieve very high estimation accuracy.

To address the accuracy problem of disparity estimation, researchers have proposed CVM-Conv3D networks to better capture the features of stereo images and thus improve the estimation accuracy [3][18][6][7][19]. The key idea of the

\*Authors have contributed equally.

†Corresponding authors.

<sup>1</sup>Department of Computer Science, Hong Kong Baptist University,  
{qiangwang, csshshi, szzheng, kyzhao, chxw}  
@comp.hkbu.edu.hk

CVM-Conv3D methods is to generate the cost volume by concatenating left feature maps with their corresponding right counterparts across each disparity level [18][6]. The features of cost volume are then automatically extracted by 3D convolution layers. However 3D operations in DNNs are computing-intensive and hence very slow even with current powerful AI accelerators (e.g., GPUs). Although the 3D convolution based DNNs can achieve state-of-the-art disparity estimation accuracy, they are difficult for deployment due to their resource requirements. On one hand, it requires a large amount of memory to install the model; so only a limited set of accelerators (like Nvidia Tesla V100 with 32GB memory) can run these models. On the other hand, it takes several seconds to generate a single result even on the very powerful Tesla V100 GPU using CVM-Conv3D models. The memory consumption and the inefficient computation make the CVM-Conv3D methods difficult to be deployed in practice. Therefore, it is crucial to address the accuracy and efficiency problems for real-world applications.

To this end, we propose FADNet which is a Fast and Accurate Disparity estimation Network based on ED-Conv2D architectures. FADNet can achieve high accuracy while keeping a fast inference speed. As illustrated in Fig. 1, our FADNet can easily obtain comparable performance as state-of-the-art PSMNet [6], while it runs approximately  $20\times$  faster than PSMNet and consumes  $10\times$  less GPU memory. In FADNet, we first exploit the multiple stacked 2D-based convolution layers with fast computation, and then we combine state-of-the-art residual architectures to improve the learning capability, and finally we introduce multi-scale outputs for FADNet so that it can exploit the multi-scale weight scheduling to improve the training speed. These features enable FADNet to efficiently predict the disparity with high accuracy as compared to existing work. Our contributions are summarized as follows:

- We propose an accurate yet efficient DNN architecture for disparity estimation named FADNet, which achieves comparable prediction accuracy as CVM-Conv3D models and it runs at an order of magnitude faster speed than the 3D-based models.
- We develop a multiple rounds training scheme with multi-scale weight scheduling for FADNet during training, which improves the training speed yet maintains the model accuracy.
- We achieve state-of-the-art accuracy on the Scene Flow dataset with up to  $20\times$  and  $45\times$  faster disparity prediction speed than PSMNet [6] and GANet [7] respectively.

The rest of the paper is organized as follows. We introduce some related work in DNN based stereo matching problems in Section II. Section III introduces the methodology and implementation of our proposed network. We demonstrate our experimental results in Section IV. We finally conclude the paper in Section V.

## II. RELATED WORK

There exist many studies using deep learning methods in estimating image depth using monocular, stereo and

multi-view images. Although monocular vision is low cost and commonly available in practice, it does not explicitly introduce any geometrical constraint, which is important for disparity estimation[20]. On the contrary, stereo vision leverages the advantages of cross-reference between the left and the right view, and usually show greater performance and robustness in geometrical tasks. In this paper, we mainly discuss the work related to stereo images for disparity estimation, which is classified into two categories: 2D based and 3D based CNNs.

In 2D based CNNs, end-to-end architectures with mainly convolution layers [5][21] are proposed for disparity estimation, which use two stereo images as input and generate the disparity directly and the disparity is optimized as a regression task. However, the models are pure 2D CNN architectures which are difficult to capture the matching features such that the estimation results are not good. To address the problem, the correlation layer which can express the relationship between left and right images is introduced in the end-to-end architecture (e.g., DispNetCorr1D [5], FlowNet [10], FlowNet2 [22], DenseMapNet [23]). The correlation layer significantly increases the estimating performance compared to the pure CNNs, but existing architectures are still not accurate enough for production.

3D based CNNs are further proposed to increase the estimation performance [3][18][6][7][19], which employ 3D convolutions with cost volume. The cost volume is mainly formed by concatenating left feature maps with their corresponding right counterparts across each disparity level [18][6], and the features of the generated cost volumes can be learned by 3D convolution layers. The 3D based CNNs can automatically learn to regularize the cost volume, which have achieved state-of-the-art accuracy of various datasets. However, the key limitation of the 3D based CNNs is their high computation resource requirements. For example, training GANet [7] with the Scene Flow [5] dataset takes weeks even using very powerful Nvidia Tesla V100 GPUs. Even they achieve good accuracy, it is difficult to deploy due to their very low time efficiency. To this end, we propose a fast and accurate DNN model for disparity estimation.

## III. MODEL DESIGN AND IMPLEMENTATION

Our proposed FADNet exploits the structure of DispNetC [5] as a backbone, but it is extensively reformed to take care of both accuracy and inference speed, which is lacking in existing studies. We first change the structure in terms of branch depth and layer type by introducing two new modules, residual block and point-wise correlation. Then we exploit the multi-scale residual learning strategy for training the refinement network. Finally, a loss weight training schedule is used to train the network in a coarse-to-fine manner.

### A. Residual Block and Point-wise Correlation

DispNetC and DispNetS which are both from the study in [5] basically use an encoder-decoder structure equipped with five feature extraction and down-sampling layers and five feature deconvolution layers. While conducting feature

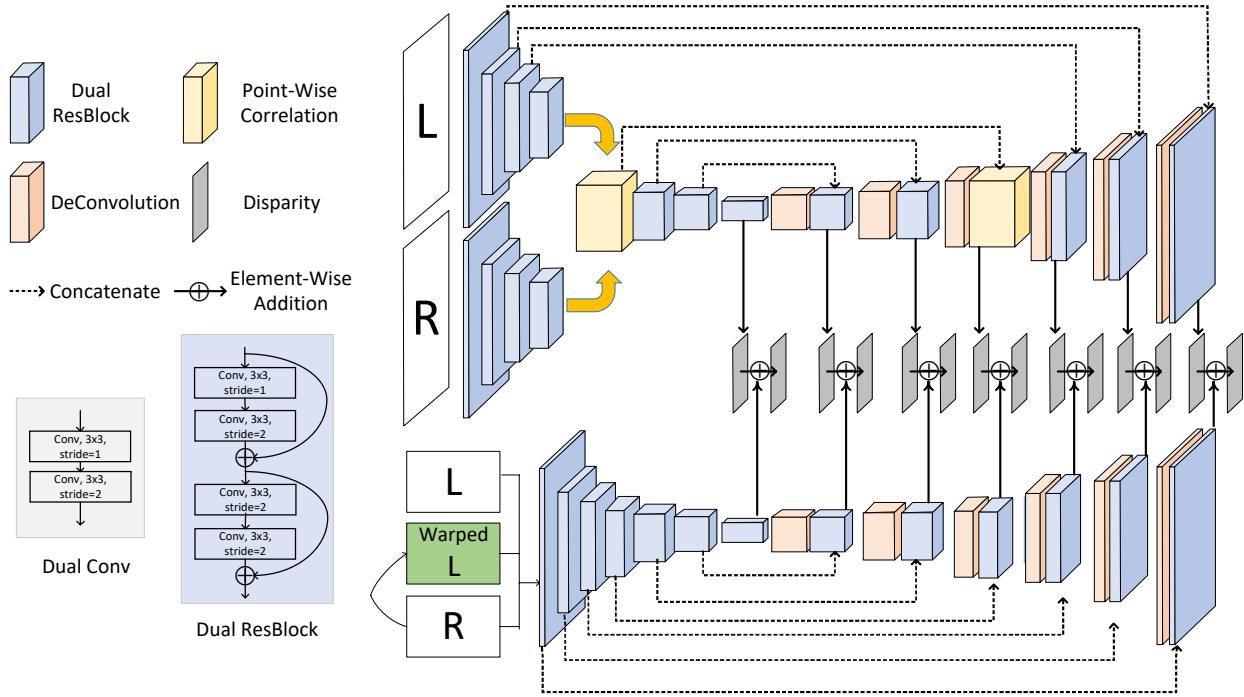


Fig. 2: The model structure of our proposed FADNet.

extraction and down-sampling, DispNetC and DispNetS first adopt a convolution layer with a stride of 1 and then a convolution layer with a stride of 2 so that they consistently shrink the feature map size by half. We call the two-layer convolutions with size reduction as Dual-Conv, which is shown in the left-bottom corner of Fig. 2. DispNetC equipped with Dual-Conv modules and a correlation layer finally achieves an end-points error (EPE) of 1.68 on the Scene Flow dataset, as reported in [5].

The residual block originally derived in [14] for image classification tasks is widely used to learn robust features and train a very deep networks. The residual block can well address the gradient vanish problem when training very deep networks. Thus, we replace the convolution layer in the Dual-Conv module by the residual block to construct a new module called Dual-ResBlock, which is shown in the left-bottom corner of Fig. 2. With Dual-ResBlock, we can make the network deeper without training difficulty as the residual block allows us to train very deep models. Therefore, we further increase the number of feature extraction and down-sampling layers from five to seven. Finally, DispNetC and DispNetS are evolving to two new networks with better learning ability, which are called RB-NetC and RB-NetS respectively, as shown in Fig. 2.

One of the most important contributions of DispNetC is the correlation layer, which targets at finding correspondences between the left and right images. Given two multi-channel feature maps  $\mathbf{f}_1, \mathbf{f}_2$  with  $w, h$  and  $c$  as their width, height and number of channels, the correlation layer calculates the cost volume of them using Eq. (1).

$$c(\mathbf{x}_1, \mathbf{x}_2) = \sum_{\mathbf{o} \in [-k, k] \times [-k, k]} \langle \mathbf{f}_1(\mathbf{x}_1 + \mathbf{o}), \mathbf{f}_2(\mathbf{x}_2 + \mathbf{o}) \rangle, \quad (1)$$

where  $k$  is the kernel size of cost matching,  $\mathbf{x}_1$  and  $\mathbf{x}_2$  are the centers of two patches from  $\mathbf{f}_1$  and  $\mathbf{f}_2$  respectively. Computing all patch combinations involves  $c \times K^2 \times w^2 \times h^2$  multiplication and produces a cost matching map of  $w \times h$ . Given a maximum searching range  $D$ , we fix  $\mathbf{x}_1$  and shift the  $\mathbf{x}_2$  on the x-axis direction from  $-D$  to  $D$  with a stride of two. Thus, the final output cost volume size will be  $w \times h \times D$ .

However, the correlation operation assumes that each pixel in the patch contributes equally to the point-wise convolution results, which may loss the ability to learn more complicated matching patterns. Here we propose point-wise correlation composed of two modules. The first module is a classical convolution layer with a kernel size of  $3 \times 3$  and a stride of 1. The second one is an element-wise multiplication which is defined by Eq. (2).

$$c(\mathbf{x}_1, \mathbf{x}_2) = \sum \langle \mathbf{f}_1(\mathbf{x}_1), \mathbf{f}_2(\mathbf{x}_2) \rangle, \quad (2)$$

where we remove the patch convolution manner from Eq. (1). Since the maximum valid disparity is 192 in the evaluated datasets, the maximum search range for the original image resolution is no more than 192. Remember that the correlation layer is put after the third Dual-ResBlock, of which the output feature resolution is  $1/8$ . So a proper searching range value should not be less than  $192/8=16$ . We set a marginally larger value 20. We also test some other values, such as 10 and 40, which do not surpass the version of using 20 in the network. The reason is that applying too small or large search range value may lead to under-fitting or over-fitting.

Table I lists the accuracy improvement brought by applying the proposed Dual-ResBlock and point-wise correlation. We train them using the same dataset as well as the training schemes. It is observed that RB-NetC outperforms DispNetC with a much lower EPE, which indicates the effectiveness of

the residual structure. We also notice that setting a proper searching range value of the correlation layer helps further improve the model accuracy.

TABLE I: Model accuracy improvement of Dual-ResBlock and point-wise correlation with different  $D$ .

Model	$D$	Training EPE	Test EPE
DispNetC	20	2.89	2.80
RB-NetC	10	2.28	2.06
RB-NetC	20	2.09	1.76
RB-NetC	40	2.12	1.83

### B. Multi-Scale Residual Learning

Instead of directly stacking DispNetC and DispNetS sub-networks to conduct disparity refinement procedure [12], we apply the multi-scale residual learning firstly proposed by [24]. The basic idea is that the second refinement network learns the disparity residuals and accumulates them into the initial results generated by the first network, instead of directly predicting the whole disparity map. In this way, the second network only needs to focus on learning the highly nonlinear residual, which is effective to avoid gradient vanishing. Our final FADNet is formed by stacking RB-NetC and RB-NetS with multi-scale residual learning, which is shown in Fig. 2.

As illustrated in Fig. 2, the upper RB-NetC takes the left and right images as input and produces disparity maps at a total of 7 scales, denoted by  $c_s$ , where  $s$  is from 0 to 6. The bottom RB-NetS exploits the inputs of the left image, right image, and the warped left images to predict the residuals. The generated residuals (denoted by  $r_s$ ) from RB-NetS are then accumulated to the prediction results by RB-NetC to generate the final disparity maps with multiple scales ( $s = 0, 1, \dots, 6$ ). Thus, the final disparity maps predicted by FADNet, denoted by  $\hat{d}_s$ , can be calculated by

$$\hat{d}_s = c_s + r_s, 0 \leq s \leq 6. \quad (3)$$

### C. Loss Function Design

Given a pair of stereo RGB images, our FADNet takes them as input and produces seven disparity maps at different scales. Assume that the input image size is  $H \times W$ . The dimension of the seven scales of the output disparity maps are  $H \times W$ ,  $\frac{1}{2}H \times \frac{1}{2}W$ ,  $\frac{1}{4}H \times \frac{1}{4}W$ ,  $\frac{1}{8}H \times \frac{1}{8}W$ ,  $\frac{1}{16}H \times \frac{1}{16}W$ ,  $\frac{1}{32}H \times \frac{1}{32}W$ , and  $\frac{1}{64}H \times \frac{1}{64}W$  respectively. To train FADNet in an end-to-end manner, we adopt the pixel-wise smooth L1 loss between the predicted disparity map and the ground truth using

$$L_s(d_s, \hat{d}_s) = \frac{1}{N} \sum_{i=1}^N \text{smooth}_{L_1}(d_s^i - \hat{d}_s^i), \quad (4)$$

where  $N$  is the number of pixels of the disparity map,  $d_s^i$  is the  $i^{th}$  element of  $d_s \in \mathcal{R}^N$  and

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2, & \text{if } |x| < 1 \\ |x| - 0.5, & \text{otherwise.} \end{cases} \quad (5)$$

Note that  $d_s$  is the ground truth disparity of scale  $\frac{1}{2^s}$  and  $\hat{d}_s$  is the predicted disparity of scale  $\frac{1}{2^s}$ . The loss function is separately applied in the seven scales of outputs, which generates seven loss values. The loss values are then accumulated with loss weights.

TABLE II: Multi-scale loss weight scheduling.

Round	$w_0$	$w_1$	$w_2$	$w_3$	$w_4$	$w_5$	$w_6$
1	0.32	0.16	0.08	0.04	0.02	0.01	0.005
2	0.6	0.32	0.08	0.04	0.02	0.01	0.005
3	0.8	0.16	0.04	0.02	0.01	0.005	0.0025
4	1.0	0	0	0	0	0	0

The loss weight scheduling technique which is initially proposed in [5] is useful to learn the disparity in a coarse-to-fine manner. Instead of just switching on/off the losses of different scales, we apply different non-zero weight groups for tackling different scale of disparity. Let  $w_s$  denote the weight for the loss of the scale of  $s$ . The final loss function is

$$L = \sum_{s=0}^6 w_s L_s(d_s, \hat{d}_s). \quad (6)$$

The specific setting is listed in Table II. Totally there are seven scales of predicted disparity maps. At the beginning, we assign low-value weights for those large scale disparity maps to learn the coarse features. Then we increase the loss weights of large scales to let the network gradually learn the finer features. Finally, we deactivate all the losses except the final predict one of the original input size. With different rounds of weight scheduling, the evaluation EPE is gradually increased to the final accurate performance which is shown in Table III on the Scene Flow dataset.

TABLE III: Model accuracy with different rounds of weight scheduling.

Round	# Epochs	Training EPE	Test EPE	Improvement (%)
1	20	1.85	1.57	-
2	20	1.33	1.32	18.9
3	20	1.04	0.93	41.9
4	30	0.92	0.83	12.0

Note: "Improvement" indicates the improvement of the current round of weight schedule over its previous.

Table III lists the model accuracy improvements (around 12%-41%) brought by the multiple round training of four loss weight groups. It is observed that both the training and testing EPEs are decreased smoothly and close, which indicates good generalization and advantages of our training strategy.

## IV. PERFORMANCE EVALUATION

In this section, we present the experimental results of our proposed FADNet compared to existing work (i.e., DispNetC [5], PSMNet [6], GANet [7] and DenseMapNet [23]) in terms of accuracy and time efficiency.

### A. Experimental Setup

We implement our FADNet using PyTorch<sup>1</sup>, which is one of popular deep learning frameworks, and we make the codes and experimental setups be publicly available<sup>2</sup>.

In terms of accuracy, the model is trained with Adam ( $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ ). We perform color normalization with the mean ([0.485, 0.456, 0.406]) and variation ([0.229, 0.224, 0.225]) of the ImageNet [25] dataset for data pre-processing. During training, images are randomly cropped to size  $H = 384$  and  $W = 768$ . The batch size is set to 16 for the training on four Nvidia Titan X (Pascal) GPUs (each of 4). We apply a four-round training scheme illustrated in Section III-C, where each round adopts one different loss weight group. At the beginning of each round, the learning rate is initialized as  $10^{-4}$  and is decayed by half every 10 epochs. We train 20 epochs for the first three rounds and 30 for the last round.

In terms of time efficiency, we evaluate the inference time of existing state-of-the-art DNNs including both 2D and 3D based networks using a pair of stereo images ( $H = 576$ ,  $W = 960$ ) from the Scene Flow dataset [5] on a desktop-level Nvidia Titan X (Pascal) GPU (with 12GB memory) and a server-level Nvidia Tesla V100 GPU (with 32GB memory).

### B. Dataset

We used two publicly popular available datasets to train and evaluate the performance of our FADNet. The first one is Scene Flow which is produced by synthetic rendering techniques. The second one is KITTI 2015 which is captured by real world cameras and laser sensors.

1) *Scene Flow* [5]: a large synthetic dataset which provides totally 39,824 samples of stereo RGB images (35,454 for training and 4,370 for testing). The full resolution of the images is  $960 \times 540$ . The dataset covers a wide range of object shapes and texture and provides high-quality dense disparity ground truth. We use the endpoint error (EPE) as error measurement. We remove those pixels whose disparity values are larger than 192 in the loss computation, which is typically done by the previous studies [6][7].

2) *KITTI 2015* [26]: an open benchmark dataset which contains 200 stereo images which are grayscale and have a resolution of  $1241 \times 376$ . The ground truth of disparity is generated by the LIDAR equipment, so the disparity map is very sparse. During training, we randomly crop  $1024 \times 256$  resolution of images and disparity maps. We use its full resolution during test.

### C. Experimental Results

The experimental results on the Scene Flow dataset are shown in Table IV. Regarding the model accuracy measured with EPE, our proposed FADNet achieves comparable performance compared to the state-of-the-art CVM-Conv3D (PSMNet and GANet), while FADNet is  $46 \times$  and  $8 \times$  faster

<sup>1</sup><https://pytorch.org>

<sup>2</sup><https://github.com/HKBU-HPML/FADNet>

TABLE IV: Disparity EPE on the scene flow dataset

Model	EPE	Memory (GB)	Runtime (ms)	
			Titan X (Pascal)	Tesla V100
FADNet(ours)	<b>0.83</b>	3.87	65.5	48.1
DispNetC	1.68	1.62	28.7	18.7
DenseMapNet	<u>5.36</u>	-	<u><math>\leq 30</math></u>	-
PSMNet	1.09	13.99	OOM	399.3
GANet	0.84	29.1	OOM	2251.1

Note: “OOM” indicates that it runs out of memory. Runtime is the inference time per pair of stereo images, and it is measured by 100 runs with average. The underline numbers are from the original paper.

than GANet and PSMNet respectively on an Nvidia Tesla V100 GPU. Even PSMNet and GANet are not runnable on the Titan X (Pascal) GPU, which implies high cost of them in practice. Compared to DispNetC and DenseMapNet, even FADNet is relatively slow, it predicts the disparity more than  $2 \times$  accurate than DispNetC and DenseMapNet, which is a huge accuracy improvement. The visualized comparison with predicted disparity maps are shown in Fig. 3.

TABLE V: Results on the KITTI 2015 dataset

Model	Noc(%)			All(%)		
	D1-bg	D1-fg	D1-all	D1-bg	D1-fg	D1-all
FADNet(ours)	2.49%	<b>3.07%</b>	2.59%	2.68%	3.50%	2.82%
DispNetC	4.11%	3.72%	4.05%	4.32%	4.41%	4.34%
GC-Net	2.02%	5.58%	2.61%	2.21%	6.16%	2.87%
PSMNet	1.71%	4.31%	2.14%	1.86%	4.62%	2.32%
GANet	<b>1.34%</b>	3.11%	<b>1.63%</b>	<b>1.48%</b>	<b>3.46%</b>	<b>1.81%</b>

Note: “Noc” and “All” indicates percentage of outliers averaged over ground truth pixels of non-occluded and all regions respectively. “D1-bg”, “D1-fg” and “D1-all” indicates percentage of outliers averaged over background, foreground and all ground truth pixels respectively.

From the visualized disparity maps shown in Fig. 3, we can see that the details of textures are successfully estimated by our FADNet while PSMNet is a little worse and DispNetC almost misses all the details. The visualization results are dramatically different although the EPE gap between DispNetC and FADNet is only 0.85. In the qualitative evaluation, FADNet is more robust and accurate than DispNetC with a 2D based network and PSMNet with a 3D based network.

From Table IV, it is noticed that the CVM-Conv3D architectures cannot be used on the desktop-level GPU which is equipped with 12 GB memory, while the proposed FADNet requires only 3.87 GB to perform the disparity estimation. The low memory requirement of FADNet makes it much easier for deployment in real-world applications. DispNetC is also an efficient architecture in terms of both memory consumption and computing efficiency, but its estimation performance is bad such that it cannot be used in real-world applications. In summary, FADNet not only achieves high disparity estimation accuracy, but it is also very efficient and practical for deployment.

The experimental results on the KITTI 2015 dataset are shown in Table V. GANet achieves the best estimation results among the evaluated models, and our proposed FADNet performs comparable error rates on the metric of D1-fg. The qualitative evaluation of the KITTI 2015 dataset is shown in Fig. 4, it is seen that the error maps of FADNet are close to PSMNet, while they are much better than that of DispNetC.



Fig. 3: Results of disparity prediction for Scene Flow testing data. The leftest column shows the left images of the stereo pairs. The rest three columns respectively show the disparity maps estimated by (a) DispNetC [5], (b) PSMNet [6], (c) FADNet.

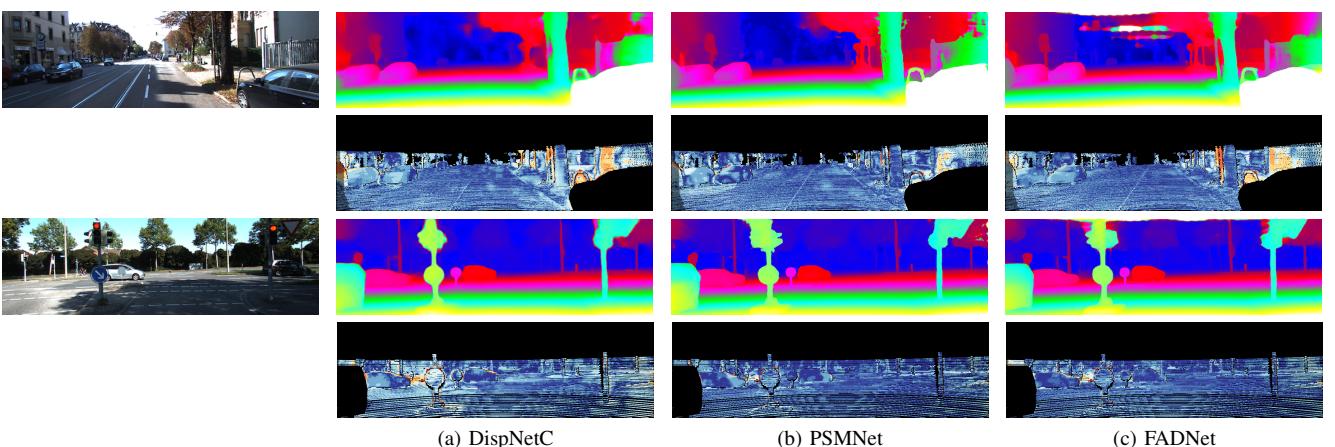


Fig. 4: Results of disparity prediction for KITTI 2015 testing data. The leftest column shows the left images of the stereo pairs. The rest three columns respectively show the disparity maps estimated by (a) DispNetC [5], (b) PSMNet [6], (c) FADNet, as well as their error maps.

## V. CONCLUSION AND FUTURE WORK

In this paper, we proposed an efficient yet accurate neural network, FADNet, for end-to-end disparity estimation to embrace the time efficiency and estimation accuracy on the stereo matching problem. The proposed FADNet exploits point-wise correlation layers, residual blocks, and multi-scale residual learning strategy to make the model be accurate in many scenarios while preserving fast inference time. We compared FADNet with existing state-of-the-art 2D and 3D based methods on two popular datasets in terms of accuracy and speed. Experimental results showed that FADNet achieves comparable accuracy while it runs much faster than the 3D based models. Compared to the 2D based models, FADNet is more than two times accurate.

We have two future directions following our discovery in

this paper. First, we would like to develop fast disparity inference of FADNet on edge devices. Since the computational capability of them is much lower than that of the server GPUs used in our experiments, it is necessary to explore the techniques of model compression, including pruning, quantization, and so on. Second, we would also like to apply AutoML [9] for searching a well-performing network structure for disparity estimation.

## ACKNOWLEDGEMENTS

This research was supported by Hong Kong RGC GRF grant HKBU 12200418. We thank the anonymous reviewers for their constructive comments and suggestions. We would also like to thank NVIDIA AI Technology Centre (NVAITC) for providing the GPU clusters for some experiments.

## REFERENCES

- [1] H. Hirschmuller, “Stereo processing by semiglobal matching and mutual information,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 30, no. 2, pp. 328–341, 2007.
- [2] S. Zagoruyko and N. Komodakis, “Learning to compare image patches via convolutional neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4353–4361.
- [3] J. Zbontar, Y. LeCun *et al.*, “Stereo matching by training a convolutional neural network to compare image patches.” *Journal of Machine Learning Research*, vol. 17, no. 1-32, p. 2, 2016.
- [4] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, “Flownet: Learning optical flow with convolutional networks,” in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.
- [5] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, “A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4040–4048.
- [6] J.-R. Chang and Y.-S. Chen, “Pyramid stereo matching network,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [7] F. Zhang, V. Prisacariu, R. Yang, and P. H. Torr, “Ga-net: Guided aggregation net for end-to-end stereo matching,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [8] T. Saikia, Y. Marrakchi, A. Zela, F. Hutter, and T. Brox, “Autodispnet: Improving disparity estimation with automl,” in *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.
- [9] X. He, K. Zhao, and X. Chu, “Automl: A survey of the state-of-the-art,” *arXiv preprint arXiv:1908.00709*, 2019.
- [10] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, “Flownet: Learning optical flow with convolutional networks,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2758–2766.
- [11] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, “Flownet 2.0: Evolution of optical flow estimation with deep networks,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
- [12] E. Ilg, T. Saikia, M. Keuper, and T. Brox, “Occlusions, motion and depth boundaries with a generic network for disparity, optical flow or scene flow estimation,” in *The European Conference on Computer Vision (ECCV)*, September 2018.
- [13] Z. Liang, Y. Feng, Y. Guo, H. Liu, W. Chen, L. Qiao, L. Zhou, and J. Zhang, “Learning for disparity estimation through feature constancy,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2811–2820.
- [14] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [15] A. E. Orhan and X. Pitkow, “Skip connections eliminate singularities,” *arXiv preprint arXiv:1701.09175*, 2017.
- [16] W. Zhan, X. Ou, Y. Yang, and L. Chen, “Dsnet: Joint learning for scene segmentation and disparity estimation,” in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 2946–2952.
- [17] X. Du, M. El-Khamy, and J. Lee, “Amnet: Deep atrous multiscale stereo disparity estimation networks,” *arXiv preprint arXiv:1904.09099*, 2019.
- [18] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry, “End-to-end learning of geometry and context for deep stereo regression,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 66–75.
- [19] G.-Y. Nie, M.-M. Cheng, Y. Liu, Z. Liang, D.-P. Fan, Y. Liu, and Y. Wang, “Multi-level context ultra-aggregation for stereo matching,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3283–3291.
- [20] Y. Luo, J. Ren, M. Lin, J. Pang, W. Sun, H. Li, and L. Lin, “Single view stereo matching,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [21] J. Pang, W. Sun, J. S. Ren, C. Yang, and Q. Yan, “Cascade residual learning: A two-stage convolutional neural network for stereo matching,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 887–895.
- [22] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, “Flownet 2.0: Evolution of optical flow estimation with deep networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2462–2470.
- [23] R. Atienza, “Fast disparity estimation using dense networks,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 3207–3212.
- [24] J. Pang, W. Sun, J. S. Ren, C. Yang, and Q. Yan, “Cascade residual learning: A two-stage convolutional neural network for stereo matching,” in *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.
- [25] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 248–255.
- [26] M. Menze, C. Heipke, and A. Geiger, “Joint 3d estimation of vehicles and scene flow,” in *ISPRS Workshop on Image Sequence Analysis (ISA)*, 2015.