EAI-Stereo: Error Aware Iterative Network for Stereo Matching

Haoliang Zhao[1,4][0000−0002−3789−7451]†, Huizhou Zhou[2,4][0000−0003−1117−2919]†, Yongjun Zhang[1][0000−0002−7534−1219]⋆, Yong Zhao[1,3,4][0000−0002−7999−1083], Yitong Yang[1][0000−0001−5855−7248], and Ting Ouyang[1][0000−0002−8919−2947]

[1] State Key Laboratory of Public Big Data, Institute for Artificial Intelligence, College of Computer Science and Technology, Guizhou University, Guiyang, 550025, Guizhou, China
[2] School of Physics and Optoelectronic Engineering, Guangdong University of Technology, Guangzhou 510006, China
[3] The Key Laboratory of Integrated Microsystems, Shenzhen Graduate School, Peking University, China The Key Laboratory of Integrated Microsystems, Shenzhen Graduate School, Peking University, China
[4] Ghost-Valley AI Technology, Shenzhen, Guangdong, China

Abstract. Current state-of-the-art stereo algorithms use a 2D CNN to extract features and then form a cost volume, which is fed into the following cost aggregation and regularization module composed of 2D or 3D CNNs. However, a large amount of high-frequency information like texture, color variation, sharp edge etc. is not well exploited during this process, which leads to relatively blurry and lacking detailed disparity maps. In this paper, we aim at making full use of the high-frequency information from the original image. Towards this end, we propose an error-aware refinement module that incorporates high-frequency information from the original left image and allows the network to learn error correction capabilities that can produce excellent subtle details and sharp edges. In order to improve the data transfer efficiency between our iterations, we propose the Iterative Multiscale Wide-LSTM Network which could carry more semantic information across iterations. We demonstrate the efficiency and effectiveness of our method on KITTI 2015, Middlebury, and ETH3D. At the time of writing this paper, EAI-Stereo ranks 1^{st} on the Middlebury leaderboard and 1^{st} on the ETH3D Stereo benchmark for 50% quantile metric and second for 0.5px error rate among all published methods. Our model performs well in cross-domain scenarios and outperforms current methods specifically designed for generalization. Code is available at https://github.com/David-Zhao-1997/EAI-Stereo.

1 Introduction

Stereo Matching is a fundamental vision problem in computer vision with direct real-world applications in robotics, 3D reconstruction, augmented reality, and

⋆ Corresponding author. Email: zyj6667@126.com.
† These authors contributed equally.

autonomous driving. The task is to estimate pixel-wise correspondences of an image pair and generate a displacement map termed disparity which can be converted to depth using the parameters of the stereo camera system.

Generally, traditional stereo matching algorithms [13, 12, 9] perform the following four steps: matching cost computation, cost aggregation, disparity computation and refinement [29]. These algorithms can be classified into global methods and local methods. Global methods [33, 20, 8, 19] take advantage of solving the problem by minimizing a global energy function [5], which consumes time in exchange for accuracy. While local methods [15, 1] only make use of neighboring pixels, which usually runs faster [44]. However, in open-world environments, it is difficult for traditional methods to achieve satisfactory results in textureless regions and regions with repetitive patterns, while traditional high-precision algorithms are often limited in terms of computational speed.

Recently, with the continuous research in convolutional neural networks, learning-based methods are widely used in the field of binocular stereo matching. Compared with traditional methods, learning-based methods tend to produce more accurate and smooth [23, 35] disparity maps, and some of them also have advantages in computational speed [35, 41]. However, to apply algorithms in real scenarios, there are still some challenges to be solved.

One challenge is that current methods do not perform well in recovering thin objects and sharp edges. Most current algorithms use a 2D CNN to extract features and then form a cost volume, which is fed into the following cost aggregation and regularization module composed of 2D or 3D CNNs. During this process, a large amount of high-frequency information is ignored, which leads to relatively blurry and lacking detailed disparity maps. However, stereo vision is often used in areas such as navigation, where it is important to recognize thin objects such as wires and highly reflective surfaces such as glass.

Another challenge is that current state-of-the-art stereo methods [36, 23, 38] use an iterative structure based on stock GRU which we found to be a bottleneck for the iterative model designed for stereo matching. A more efficient iterative structure is needed for performance improvements.

The other key issue is that learning-based algorithms are often not as effective as on specific datasets when applied to real-world scenarios due to their limited generalization capabilities [39].

In this work, we propose EAI-Stereo (Error Aware Iterative Stereo), a new end-to-end data-driven method for stereo matching.

The major contributions of this paper can be summarized as follows:

1. We propose an error-aware refinement module that combines left-right warping with learning-based upsampling. By incorporating the original left image that contains more high-frequency information and explicit calculating error maps, our refinement module enables the network to better cope with overexposure, underexposure as well as weak textures and allows the network to learn error correction capabilities which allows EAI-Stereo to produce extreme details and sharp edges. The learning-based upsampling method in the module can provide more refined upsampling results compared to

bilinear interpolation. We have carefully studied the impact of the module's microstructure on performance. From our experiments, we find that the structure improves generalization ability while improving performance. This approach is highly general and can be applied to all models that produce disparity or depth maps.
2. We propose an efficient iterative update module, called Multiscale Wide-LSTM, which can efficiently combine multi-scale information from feature extraction, cost volume, and current state, thus enhancing the information transfer between each iteration.
3. We propose a flexible overall structure that can balance inference speed and accuracy. The tradeoff could be done without retraining the network or even at run time. The number of iterations can also be determined dynamically based on the minimum frame rate.

2 Related Works

2.1 Data-driven Stereo Matching

Recently, data-driven methods dominate the field of stereo matching. Zbontar and LeCun proposed the first deep learning stereo matching method [46]. Mayer et al. proposed DispNetC [24], the first end-to-end stereo matching network.

In order to improve accuracy, 3D convolution was adopted by various of works [2, 10, 49, 18, 42]. Chang et al. propose PSMNet [2], a pyramid stereo matching network consisting of spatial pyramid pooling and several 3D convolutional layers. Taking advantage of the strong regularization effect of 3D convolution, PSMNet outperformed other methods at that time while 3D convolutions are very computationally expensive. To further increase accuracy, Zhang et al. proposes GANet [47] which approximates semi-global matching (SGM) [13] by introducing a semi-global guided aggregation (SGA) layer [47]. The accuracy is improved by cost aggregation from different directions, which improves the performance in occluded and textureless regions. However, these networks have limited ability to generalize across datasets. After training on simulated datasets, these feature maps tend to become noisy and discrete when the network is used to predict real-world scenes, and therefore output inaccurate disparity maps. To address this problem, DSMNet [48] improves the generalization ability of the network by adding domain normalization and a non-local graph-based filter, which also improves the accuracy. Shen et al. believe that the large domain differences and unbalanced disparity distribution across a variety of datasets limit the real-world performance of the model and propose CFNet [31], which introduces Cascade and Fused Cost Volume to improve the robustness of the network.

Due to the high computational cost, some methods come up with new ways to avoid the use of 3D convolutions. Xu et al. proposed AANet [41], which replaces the computationally intensive 3D convolution and improves the accuracy by using the ISA module and CSA module. While some researchers proposed a coarse-to-fine routine [37, 43, 32] to replace 3D convolution in order to further speed up inference. Tankovich et al. proposed HITNet [35], which introduced

slanted plane hypotheses that allow performing geometric warping and upsampling operations more accurately which achieve a higher level of accuracy [35].

2.2 Iterative Network

With the development of deep learning, there is a tendency to add more layers to convolutional neural networks to achieve better accuracy. However, as the network gets deeper and deeper, the computational cost and the number of parameters have greatly increased. To address the problem, Neshatpour et al. proposed ICNN [27] (Iterative Implementation of Convolutional Neural Networks) which replaces the single heavy feedforward network with a series of smaller networks executed sequentially. Since many images are detected in early iterations, this method draws much less computational complexity.

IRR [16] first introduce a recurrent network for Optic Flow, which uses FlowNetS [6] or PWC-Net [32] as its recurrent module. This enables IRR to be able to achieve better performance by increasing its iterations. However, both FlowNetS and PWC-Net are relatively heavy when used as iterative modules, which limits the number of iterations. To address this problem, RAFT [36], proposed by Teed et al. uses GRU [3] as its iterative module to update the flow predictions and achieve state-of-the-art performance in optic flow. It is proved that RAFT has strong cross-dataset generalization ability while keeping a high efficiency in inference time [36].

In our work, we found that the stock GRU is becoming a bottleneck for the iterative model designed for stereo matching. To alleviate this problem, we proposed an improved iterative module to achieve better performance.

3 Approach

Our network takes a pair of rectified images I_l and I_r as input. Then the features are extracted and injected into the cost volume. The Multiscale Iterative Module retrieves data from the cost volume and iterates to update the disparity map. Finally, the iterated 1/4 resolution disparity map is fed into the Error Aware Refinement module, which can perform learned upsampling and error-aware correction to obtain the final disparity map.

3.1 Multi-scale Feature Extractor

We use a ResNet-like network [11] as our feature extractor, feature maps of a pair of images I_l and I_r are extracted using two shared-weight extractors which are used to construct a 3D correlation volume following RAFT-Stereo [23]. The network consists of a sequence of residual blocks and then followed by two downsampling layers which are used to provide multi-scale information F_h, F_m and F_l for the following iterative Wide-LSTM modules. The spatial sizes of the features maps F_h, F_m and F_l are 1/4, 1/8 and 1/16 of the original input image size.

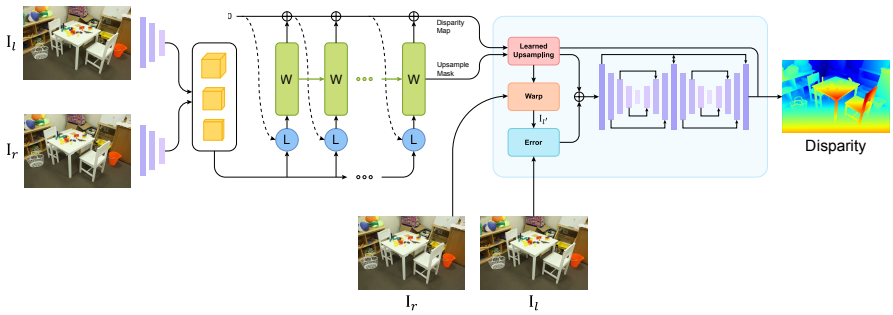

Fig. 1. The overall structure of EAI-Stereo. The left and right images are extracted by a weight-sharing feature extractor and the features are injected into the multiscale correlation volume. The following Wide-LSTM Module combines information from correlation volume, the previous iteration and the current disparity map to produce a disparity map and an unsampling mask which are used for our Error Aware Refinement. The refinement module upsamples the disparity map and then uses it to warp the right image to the left and calculate the error map. Then the combined information is fed to the hourglass models to output the final refined disparity map.

3.2 Iterative Multiscale Wide-LSTM Network

For iterative networks, the design of the iteration module has a significant impact on the network performance. For image tasks, most of the models take the stock GRU [3] as their iterative module. However, in our research, we found that the performance of the network could be increased by improving the iterative module. To address this problem, we propose an efficient iterative update module named Multiscale Wide-LSTM that can efficiently combine the information from feature extraction, cost volume, and current state, which also enhances the information transfer between each iteration. Experiments show that our model increases performance with a minor computational cost increase.

The network predicts a sequence of disparity maps $\{d_1, \ldots, d_n\}$ with an initial disparity map $d_0 = 0$. And the first hidden state h_0 is initialized using the information extracted by feature extractors. For each iteration, the LSTM module takes the previous hidden state h_{i-1}, the previous state of the disparity map d_{i-1}, and the information from the feature extraction F as inputs and then outputs Δd which adds to the current disparity: $d_i = d_{i-1} + \Delta d$. After n iterations, the iteration result d_n is fed into the refinement module for the final disparity map $d_{refined}$. We supervised our network by the following equation:

$$L_{regress} = \sum_{i=1}^{n-1} \gamma^{n-i} ||d_{gt} - d_i||_1 + ||d_{gt} - d_{refined}||_1, \; where \, \gamma = 0.9. \quad (1)$$

Multiscale Iterative Module. In our study, we found that for image tasks, the width of information transfer between iterative modules affects the model perfor-

mance. Therefore, we widen the iterative modules in our design. In our module, each of the three submodules of different scales establishes two data paths, C and h with the preceding and following iterative modules. Where h contains the information to update the disparity map for every iteration. While C extends the data path and carries extra semantic information between iterations to improve the efficiency of the iterative network. For the submodules themselves, the three different scales also interact through upsampling and downsampling to share data, thus increasing the width of information interaction and thus improving multiscale performance. Specifically, the lowest resolution mutual Conv-LSTM Cell is fused across scales by introducing features of the medium resolution, the medium resolution Conv-LSTM Cell is fused by introducing features of both low and high resolution, and the highest resolution cell is fused by introducing features of medium resolution.

The multiscale fusion mechanism follows the following formulas:

$$C_l, h_l = MutualLSTMCell(C_{l_{prev}}, h_{l_{prev}}, ctx, pool(h_{m_{prev}})) \tag{2}$$

$$C_m, h_m = CLSTMCell(C_{m_{prev}}, h_{m_{prev}}, ctx, pool(h_{h_{prev}}), interp(h_{l_{prev}})) \tag{3}$$

$$C_h, h_h = CLSTMCell(C_{l_{prev}}, h_{l_{prev}}, ctx, disp, interp(h_{m_{prev}})) \tag{4}$$

where subscript l, m and h denote low, middle and high resolution respectively. $CLSTMCell$ is short for Conv-LSTM Cell. The low-resolution MutualLSTM-Cell not only takes $C_{l_{prev}}$ and $h_{l_{prev}}$ as input but also makes use of the features from downsampled middle resolution. The middle-resolution ConvLSTM-Cell takes advantage of using both downsampled high-resolution features and upsampled low-resolution features. For the highest resolution, the module not only makes use of upsampled middle resolution but also takes $disp$ as input.

In general, low-resolution features have a larger perceptual field, which helps to improve the matching accuracy in textureless regions, while high-resolution features contain more high-frequency details, the combined use of this information can increase the perceptual field without adding much computational cost, thus improving the results. Another advantage of using multiscale is that we can use different iterative submodules at each scale, and the lower resolution feature maps have fewer pixels, which allows relatively time-consuming operations to be performed. In our model, the Mutual Conv-LSTM Cell is used only in the 1/16 resolution module. Experimental results show that using this module only at low resolution improves the performance with little change in computational cost.

Mutual Conv-LSTM Cell. To further improve the performance of the iterative network, improvements have also been made to the cells that make up the iterative module. Currently, the use of iterative networks to process image data is gaining popularity. However, most of the networks simply use GRU [3] cell and replace the fully connected layers in them with convolutional layers. However, according to our observation, widening and increasing the hidden state of the network can improve the performance very well, so we use the LSTM [21] which has a performance improvement in this task with little difference in computational cost as our baseline.

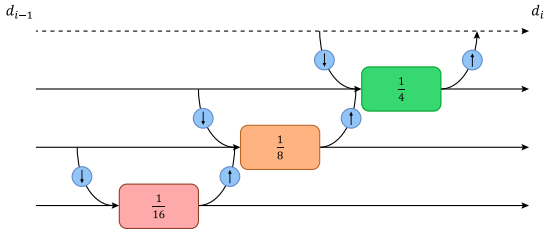

Fig. 2. Structure of Multiscale Iterative Module. Instead of fully connected fusion, our model transfer information between adjacent resolutions.

In ordinary LSTM [14], the input and hidden states do not interact much, but simply perform concatenation operations and then followed by various gate operations, which do not make good use of the input and hidden state information. In the field of natural language processing, there are several attempts to modify the LSTM cell to get better results. Inspired by the Multiplicative LSTM [21] and the MOGRIFIER LSTM [25], we propose the Mutual Conv-LSTM Cell.

In the Mutual Conv-LSTM Cell, the input x_t and the hidden state h_{t-1} interact following the formulas:

$$x_t^i = \delta Sigmoid(W_{conv_x} h_{t-1}^{i-1}) \odot x_t^{i-2}, for\ i \in \{x | x \leq n, x\%2 \neq 0\} \quad (5)$$

$$h_{t-1}^i = \delta Sigmoid(W_{conv_h} x_t^{i-1}) \odot x_{t-1}^{i-2}, for\ i \in \{x | x \leq n, x\%2 = 0\} \quad (6)$$

where n denotes the number of interactions between the input and the hidden state, W_{conv_x} and W_{conv_h} denote the weights of the two convolutional layers, δ denotes a constant to balance the distribution and is set to 2 in our experiments.

As depicted in Figure 3, the convolution-processed feature maps of h_{t-1}^i are element-wise multiplied with x_t^i to generate a new hidden state. Similarly, the convolution-processed feature maps of x_t^i are element-wise multiplied with h_{t-1}^i to generate a new input. After interactions, the generated input x_t^i and hidden state h_{t-1}^n are processed with a procedure similar to a regular LSTM module following the equations:

$$f_t = Sigmoid(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (7)$$

$$i_t = Sigmoid(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (8)$$

$$\tilde{C}_t = tanh(W_i \cdot [h_{t-1}, x_t] + b_C) \quad (9)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (10)$$

$$o_t = Sigmoid(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (11)$$

$$h_t = o_t * tanh(C_t) \quad (12)$$

The features from the input and hidden state are fully fused by multiple interactions, and the effective part of the features in both are retained and enhanced.

In our model, to lower the parameters as well as increase inference speed, the Mutual Conv-LSTM Cell is only applied to the lowest resolution. Ablation experiments show that the module brings significant performance improvement.

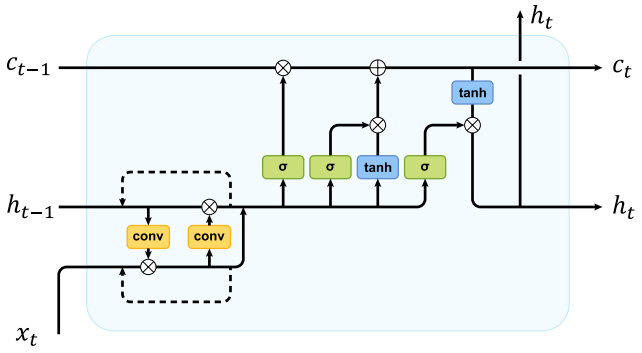

Fig. 3. Mutual Conv-LSTM Cell. The input and hidden state gate each other with convolution-processed features for n times. After interactions, the generated input and hidden state are processed with a procedure similar to a regular LSTM.

3.3 Error Aware Refinement

As we motioned before, a large amount of high-frequency information is ignored in the former structure of the model. In our refinement model, we aim to make full use of the former information and incorporate the high-frequency information from the original left image.

To make full use of the former information, we use learned upsampling to upsample the 1/4 resolution raw disparity map predicted by the LSTM network. Following RAFT[36], the highest resolution output is fed to a series of convolutional layers and generates an upsampling mask which is used to provide information to the convex upsampling. This method is proved to be much more efficient than bilinear upsampling. After the Learned Upsampling process, we get the disparity map of the same size as the original image. However, the disparity map is not error-aware processed at this point.

To incorporate the high-frequency information from the original left image and alleviate the problem of false matching, in Error Aware Module, we perform error perception by the following equations:

$$I_l^{'} = warp(I_r, disp) \tag{13}$$

$$e = I_l^{'} - I_l \tag{14}$$

$$I_{fuse} = Conv_{3\times3}([e, I_l]) \tag{15}$$

$$disp^{'} = hourglass([I_{fuse}, Conv_{3\times3}(disp)]) \tag{16}$$

where $I_l^{'}$ denotes the warped right image, e denotes the reprojection error, $disp^{'}$ denotes the refined disparity map.

The disparities are the correspondences between the left and right images. By using the warp method, the reconstructed left image can be calculated using the right image and the disparity map. Then subtraction is performed to get the error map which we called explicit error. As we motioned before, a large amount of high-frequency information is ignored during the former process. To alleviate this problem, we introduce the left image directly into the module, which composes of our implicit error. These two different forms of error improve the performance of our model in different aspects. We analyze them in the subsequent experiment section. By introducing more high-frequency information from the original left image, our model is therefore capable of recovering extreme details and sharp edges. Comparisons with the state-of-the-art methods are shown in Figure 6.

In the Hourglass model, we reduced the number of its same-resolution convolution layers to streamline it. We tried deformable convolution [4] and dilation convolution [45], experimental results show that using deformable convolution is not as effective as dilation convolution. The reason behind it may be that deformable convolution is relatively weak for different scenes, while dilation convolution has a larger perceptual field and is capable of long-distance modeling.

We have carefully studied the impact of the module's microstructure on performance. Details are shown in Table 5(a).

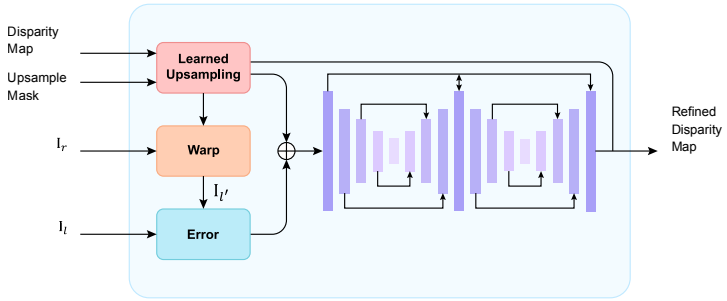

Fig. 4. Error Aware Refinement. The error map, the left image, and the original disparity map are passed into the hourglass to calculate the refined disparity map.

4 Experiments

EAI-Stereo is implemented in PyTorch and trained with two Tesla A100 GPUs. All models are trained using AdamW optimizer with a weight decay of $1e^{-5}$. Warm-up takes 1% of the whole training schedule. We used data augmentation in all experiments. The methods are saturation change, image perturbance, and random scales. For all the pretraining, we train our model on Scene Flow for 200k iterations with a learning rate of $2e^{-4}$.

We evaluate our EAI-Stereo with different settings using Scene Flow [24], KITTI-2015 [26], ETH3D [30] and Middlebury [28] datasets.

4.1 Middlebury

EAI-Stereo ranks 1^{st} on the Middlebury test set, with an average error of 1.92% for all pixels, outperforming the next best method by 8.6%. Our method outperforms state-of-the-art methods on most of the metrics. See Table 1.

We fine-tune our model on the 23 Middlebury training images with a maximum learning rate of 2e-5 for 4000 iterations. Though Middlebury provides images with different color temperatures, we only use the standard images for training. Experiments show that EAI-Stereo is robust for various lighting conditions with simple data augmentation methods.

We also evaluate our EAI-Stereo on the Middlebury dataset without any fine-tuning, results are shown in Table 4, which prove the strong cross-domain performance of our model.

Table 1. Results on the Middlebury stereo dataset V3 [28] leaderboard.

Method	bad 0.5 nonocc (%)	bad 1.0 nonocc (%)	bad 2.0 nonocc (%)	avgerr nonocc (%)	avgerr all (%)
LocalExp [34]	38.7	13.9	5.43	2.24	5.13
NOSS-ROB [17]	38.2	13.2	5.01	2.08	4.80
HITNet [35]	34.2	13.3	6.46	1.71	3.29
RAFT-Stereo [23]	27.2	9.37	4.74	1.27	2.71
CREStereo [22]	28.0	8.25	3.71	1.15	2.10
EAI-Stereo (Ours)	**25.1**	**7.81**	**3.68**	**1.09**	**1.92**

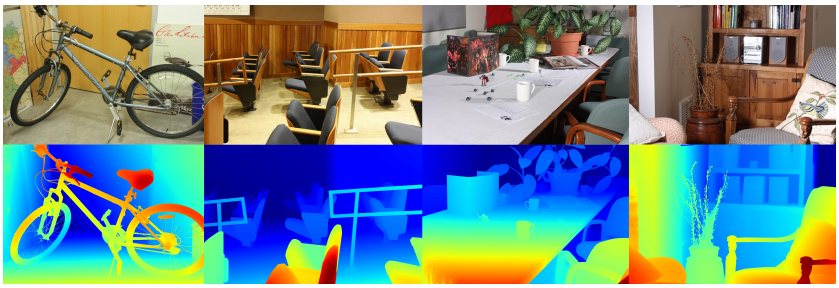

Fig. 5. Results on Middlebury dataset. Our EAI-Stereo recovers extreme details such as the spokes of the bicycle, toys on the table, and the subtle structures of plants.

4.2 ETH3D

For the ETH3D [30] dataset, we did not perform further fine-tune. Since the images are all in grayscale with many overexposed and underexposed areas. We use data augmentation to simulate the situation by setting saturation to 0, adjusting image gamma between 0.5 and 2.0, and adjusting image gain between 0.8 and 1.2.

At the time of writing this paper, EAI-Stereo ranks 1st on the ETH3D Stereo benchmark for 50% quantile metric and second for 0.5px error rate (see Table 2) among all published methods.

Table 2. Results on the ETH3D [30] leaderboard.

Method	bad 0.5 (%)	bad 1.0 (%)	50% quantile
AANet_RVC [41]	13.16	5.01	0.16
CFNet [31]	9.87	3.31	0.14
ACVNet [40]	10.36	2.58	0.15
HIT-Net [35]	7.83	2.79	0.10
RAFT-Stereo [23]	7.04	2.44	0.10
EAI-Stereo (Ours)	**5.21**	**2.31**	**0.09**

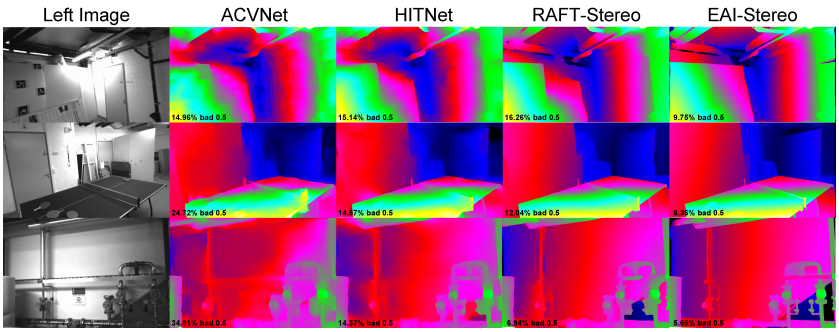

Fig. 6. Results on ETH3D compared to state-of-the-art methods. Bad 0.5 error is reported at the corners. EAI-Stereo shows advantages in recovering extreme details and sharp edges of the scenes such as the detailed structure of the pipes and valves. Our model is also capable of handling extreme overexposure and underexposure such as the reflective cardboard and pitch-black pipes on the roof of the top image.

4.3 KITTI-2015

We trained our model on the Scene Flow dataset and then fine-tuned our model on the KITTI training set for 6000 iterations with a batch size of 8 and a maximum learning rate of 2e-5. The images and disparity maps are randomly cropped with a resolution of 320×1024. We iterate the Multiscale Iterative Module 32 times.

Ground Truth values in the KITTI dataset are sparse and only cover the lower part of the image. We can observe from Figure 7 that other methods fail to generalize in the upper part while our method recovers extreme details and sharp edges which proves the strong generalization performance of our model.

Table 3. Results on the KITTI-2015 [26] leaderboard. Only published results are included. The best results for each metric are bolded, second best are underlined.

Method	D1-all	D1-fg	D1-bg
AcfNet [49]	1.89	3.80	1.51
AMNet [7]	<u>1.82</u>	3.43	1.53
OptStereo [39]	<u>1.82</u>	3.43	<u>1.50</u>
GANet-deep [47]	**1.81**	3.46	**1.48**
RAFT-Stereo [23]	1.96	**2.89**	1.75
HITNet [35]	1.98	3.20	1.74
CFNet [31]	1.88	3.56	1.54
EAI-Stereo (Ours)	**1.81**	<u>2.92</u>	1.59

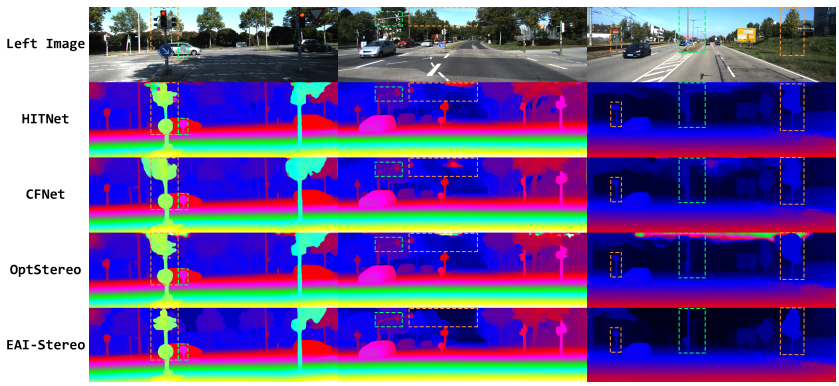

Fig. 7. Results on the KITTI-2015 test set compared to state-of-the-art methods. EAI-Stereo shows an advantage in recovering extreme details and sharp edges of the scenes. Zoom in for a better view. RAFT-Stereo[23] is not included in this comparison because it does not have an official submission to the benchmark.

4.4 Cross-domain Generalization

Generalization performance is crucial for real-world applications. Towards this end, we evaluated our model on three public datasets. We train our model on the Scene Flow dataset using the strategy exactly the same as pretrain and then use the weight for evaluation directly. In Table 4, we compare our model with some state-of-the-art methods and some classical methods. The comparison shows that our method outperforms DSMNet and CFNet, which are specifically designed for generalization performance, by a notable margin.

4.5 Ablations

We evaluate the performance of EAI-Stereo with different settings, including different architectures and different numbers of iterations.

Iterative Multiscale Wide-LSTM Network. We observe a significant performance leap (10.14% D1 error decrease on Scene Flow validation set and 4.80% EPE decrease on KITTI validation set) by using the wide LSTM module. Most iterative networks use GRUs as their iterative modules. However, we found that the performance of the network can be increased by refining the iterative module. A comparison between the GRU-based network and our iterative multiscale wide LSTM network is shown in Table 5(c). Using Mutual Conv-LSTM Cell at the lowest resolution can further improve the performance of the model. As shown in Table 5(c), this module led to 1.4% D1 error decrease on the Scene Flow validation set and 2.58% D1 error decrease on the KITTI validation set.

Error Aware Refinement. The Error Aware Refinement module is used to do the upsampling and refinement work. To verify and analyze the effects of our Error Aware Refinement module, we evaluate the different structures of the refinement module, and the results are shown in Table 5(c). Compared with the Wide LSTM baseline, Dilation Refinement decreases the D1-error by 2.81% on the Scene Flow validation set and 12.39% EPE (end-point-error) decrease on the KITTI validation set. Using deformable convolution is not as effective as dilation convolution, and we think the reason behind it may be that deformable convolution is relatively weak for different scenes, while dilated convolution has a larger perceptual field and is capable of long-distance modeling. Detailed comparisons are shown in Table 5(c).

Number of iterations. Due to the structural improvements of our model, inference can be accelerated by reducing iterations while maintaining competitive performance. Since the model requires only a single training, the number of iterations can be adjusted after training, which increases the flexibility of the model. In practical applications, the number of iterations can also be inferred in the running state by giving a minimum frame rate, which is useful for scenarios with real-time requirements. Details are shown in Table 5(b).

5 Conclusion

We have proposed a novel error-aware iterative network for stereo matching. Several experiments were conducted to determine the structure of the module. Experiment results show that our model performs well on various datasets for both speed and accuracy while having a strong generalization performance.

Table 4. Cross-domain generalization experiments.

Method	KITTI2015 bad 3.0 (%)	Middlebury bad 2.0 (%)	ETH3D bad 1.0 (%)
PSMNet [2]	16.3	39.5	23.8
GANet [47]	11.7	32.2	14.1
DSMNet [48]	6.5	21.8	6.2
CFNet [31]	-	28.2	5.8
EAI-Stereo(Ours)	**6.1**	**14.5**	**3.3**

Table 5. Ablations Experiments.

(a) Ablations on refinement microstructures.

Hourglass	Error	Left image	Scene Flow D1	KITTI EPE	KITTI D1
			5.88	0.47	1.11
✓			5.85	0.47	0.89
✓	✓		5.84	0.40	0.85
✓		✓	5.76	0.41	0.89
✓	✓	✓	5.74	0.40	0.85

(b) Inference time.

Iters	Scene Flow EPE	Scene Flow D1	Time (ms)
5	0.596	7.326	92
7	0.539	6.527	100
10	0.510	6.046	132
16	0.495	5.821	154
32	0.491	5.661	236

(c) Ablations on different structures of the model.

Model	Conv GRU	Wide LSTM	Deform Refine	Dilation Refine	Mutual Conv LSTM	Scene Flow D1	KITTI EPE	KITTI D1
Baseline	✓					6.542	0.491	1.290
Wide LSTM		✓				5.879	0.468	1.108
EAI-Deform		✓	✓			5.840	0.410	0.850
EAI-Dilation		✓		✓		5.741	0.401	0.854
EAI-Mutual		✓		✓	✓	5.661	0.397	0.832

Acknowledgements This work is supported by Shenzhen Fundamental Research Program (JCYJ20180503182133411).

References

1. Bleyer, M., Gelautz, M.: Simple but effective tree structures for dynamic programming-based stereo matching. In: International Conference on Computer Vision Theory and Applications. vol. 2, pp. 415–422. SCITEPRESS (2008)
2. Chang, J.R., Chen, Y.S.: Pyramid stereo matching network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5410–5418 (2018)
3. Cho, K., van Merriënboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: Encoder–decoder approaches. In: Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation. pp. 103–111. Association for Computational Linguistics, Doha, Qatar (Oct 2014). https://doi.org/10.3115/v1/W14-4012, https://aclanthology.org/W14-4012
4. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp. 764–773 (2017)
5. Dinh, V.Q., Munir, F., Sheri, A.M., Jeon, M.: Disparity estimation using stereo images with different focal lengths. IEEE Transactions on Intelligent Transportation Systems **21**(12), 5258–5270 (2019)
6. Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2758–2766 (2015)
7. Du, X., El-Khamy, M., Lee, J.: Amnet: Deep atrous multiscale stereo disparity estimation networks. arXiv preprint arXiv:1904.09099 (2019)
8. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient belief propagation for early vision. International journal of computer vision **70**(1), 41–54 (2006)
9. Fife, W.S., Archibald, J.K.: Improved census transforms for resource-optimized stereo vision. IEEE Transactions on Circuits and Systems for Video Technology **23**(1), 60–73 (2012)
10. 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 (CVPR). pp. 3273–3282 (2019)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90
12. Heo, Y.S., Lee, K.M., Lee, S.U.: Robust stereo matching using adaptive normalized cross-correlation. IEEE Transactions on pattern analysis and machine intelligence **33**(4), 807–822 (2010)
13. Hirschmuller, H.: Stereo processing by semiglobal matching and mutual information. IEEE Transactions on pattern analysis and machine intelligence **30**(2), 328–341 (2007)
14. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
15. Hosni, A., Rhemann, C., Bleyer, M., Rother, C., Gelautz, M.: Fast cost-volume filtering for visual correspondence and beyond. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**(2), 504–511 (2012)
16. Hur, J., Roth, S.: Iterative residual refinement for joint optical flow and occlusion estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)

17. Ji, P., Li, J., Li, H., Liu, X.: Superpixel alpha-expansion and normal adjustment for stereo matching. Journal of Visual Communication and Image Representation **79**, 103238 (2021)
18. 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: Proceedings of the IEEE international conference on computer vision. pp. 66–75 (2017)
19. Klaus, A., Sormann, M., Karner, K.: Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In: 18th International Conference on Pattern Recognition (ICPR'06). vol. 3, pp. 15–18. IEEE (2006)
20. Kolmogorov, V., Zabih, R.: Computing visual correspondence with occlusions using graph cuts. In: Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001. vol. 2, pp. 508–515. IEEE (2001)
21. Krause, B., Lu, L., Murray, I., Renals, S.: Multiplicative lstm for sequence modelling. arXiv preprint arXiv:1609.07959 (2016)
22. Li, J., Wang, P., Xiong, P., Cai, T., Yan, Z., Yang, L., Liu, J., Fan, H., Liu, S.: Practical stereo matching via cascaded recurrent network with adaptive correlation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 16263–16272 (2022)
23. Lipson, L., Teed, Z., Deng, J.: Raft-stereo: Multilevel recurrent field transforms for stereo matching. In: 2021 International Conference on 3D Vision (3DV). pp. 218–227. IEEE (2021)
24. 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: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4040–4048 (2016)
25. Melis, G., Kočiskỳ, T., Blunsom, P.: Mogrifier lstm. arXiv preprint arXiv:1909.01792 (2019)
26. Menze, M., Geiger, A.: Object scene flow for autonomous vehicles. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3061–3070 (2015)
27. Neshatpour, K., Behnia, F., Homayoun, H., Sasan, A.: Icnn: An iterative implementation of convolutional neural networks to enable energy and computational complexity aware dynamic approximation. In: 2018 Design, Automation & Test in Europe Conference & Exhibition (DATE). pp. 551–556. IEEE (2018)
28. Scharstein, D., Hirschmüller, H., Kitajima, Y., Krathwohl, G., Nešić, N., Wang, X., Westling, P.: High-resolution stereo datasets with subpixel-accurate ground truth. In: Jiang, X., Hornegger, J., Koch, R. (eds.) Pattern Recognition. pp. 31–42. Springer International Publishing, Cham (2014)
29. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International journal of computer vision **47**(1), 7–42 (2002)
30. Schöps, T., Schönberger, 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 (CVPR) (2017)
31. Shen, Z., Dai, Y., Rao, Z.: Cfnet: Cascade and fused cost volume for robust stereo matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 13906–13915 (2021)

32. 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). pp. 8934–8943 (2018)
33. Sun, J., Zheng, N.N., Shum, H.Y.: Stereo matching using belief propagation. IEEE Transactions on pattern analysis and machine intelligence **25**(7), 787–800 (2003)
34. Taniai, T., Matsushita, Y., Sato, Y., Naemura, T.: Continuous 3d label stereo matching using local expansion moves. IEEE transactions on pattern analysis and machine intelligence **40**(11), 2725–2739 (2017)
35. Tankovich, V., Hane, C., Zhang, Y., Kowdle, A., Fanello, S., Bouaziz, S.: Hitnet: Hierarchical iterative tile refinement network for real-time stereo matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 14362–14372 (2021)
36. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: European conference on computer vision. pp. 402–419. Springer (2020)
37. 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 (CVPR). pp. 195–204 (2019)
38. Wang, F., Galliani, S., Vogel, C., Pollefeys, M.: Itermvs: Iterative probability estimation for efficient multi-view stereo (2022)
39. Wang, H., Fan, R., Cai, P., Liu, M.: Pvstereo: Pyramid voting module for end-to-end self-supervised stereo matching. IEEE Robotics and Automation Letters **6**(3), 4353–4360 (2021)
40. Xu, G., Cheng, J., Guo, P., Yang, X.: Acvnet: Attention concatenation volume for accurate and efficient stereo matching. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
41. Xu, H., Zhang, J.: Aanet: Adaptive aggregation network for efficient stereo matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1959–1968 (2020)
42. Yao, Y., Luo, Z., Li, S., Fang, T., Quan, L.: Mvsnet: Depth inference for unstructured multi-view stereo. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 767–783 (2018)
43. 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 (CVPR). pp. 6044–6053 (2019)
44. Yoon, K.J., Kweon, I.S.: Adaptive support-weight approach for correspondence search. IEEE transactions on pattern analysis and machine intelligence **28**(4), 650–656 (2006)
45. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122 (2015)
46. Zbontar, J., LeCun, Y.: Computing the stereo matching cost with a convolutional neural network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1592–1599 (2015)
47. Zhang, F., Prisacariu, V., Yang, R., Torr, P.S.: Ga-net: Guided aggregation net for end-to-end stereo matching. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 185–194. IEEE Computer Society, Los Alamitos, CA, USA (jun 2019). https://doi.org/10.1109/CVPR.2019.00027, https://doi.ieeecomputersociety.org/10.1109/CVPR.2019.00027
48. Zhang, F., Qi, X., Yang, R., Prisacariu, V., Wah, B., Torr, P.: Domain-invariant stereo matching networks. In: European Conference on Computer Vision. pp. 420–439. Springer (2020)

49. Zhang, Y., Chen, Y., Bai, X., Yu, S., Yu, K., Li, Z., Yang, K.: Adaptive unimodal cost volume filtering for deep stereo matching. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 12926–12934 (2020)