Attention-Aware Feature Aggregation for Real-time Stereo Matching on Edge Devices

Jia-Ren Chang[1,2], Pei-Chun Chang[1], and Yong-Sheng Chen[1]

[1] Department of Computer Science, National Chiao Tung University,
Hsinchu, Taiwan
[2] aetherAI, Taipei, Taiwan
{followwar.cs00g, maplepig.cs05g, yschen}@nctu.edu.tw

Abstract. Recent works have demonstrated superior results for depth estimation from a stereo pair of images using convolutional neural networks. However, these methods require large amounts of computational resources and are not suited to real-time applications on edge devices. In this work, we propose a novel method for real-time stereo matching on edge devices, which consists of an efficient backbone for feature extraction, an attention-aware feature aggregation, and a cascaded 3D CNN architecture for multi-scale disparity estimation. The efficient backbone is designed to generate multi-scale feature maps with constrained computational power. The multi-scale feature maps are further adaptively aggregated via the proposed attention-aware feature aggregation module to improve representational capacity of features. Multi-scale cost volumes are constructed using aggregated feature maps and regularized using a cascaded 3D CNN architecture to estimate disparity maps in anytime settings. The network infers a disparity map at low resolution and then progressively refines the disparity maps at higher resolutions by calculating the disparity residuals. Because of the efficient extraction and aggregation of informative features, the proposed method can achieve accurate depth estimation in real-time inference. Experimental results demonstrated that the proposed method processed stereo image pairs with resolution 1242×375 at 12-33 fps on an NVIDIA Jetson TX2 module and achieved competitive accuracy in depth estimation. The code is available at https://github.com/JiaRenChang/RealtimeStereo.

1 Introduction

Depth estimation from stereo images is an essential task for computer vision applications, including autonomous driving for vehicles, 3D model reconstruction, and object detection and recognition [1, 2]. Given a pair of rectified stereo images, the goal of depth estimation is to compute the disparity d for each pixel in the reference image. Disparity refers to the horizontal displacement between a pair of corresponding pixels in the left and right images. If the corresponding point for pixel (x, y) in the left image is found at $(x - d, y)$ in the right image, then the depth of this pixel is calculated by $\frac{fB}{d}$, where f is the focal length of the camera and B is the distance between two camera centers.

The typical pipeline for stereo matching involves finding the corresponding points based on matching cost and post-processing [3]. Recently, convolutional neural networks (CNNs) have been applied to learn to compute the matching cost between two image patches in MC-CNN [4]. Early approaches using CNNs treat the problem of correspondence estimation as similarity computation [4–6], where CNNs compute the similarity score for a pair of image patches to further determine whether they are matched. Further studies propose more complex network architectures for better matching cost computation. Some studies propose end-to-end networks for stereo depth estimation. Specifically, these studies first extract image features via a CNN, use features to form a cost volume, and further process the cost volume with a 3D CNN. GC-Net [7] develops an encoder-decoder architecture of 3D CNN for learning context of cost volume. PSMNet [8] exploits pyramid features via spatial pyramid pooling and a stacked hourglass 3D CNN for regularizing cost volume. GA-Net [9] integrates semi-global matching [10] into 3D CNN for cost filtering. Further studies, such as AANet [11] and DeepPruner [12], are attempts to reduce the latency during inference by removing 3D convolutions or reducing disparity searching space, respectively. These end-to-end approaches demonstrated the state-of-the-art performance on stereo matching. However, difficulties still remain in issues such as robustness, generalization ability, and computational cost.

One major problem with current CNN-based stereo matching methods is how to efficiently perform inference using low-budget devices which has limited computational resources. As an example, PSMNet [8], which is one of the current state-of-the-art methods for stereo depth estimation, runs at a frame rate below 0.16 fps on an NVIDIA Jetson TX2 module. This is far from real-time applications on drones or robots. AnyNet [13] is proposed as a trade-off between computation and accuracy at inference time. It can run 10-35 fps on an NVIDIA Jetson TX2, at the expense of marginal accuracy on stereo matching.

In this paper, we propose a novel method for real-time stereo matching on edge devices in anytime settings, as shown in Fig. 1. Inspired by designing efficient CNNs on edge devices [14–16], we first build an efficient backbone to extract multi-scale features. We further propose an attention-aware feature aggregation module to learn adaptive fusion of multi-scale features. The aggregated features are utilized to construct multi-scale cost volumes. These cost volumes are regularized via a cascaded 3D CNN architecture to perform depth estimation in anytime settings. Similar to other coarse-to-fine approaches [13, 17], the proposed method begins by estimating disparity maps at a low resolution and further refines the upsampled disparity maps. A wide range of possible frame rates are attainable (12-33 fps on an NVIDIA Jetson TX2 module) while still preserving accurate disparity estimation in a high-latency setting.

We evaluate the proposed method on several datasets and the major contributions of this work are:

- We propose a novel architecture to efficiently extract informative features by adaptively incorporating multi-resolutional information into the encoded features via cross-scale connections.

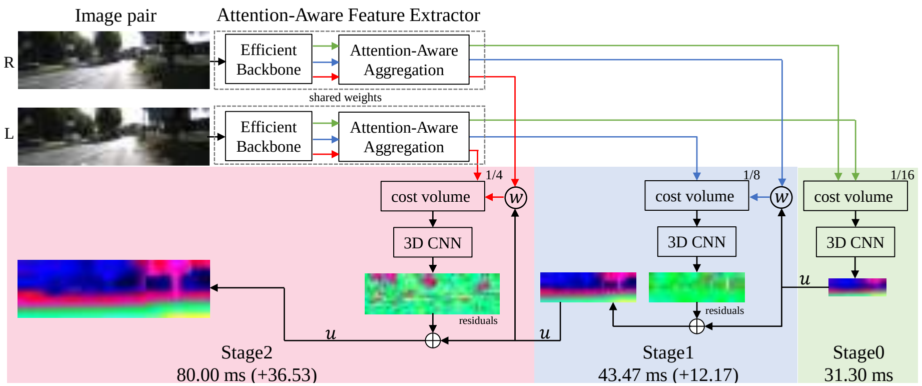


Fig. 1. Architecture overview of the proposed method. Left and right input stereo images are fed to two weight-sharing attention-aware feature extractor, consisting of an efficient backbone and an attention-aware feature aggregation. The left and right feature maps are then used to form a cost volume, which is fed into a cascaded 3D CNN architecture for cost volume regularization. The operator u indicates bilinear upsampling and the operator $+$ indicates element-wise summation. The warping function w uses the upsampled disparity map from previous stage to pixel-wise align the image pairs. The network can be queried to output its current best estimate in anytime settings. For example, a coarse disparity map can be obtained at 31.3 ms while a fine disparity map can be obtained at 80 ms on an NVIDIA Jetson TX2 for 1242×375 input image pairs.

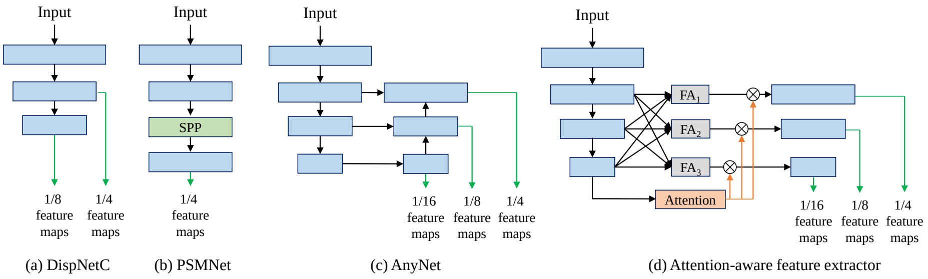
– We demonstrate that the proposed method considers both computational cost and accuracy at inference time and runs at real-time on low-budget devices.
– We show that the proposed method achieves competitive results compared to state-of-the-art methods.

2 Methods

The proposed network consists of an efficient backbone for feature extraction, an attention-aware aggregation module for feature fusion, and a cascaded 3D CNN architecture for estimating disparity maps and residuals, as illustrated in Fig. 1.

2.1 Attention-Aware Feature Extraction

We propose an attention-aware feature extractor to obtain multi-scale feature maps from stereo image pairs. As shown in Fig. 2, we compare the proposed feature extractor with those of other widely-used stereo matching architectures. The DispNetC [18] directly adopted different scales of feature maps. The PSMNet [8] introduced spatial pyramid pooling (SPP) to incorporate pyramid representations. The AnyNet [13] adopted the U-Net architecture [20] to learn top-down



(a) DispNetC (b) PSMNet (c) AnyNet (d) Attention-aware feature extractor

Fig. 2. The comparison of feature extractors of widely-used stereo matching methods, including (a) DispNetC [18], (b) PSMNet [8], (c) AnyNet [13], and (d) the proposed attention-aware feature extractor.

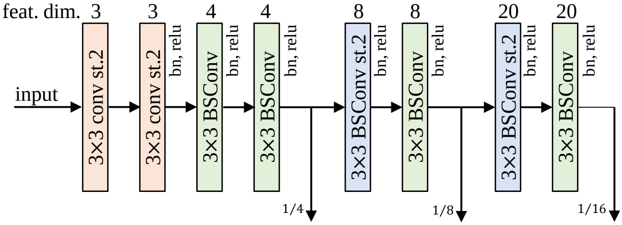
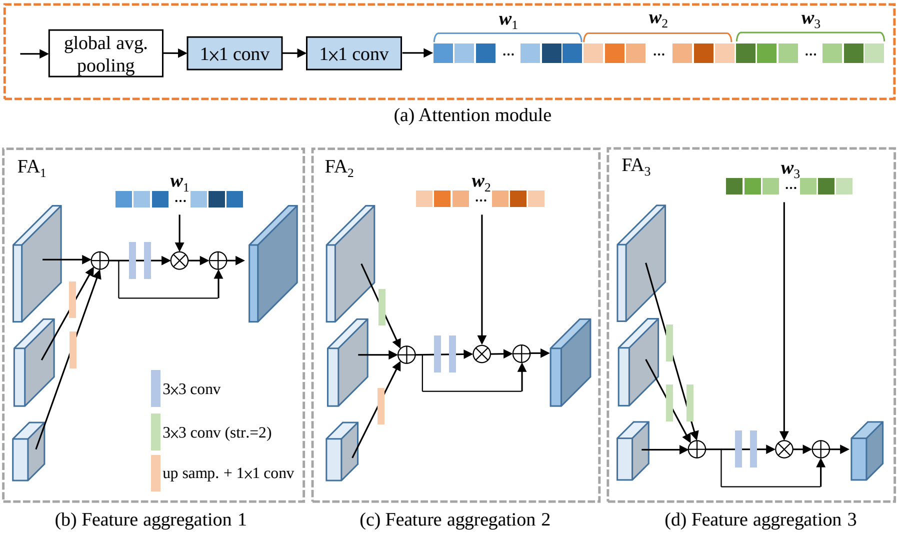


Fig. 3. The architecture of the proposed efficient backbone for feature extraction. The feat. dim. indicates the number of output feature maps. Each convolution follows batch normalization [19] and ReLU except the first one.

features. By contrast, the proposed attention-aware feature extractor can adaptively aggregate information from different scales into the encoded features with attention mechanism. As shown in Fig. 2 (d), the attention-aware feature extractor consists of an efficient backbone, three feature aggregation modules, and an attention module. We describe these components of the proposed method in the following.

Efficient Backbone Previous studies [14–16] have developed efficient network architecture on limited resource devices. Remarkably, the depthwise separable convolution (DSConv) [14, 16] can balance the tradeoff between computation and accuracy, which factorizes a standard convolution into a depthwise convolution and a 1×1 convolution. The blueprint separable convolutions (BSConv) [15] further provides a justification for DSConv by analysing intra-kernel correlations of vanilla CNNs. As shown in Fig. 3, two strided 3×3 convolutions are used to extract coarse features of input image, and several stacked BSConvs are applied to distill information from these features with less computational cost. In order to obtain multi-scale representation, the strided BSConv are utilized to reduce the size of feature maps and to obtain multi-scale feature maps sequentially. Note that these multi-scale feature maps are parallelly aggregated using the attention-



(a) Attention module

(b) Feature aggregation 1

(c) Feature aggregation 2

(d) Feature aggregation 3

Fig. 4. The architecture of the proposed attention-aware feature aggregation module. The feature aggregation modules FA$_1$, FA$_2$ and FA$_3$ are correspond to modules illustrated in Fig. 2 (d). Each convolution follows batch normalization [19] and ReLU except in the attention module.

aware feature aggregation module described in the next section, and only the 1/16 feature map is used to learn all channel-wised weights across multi-scale information.

Attention-Aware Feature Aggregation As shown in Fig. 2, feature information at different scales is propagated sequentially in state-of-the-art stereo matching architectures. In stead of sequential propagation from coarse to fine levels such as U-Net [20], the proposed method parallelly aggregate information from all levels. There are three levels ($1/4, 1/8, 1/16$ size) of output feature maps from the efficient backbone. As shown in Fig. 4 (b), (c) and (d), we proposed three parallel aggregation modules for these three level of features. We formulate the aggregation module in the following:

$$\hat{F^s} = \sum_{k=1}^{S} f_k(F^k), \qquad s = 1, 2, ..., S \ , \tag{1}$$

where S is the level number of feature maps ($S = 3$ in this work) and F^k is the output feature map of the efficient backbone at level k. Similar to HRNet [21], we adopt the definition of f_k to calculate feature maps depending on their resolutions:

$$f_k = \begin{cases} I, & k = s, \\ (s-k)\ 3 \times 3 \text{ conv with stride 2}, & k < s, \\ \text{bilinear upsampling and } 1 \times 1 \text{ conv}, & k > s, \end{cases} \tag{2}$$

where I denotes the identity function, $s-k$ convolutions with stride 2 are used for downsampling the feature maps to achieve consistent size. Bilinear upsampling is applied to achieve consistent size followed by a 1×1 convolution to align the number of channels.

Inspired by SENet [22], we propose an attention module, as shown in Fig. 4 (a), to recalibrate channel importance for boosting feature discriminability. The recalibrated feature maps $\tilde{F^s}$ can be formulated as:

$$\tilde{F^s} = \phi_s(\hat{F^s}) \cdot w_s + \hat{F^s}, \qquad s = 1, 2, ..., S \ , \tag{3}$$

where ϕ_s consists of two 3×3 convolutions with batch normalization and ReLU, and w_s is the attention weights learnt from the proposed attention module. We apply global average pooling on the final output feature maps of the efficient backbone, followed by two 1×1 convolutions to compute channel-wise attention weights w_s for multi-scale feature maps. The final recalibrated feature maps at each level are utilized to form cost volumes for disparity estimation.

2.2 Cost Volume

The GC-Net [7] and PSMNet [8] approaches concatenate left and right features to learn matching cost estimation using 3D CNNs. For real-time constraints, we adopt a distance metric to form a cost volume by computing the L_1 distance of left feature maps with their corresponding right feature maps across each disparity level, resulting in a 4D volume (height×width×disparity×1).

A crucial aspect of the proposed architecture is that we compute the full disparity map only at a very low resolution in Stage 0 and compute disparity residuals in Stages 1 and 2. By focusing on residuals, that is, correction of existing disparities, we can greatly reduce the range of searching correspondences to D = 5 (that is, offsets -2, -1, 0, 1, 2) and obtain significant speedups. As shown in Fig. 1, in order to compute the residuals in Stages 1 and 2, we first upscale the coarse disparity map and use it to warp the input features at the higher scale by applying the disparity estimation in pixel-wise manner. If the current disparity estimate is correct, the updated right feature maps should match the left feature maps. Because of the coarseness of the low resolution inputs, mismatch may occur but may be corrected by computing residual disparity maps. Estimation of the residual disparity is accomplished similarly to the full disparity map computation. The only difference is that the residual disparity map is limited to -2 to 2, and the resulting residual disparity map is added to the upscaled disparity map from the previous stage.

2.3 Cascaded 3D CNN Architecture

The attention-aware feature extractor facilitates stereo matching by taking into account multi-resolution information and channel-wise recalibration. To aggregate the feature information along the disparity dimension as well as spatial dimensions, we follow the design in [13] to construct a cascaded 3D CNN architecture for cost volume regularization. The cascaded 3D CNN architecture contains three subnetworks depending on spatial resolutions, and each subnetwork has five 3×3×3 convolutions followed by batch normalization and ReLU except the final one. The subnetworks are further used for regularizing cost volume at each level. We then apply regression to calculate the disparity map, which is introduced in Section 2.4. Finally, we upsample the disparity map to the original image size.

2.4 Disparity Regression

We use disparity regression as proposed in [7] to estimate the continuous disparity map. The probability of each disparity d is calculated from the calculated cost c_d via the softmax operation $\sigma(\cdot)$. The disparity $\hat{d}$ is calculated as the sum of each disparity d weighted by its probability, as

$$\hat{d} = \sum_{d=0}^{D_{max}} d \times \sigma(-c_d)\,. \tag{4}$$

As reported in [7], the above disparity regression is more robust than classification-based stereo matching methods.

3 Experiments

We evaluated our method on three stereo datasets: Scene Flow [18], KITTI 2012, and KITTI 2015 [23].

3.1 Datasets

1. Scene Flow: a large-scale synthetic dataset containing 35,454 training and 4,370 testing images with $H = 540$ and $W = 960$. This dataset provides dense and elaborate disparity maps as ground truth. Some pixels have large disparities and are excluded in the loss computation if the disparity is larger than the limits set in our experiment.
2. KITTI 2012/2015: a real-world dataset with street views from a driving car. KITTI2012 contains 194 training stereo image pairs with sparse ground-truth disparities obtained using LiDAR and 195 testing image pairs without ground-truth disparities. KITTI2015 contains 200 training stereo image pairs with sparse ground-truth disparities obtained using LiDAR and another 200 testing image pairs without ground-truth disparities. Image size is $H = 376$ and $W = 1240$. We further divided the whole training data into a training set (80%) and a validation set (20%).

3.2 Implementation Details

The proposed method was implemented using PyTorch. All models were end-to-end trained with Adam ($\beta_1 = 0.9, \beta_2 = 0.999$). We performed color normalization as in [8] on the entire dataset for data preprocessing. During training, images were randomly cropped to size $H = 288$ and $W = 576$. The maximum disparity (D) was set to 192. We trained our models from scratch using the Scene Flow dataset with a learning rate of 0.0005 for 10 epochs. For Scene Flow, the trained model was directly used for testing. For KITTI validation, we used the model trained with Scene Flow after fine-tuning on the KITTI training set for 300 epochs. The learning rate of this fine-tuning began at 0.0005 for the first 200 epochs and 0.00005 for the remaining 100 epochs. For KITTI submission, we fine-tuned the pre-trained model on the combination of KITTI 2012/2015 for 500 epochs. The learning rate of this fine-tuning began at 0.0005 for the first 300 epochs and 0.0001 for the remaining 200 epochs. Then another 500 epochs were trained on the separate KITTI 2012/2015 training set, and the learning rate schedule remained the same. The batch size was set to 6 for Scene Flow and 4 for KITTI for the training on an NVIDIA Titan-Xp GPU. We adopted smooth L_1 loss as in [8]. The loss weights of three output stages were set to 0.25, 0.5, and 1.0. We calculated end-point-error (EPE) for Scene Flow dataset and three pixel error (3-px err.) for KITTI dataset.

3.3 Ablation Studies

To validate the effectiveness of each component proposed in this paper, we conducted controlled experiments on Scene Flow test set and KITTI 2015 validation set.

Components: As shown in Tab. 1, removing the feature aggregation and attention module leads to significant performance drop. The best performance is obtained by integrating these two modules.

Usage of Convolution: As described in Sec. 2.1, we adopt BSConv [15] to design the efficient backbone for feature extraction. As shown in Tab. 2, we compare the performance of vanilla convolution, DSConv [14], and BSConv [15] in our backbone. We used full architecture of the proposed method, and only replaced the convolutions in backbone. The results demonstrate that the BSConv can achieve the best performance with moderate FLOPs and number of parameters.

Cost Metric: We form cost volumes by computing the L_1 distance of left feature maps with their corresponding right feature maps, as described in Sec. 2.2. As shown in Tab. 3, we evaluated the performance with other metrics, such as L_2 distance and correlation. The results demonstrate that the usages of L_1 and L_2 distance have similar performance, but the correlation has the worst performance.

Table 1. Ablation study of feature aggregation (Feat. Agg.) and attention (Att.) modules. The best performance is obtained by integrating these two modules. S0, S1, and S2 indicate Stage 0, Stage 1, and Stage 2, respectively.

			Scene Flow (EPE)			KITTI 2015 (3-px err.)		
	Feat. Agg.	Att.	S0	S1	S2	S0	S1	S2
Baseline			4.49	4.26	4.07	10.95	9.27	7.28
+Agg.	√		4.44	4.26	4.03	10.78	9.07	7.07
Ours	√	√	**4.34**	**4.14**	**3.90**	**11.03**	**9.15**	**6.76**

Table 2. Ablation study of convolution usage in efficient backbone. The best performance is obtained by using BSConv [15].

	Scene Flow (EPE)			FLOPs (B)	Params
Conv Type	S0	S1	S2		
Vanilla Conv	4.40	4.17	3.98	0.578	28,054
DSConv	4.62	4.40	4.18	0.543	23,005
BSConv	**4.34**	**4.14**	**3.90**	0.548	23,158

3.4 Evaluation Results

We evaluated the proposed method on KITTI 2012/2015 validation set for anytime estimation and test set for benchmarking.

Anytime Setting Estimation The proposed method performs anytime depth estimation that balance the tradeoff between accuracy and speed. It can output three stages of disparity maps with progressive precision and runtime. As shown in Tab. 4, we clocked the running time of each stage on a low-budget NVIDIA Jetson TX2 with 1242×375 image pairs.

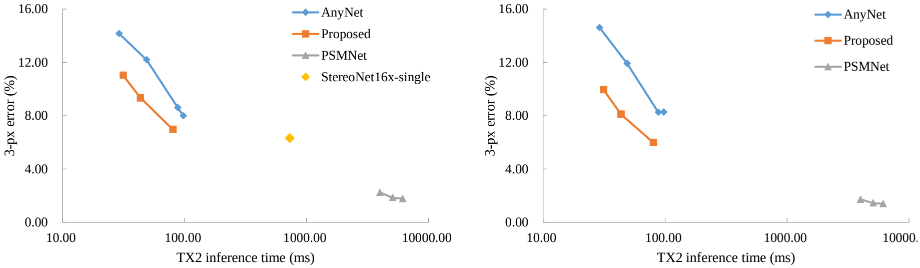
We further compare our results to other state-of-the-art approaches, such as AnyNet [13], PSMNet [8] and StereoNet [24], as shown in Tab. 4. For a fair comparison, we adopted their public codes and trained on the same train/validation split. Note that in the results of PSMNet [8], the evaluation of intermediate output in stacked hourglass 3D CNN are also reported. As shown in Fig. 5, the proposed method balanced best tradeoff between accuracy and runtime. It can achieve similar performance but runs 9× faster compared with StereoNet [24]. The proposed method can outperform AnyNet [13] by a notable margin with only

Table 3. Ablation study of cost metric. The best performance for forming cost volume is achieved by computing the L_1 distance.

	Scene Flow (EPE)		
Cost metric	S0	S1	S2
Correlation	4.7	4.44	4.20
L_2 distance	4.37	4.19	3.92
L_1 distance	**4.34**	**4.14**	**3.90**

Table 4. The evaluation results (3-px error) of the proposed method on KITTI 2012/2015 validation set. The runtime is clocked on an NVIDIA Jetson TX2 with 1242×375 image pairs. It clearly demonstrates that the proposed method can produce accurate disparity maps in real-time on edge device.

	KITTI 2012			KITTI 2015			FLOPs	Params	Runtime (ms)		
	S0	S1	S2	S0	S1	S2	(B)	(M)	S0	S1	S2
StereoNet-16× [24]	-	-	-	-	-	6.32	59.82	0.425	-	-	727.3
PSMNet [8]	1.72	1.45	1.39	2.24	1.86	1.77	937.9	5.224	3995	5072	6131
AnyNet [13]	14.61	11.92	8.26	14.24	12.11	8.51	1.339	0.043	29.00	48.80	87.90
Ours	9.75	8.09	6.01	11.03	9.15	6.76	0.548	0.023	31.30	43.47	80.00



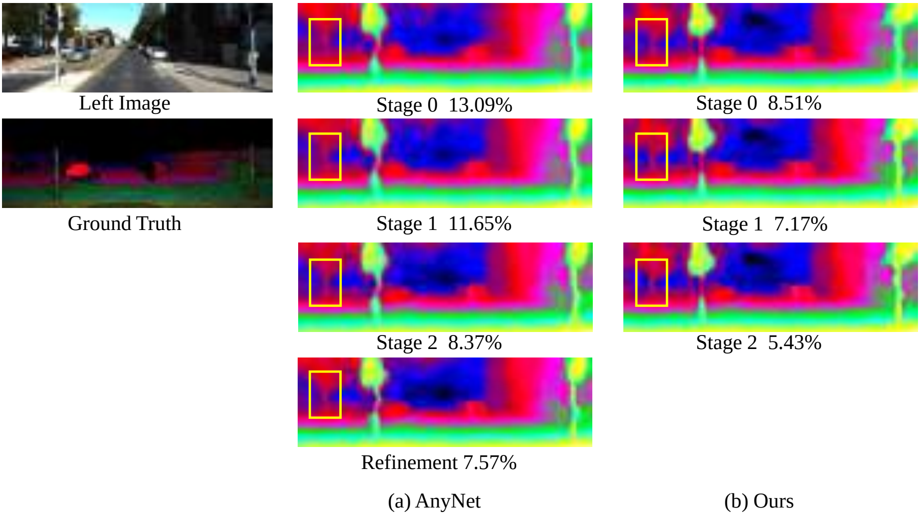
(a) KITTI 2015

(b) KITTI 2012

Fig. 5. The proposed method runs significantly faster than other approaches, such as AnyNet [13] and PSMNet [8], on a low-budget device (TX2) with competitive performance.

half of FLOPs and parameters compared to AnyNet. PSMNet [8] can produce very accurate results; however, it needs high computational costs and cannot run in real-time for applications. We also compared qualitative results in Fig. 6. The yellow squares show that the proposed method can produce more sharp results than AnyNet [13]. The results demonstrate that the progressive refinement of the proposed method efficiently improves the quality of disparity maps at a low computational cost.

Benchmark Results We further evaluated the proposed method on KITTI 2012/2015 test set for benchmarking. According to the online leaderboard, as shown in Tab. 5, the overall three-pixel-error for the proposed method was 6.94 on KITTI 2012 and 7.54% on KITII 2015, which is comparable to state-of-the-art methods with very low computational costs. We also clocked the runtime for each method on a Jetson TX2 according to their public codes. We found that GC-Net [7] and GA-Net-15 [9] cannot run on TX2 due to the problem of out of memory (OOM). Other state-of-the-art approaches, such as PSMNet [8], DeepPruner-Best [12], HD^3 [25] and GwcNet [26], can produce accurate results; however, they have very high latency (>2000 ms) and cannot be applied for real-time applications.



Left Image

Ground Truth

Stage 0 13.09%

Stage 1 11.65%

Stage 2 8.37%

Refinement 7.57%

(a) AnyNet

Stage 0 8.51%

Stage 1 7.17%

Stage 2 5.43%

(b) Ours

Fig. 6. The comparison of stage outputs between AnyNet [13] and the proposed method. The 3-px error is shown below each disparity map. The yellow squares depict that the proposed method can produce sharper results than those of AnyNet. Moreover, the accuracy in terms of 3-px error is progressively improved from Stage 0 to Stage 2.

There are some approaches designed for real-time applications, such as MADNet [27], DispNetC [18] and DeepPruner-Fast [12]. They can run in real-time in a high-budget device (NVIDIA 1080ti or higher), but still have noticeable latency on edge devices (>100 ms in an NVIDIA Jetson TX2). In contrast, the proposed method can achieve competitive performance of anytime depth estimation at 12-33 fps on an NVIDIA Jetson TX2. As shown in Fig. 7, we compare the qualitative results adopted from KITTI 2015 leaderboard with MADNet [27] and DispNetC [18]. It suggests that the proposed method can produce accurate disparity maps for real-world applications with very low amount of model parameters and computational costs.

3.5 Generalization

We further test the generalization ability of the proposed method on ETH3D dataset [28]. Specifically, we directly use our KITTI fine-tuned model to estimate the disparity map on ETH3D without further training. As shown in Fig. 8, the proposed method can generalize to other domains and can produce accurate disparity maps.

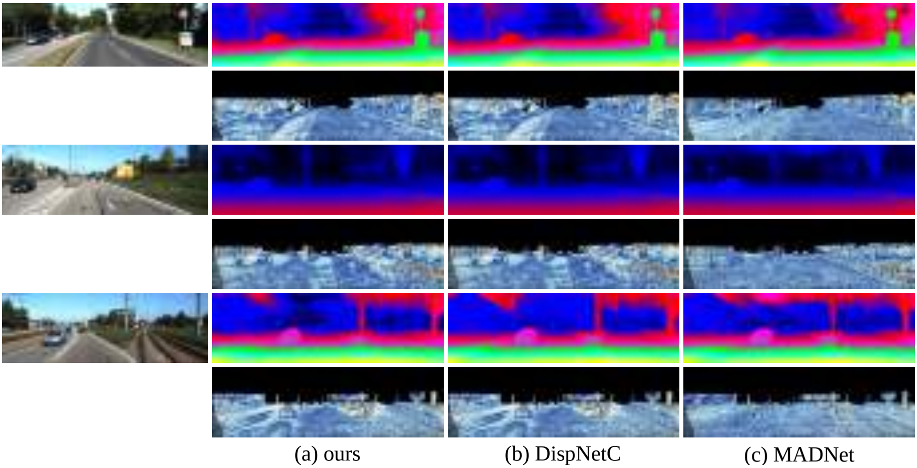


Fig. 7. The qualitative comparisons between (a) the proposed method, (b) DispNetC [18], and MADNet [27]. The first row of each left image is the predicted disparity map and the second row is the 3-px error map. The proposed method can produce accurate disparity of objects and background, and gives the possibility of real-time dense depth estimation on the low-budget devices.

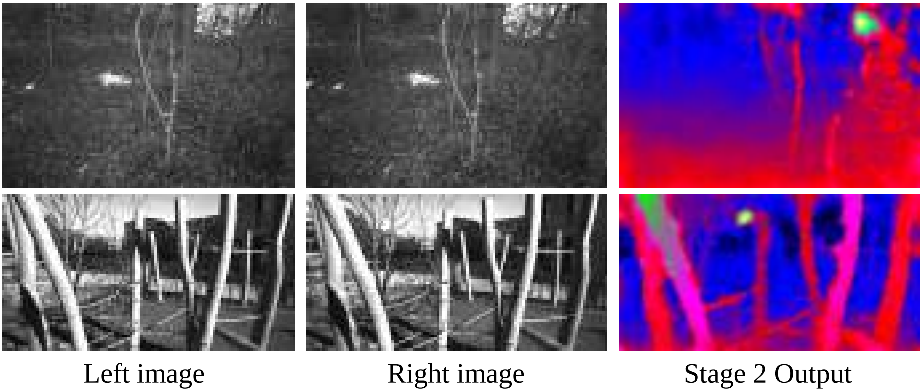


Fig. 8. Generalization on ETH3D dataset. The proposed method can produce accurate disparity maps, indicating its potential for practical applications, such as obstacle avoidance.

Table 5. We evaluate the proposed method on KITTI 2012/2015 test set. These results are adopted from KITTI leaderboard except the runtime. We clocked the runtime for each method on an Jetson TX2 according to their public codes with 1242×375 image pairs. The runtime ratios of the proposed method to other methods are also provided in the last column. Notably, our approach takes 61.5% time of DisPNetC [18]. The OOM indicates that the method cannot run on TX2 caused by out of memory.

Method	KITTI 2012		KITTI 2015		Params	Time	Ratio
	Noc	All	Noc	All	(M)	(ms)	(%)
GC-Net [7]	1.77	2.30	2.61	2.87	2.84	OOM	-
PSMNet [8]	1.49	1.89	2.14	2.32	5.22	6131	1.3
DeepPruner-Best [12]	-	-	1.95	2.15	7.39	3239	2.4
HD^3 [25]	1.40	1.80	1.87	2.02	39.50	2396	3.3
GwcNet [26]	1.32	1.70	1.92	2.11	6.52	3366	2.4
GA-Net-15 [9]	1.36	1.80	1.73	1.93	2.30	OOM	-
MADNet [27]	-	-	4.27	4.66	3.82	328	24.4
DispNetC [18]	4.11	4.65	4.05	4.34	38.14	130	61.5
DeepPruner-Fast [12]	-	-	2.35	2.59	7.47	1208	6.6
AANet [11]	1.91	2.42	2.32	2.55	4.00	1575	5.1
Ours	6.10	6.94	7.12	7.54	0.023	80	

3.6 Role of Attention

To have a profound understanding of the proposed attention-aware feature aggregation, we illustrate the weighting modulation of attention module in this section. From the Scene Flow test dataset, we chose three pairs of images exhibiting diverse disparity distribution and their attention activation profiles are illustrated in Fig. 9.

The role of attention module can be observed from two aspects. First, the activation across different disparity distributions is nearly identical in lower feature level, that is, w_1. This suggests that the importance of feature channels in the early stage of the network is prone to be fixed in estimating disparity maps. Second, the channel weights of high level feature maps, w_2 and w_3, tend to be disparity-specific because different disparity distribution exhibit different preferences to the weighting values of feature maps. These observations are consistent with findings in previous work [22]. That is, lower layer features are typically more general whereas higher layer features have greater specificity, suggesting the importance of recalibration during feature extraction in stereo matching.

4 Conclusion

Recent studies using CNNs for stereo matching have achieved prominent performance. Nevertheless, it remains intractable to estimate disparity in real-time on low-budget devices for practical applications. In this work, we proposed a novel network architecture that can perform anytime depth estimation of 12-33 fps with competitive accuracy on an edge device (NVIDIA Jetson TX2). Specifically, we designed an efficient backbone for balancing the trade-off between

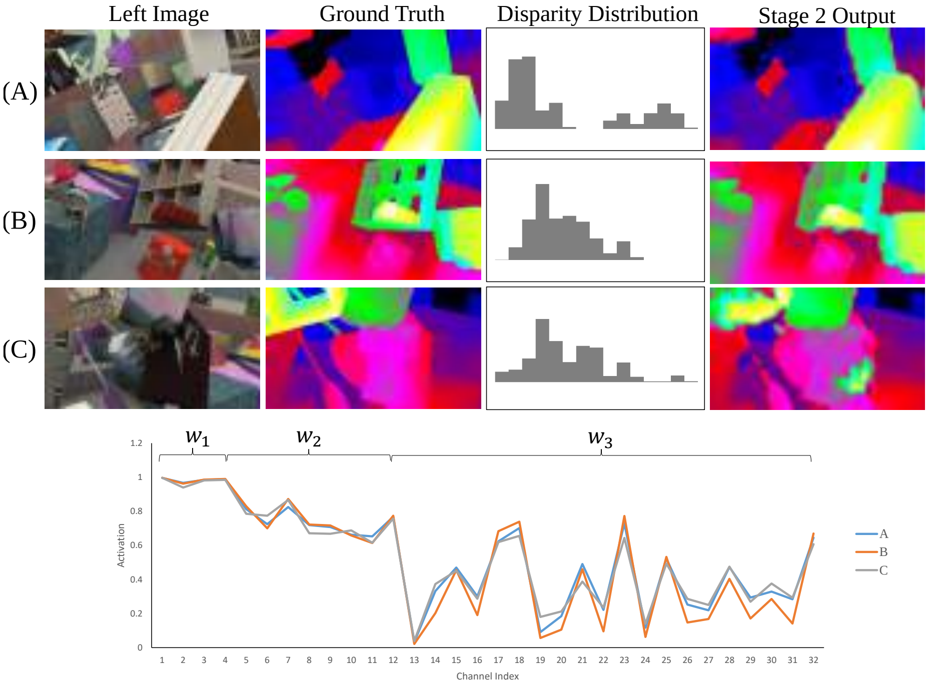


Fig. 9. Activation induced by the proposed attention module on Scene Flow test dataset. Note that the proposed module is a top-down attention which splits activation weights to w_1, w_2, and w_3 for modulating all three levels of feature maps.

speed and accuracy. Furthermore, an attention-aware feature aggregation is proposed to improve the representational capacity of features. A cascaded 3D CNN architecture is further used to estimate disparity maps in multiple spatial resolutions according to the constraints of computational costs. In our experiments, the proposed method achieved competitive performance in terms of accuracy and stat-of-the-art performance in terms of speed which was evaluated on an low-budget device. The estimated disparity maps clearly demonstrate that the proposed method can produce accurate depth estimation for practical applications.

5 Acknowledgments

This work was supported in part by the Taiwan Ministry of Science and Technology (Grants MOST-109-2218-E-002-038 and MOST-109-2634-F-009-015), Pervasive Artificial Intelligence Research (PAIR) Labs, Qualcomm Technologies Inc., and the Center for Emergent Functional Matter Science of National Chiao Tung University from the Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education in Taiwan.

References

1. Chen, X., Kundu, K., Zhu, Y., Berneshawi, A.G., Ma, H., Fidler, S., Urtasun, R.: 3D object proposals for accurate object class detection. In: Advances in Neural Information Processing Systems. (2015) 424–432
2. Zhang, C., Li, Z., Cheng, Y., Cai, R., Chao, H., Rui, Y.: Meshstereo: A global stereo model with mesh alignment regularization for view interpolation. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 2057–2065
3. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International journal of computer vision **47** (2002) 7–42
4. Zbontar, J., LeCun, Y.: Stereo matching by training a convolutional neural network to compare image patches. Journal of Machine Learning Research **17** (2016) 2
5. Shaked, A., Wolf, L.: Improved stereo matching with constant highway networks and reflective confidence learning. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2017)
6. Seki, A., Pollefeys, M.: SGM-Nets: Semi-global matching with neural networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2017)
7. Kendall, A., Martirosyan, H., Dasgupta, S., Henry, P., Kennedy, R., Bachrach, A., Bry, A.: End-to-end learning of geometry and context for deep stereo regression. In: The IEEE International Conference on Computer Vision (ICCV). (2017)
8. Chang, J.R., Chen, Y.S.: Pyramid stereo matching network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2018) 5410–5418
9. Zhang, F., Prisacariu, V., Yang, R., Torr, P.H.: Ga-net: Guided aggregation net for end-to-end stereo matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 185–194
10. Hirschmuller, H.: Accurate and efficient stereo processing by semi-global matching and mutual information. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. Volume 2., IEEE (2005) 807–814
11. Xu, H., Zhang, J.: Aanet: Adaptive aggregation network for efficient stereo matching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 1959–1968
12. Duggal, S., Wang, S., Ma, W.C., Hu, R., Urtasun, R.: Deeppruner: Learning efficient stereo matching via differentiable patchmatch. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 4384–4393
13. Wang, Y., Lai, Z., Huang, G., Wang, B.H., van der Maaten, L., Campbell, M., Weinberger, K.Q.: Anytime stereo image depth estimation on mobile devices. International Conference on Robotics and Automation (ICRA) (2019)
14. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2018) 4510–4520
15. Haase, D., Amthor, M.: Rethinking depthwise separable convolutions: How intra-kernel correlations lead to improved mobilenets. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 14600–14609
16. Ma, N., Zhang, X., Zheng, H.T., Sun, J.: Shufflenet v2: Practical guidelines for efficient cnn architecture design. In: Proceedings of the European conference on computer vision (ECCV). (2018) 116–131

17. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2018) 8934–8943
18. Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T.: A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2016)
19. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
20. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer (2015) 234–241
21. Sun, K., Xiao, B., Liu, D., Wang, J.: Deep high-resolution representation learning for human pose estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2019) 5693–5703
22. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2018) 7132–7141
23. Menze, M., Geiger, A.: Object scene flow for autonomous vehicles. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2015) 3061–3070
24. Khamis, S., Fanello, S., Rhemann, C., Kowdle, A., Valentin, J., Izadi, S.: Stereonet: Guided hierarchical refinement for real-time edge-aware depth prediction. In: Proceedings of the European Conference on Computer Vision (ECCV). (2018) 573–590
25. Yin, Z., Darrell, T., Yu, F.: Hierarchical discrete distribution decomposition for match density estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 6044–6053
26. Guo, X., Yang, K., Yang, W., Wang, X., Li, H.: Group-wise correlation stereo network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 3273–3282
27. Tonioni, A., Tosi, F., Poggi, M., Mattoccia, S., Stefano, L.D.: Real-time self-adaptive deep stereo. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 195–204
28. Schops, T., Schonberger, J.L., Galliani, S., Sattler, T., Schindler, K., Pollefeys, M., Geiger, A.: A multi-view stereo benchmark with high-resolution images and multi-camera videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 3260–3269