

# AANet: Adaptive Aggregation Network for Efficient Stereo Matching

Haofei Xu Juyong Zhang\*

University of Science and Technology of China

xhf@mail.ustc.edu.cn, juyong@ustc.edu.cn

## Abstract

Despite the remarkable progress made by learning based stereo matching algorithms, one key challenge remains unsolved. Current state-of-the-art stereo models are mostly based on costly 3D convolutions, the cubic computational complexity and high memory consumption make it quite expensive to deploy in real-world applications. In this paper, we aim at completely replacing the commonly used 3D convolutions to achieve fast inference speed while maintaining comparable accuracy. *To this end, we first propose a sparse points based intra-scale cost aggregation method to alleviate the well-known edge-fattening issue at disparity discontinuities.* Further, we approximate traditional cross-scale cost aggregation algorithm with neural network layers to handle large textureless regions. Both modules are simple, lightweight, and complementary, leading to an effective and efficient architecture for cost aggregation. With these two modules, we can not only significantly speed up existing top-performing models (e.g.,  $41\times$  than GC-Net,  $4\times$  than PSMNet and  $38\times$  than GA-Net), but also improve the performance of fast stereo models (e.g., StereoNet). We also achieve competitive results on Scene Flow and KITTI datasets while running at 62ms, demonstrating the versatility and high efficiency of the proposed method. Our full framework is available at <https://github.com/haofeixu/aanet>.

## 1. Introduction

Estimating depth from stereo pairs is one of the most fundamental problems in computer vision [29]. The key task is to find spatial pixel correspondences, i.e., stereo matching, then depth can be recovered by triangulation. Efficient and accurate stereo matching algorithms are crucial for many real-world applications that require fast and reliable responses, such as robot navigation, augmented reality and autonomous driving.

Traditional stereo matching algorithms generally per-

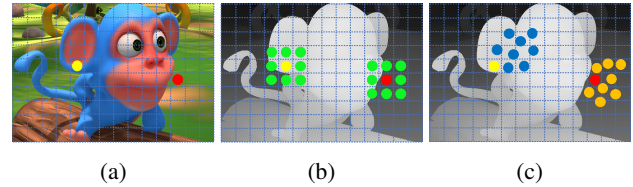


Figure 1: Illustration of the sampling locations in regular convolution based cost aggregation methods and our proposed approach, where the yellow and red points represent the locations for aggregation. (a) left image of a stereo pair. (b) fixed sampling locations in regular convolutions, also the aggregation weights are spatially shared. (c) adaptive sampling locations and position-specific aggregation weights in our approach. The background in (b) and (c) is ground truth disparity.

form a four-step pipeline: matching cost computation, cost aggregation, disparity computation and refinement, and they can be broadly classified into global and local methods [29]. Global methods usually solve an optimization problem by minimizing a global objective function that contains data and smoothness terms [31, 17], while local methods only consider neighbor information [40, 12], making themselves much faster than global methods [23, 29]. Although significant progress has been made by traditional methods, they still suffer in challenging situations like textureless regions, repetitive patterns and thin structures.

Learning based methods make use of deep neural networks to learn strong representations from data, achieving promising results even in those challenging situations. DispNetC [20] builds the first end-to-end trainable framework for disparity estimation, where a correlation layer is used to measure the similarity of left and right image features. GC-Net [14] takes a different approach by directly concatenating left and right features, and thus 3D convolutions are required to aggregate the resulting 4D cost volume. PSMNet [4] further improves GC-Net by introducing more 3D convolutions for cost aggregation and accordingly obtains better accuracy. Although state-of-the-art performance can be achieved with 3D convolutions, the high

\*Corresponding author

computational cost and memory consumption make it quite expensive to deploy in practice (for example, PSMNet costs about 4G memory and 410ms to predict a KITTI stereo pair even on high-end GPUs). The recent work, GA-Net [43], also notices the drawbacks of 3D convolutions and tries to replace them with two guided aggregation layers. However, their final model still uses fifteen 3D convolutions.

To this end, a motivating question arises: *How to achieve state-of-the-art results without any 3D convolutions while being significantly faster?* Answering this question is especially challenging due to the strong regularization provided by 3D convolutions. In this paper, we show that by designing two effective and efficient modules for cost aggregation, competitive performance can be obtained on both Scene Flow and KITTI datasets even *with simple feature correlation [20] instead of concatenation [14]*.

Specifically, we first propose a new sparse points based representation for intra-scale cost aggregation. As illustrated in Fig. 1, a set of sparse points are adaptively sampled to locate themselves in regions with similar disparities, alleviating the well-known edge-fattening issue at disparity discontinuities [29]. Moreover, such representation is flexible to sample from a large context while being much more efficient than sampling from a large window, an essential requirement for traditional local methods to obtain high-quality results [23]. We additionally learn content-adaptive weights to achieve position-specific weighting for cost aggregation, aiming to overcome the inherent drawback of spatial sharing nature in regular convolutions. We implement the above ideas with deformable convolution [45].

We further approximate traditional cross-scale cost aggregation algorithm [44] with neural network layers by constructing multi-scale cost volumes in parallel and allowing adaptive multi-scale interactions, producing accurate disparity predictions even in low-texture or textureless regions.

These two modules are simple, lightweight, and complementary, leading to an efficient architecture for cost aggregation. We also make extensive use of the key ideas in the feature extraction stage, resulting in our highly efficient and accurate Adaptive Aggregation Network (AANet). For instance, we can outperform existing top-performing models on Scene Flow dataset, while being significantly faster, e.g.,  $41\times$  than GC-Net[14],  $4\times$  than PSMNet [4] and  $38\times$  than GA-Net [43]. Our method can also be a valuable way to improve the performance of fast stereo models, e.g., StereoNet [15], which are usually based on a very low-resolution cost volume to achieve fast speed, while at the cost of sacrificing accuracy. We also achieve competitive performance on KITTI dataset while running at 62ms, demonstrating the versatility and high efficiency of the proposed method.

## 2. Related Work

This section reviews the most relevant work to ours.

**Local Cost Aggregation.** Local stereo methods (either traditional [40, 12] or 2D/3D convolution based methods [20, 14]) usually perform window based cost aggregation:

$$\tilde{C}(d, \mathbf{p}) = \sum_{\mathbf{q} \in N(\mathbf{p})} w(\mathbf{p}, \mathbf{q}) C(d, \mathbf{q}), \quad (1)$$

where  $\tilde{C}(d, \mathbf{p})$  denotes the aggregated cost at pixel  $\mathbf{p}$  for disparity candidate  $d$ , pixel  $\mathbf{q}$  belongs to the neighbors  $N(\mathbf{p})$  of  $\mathbf{p}$ ,  $w(\mathbf{p}, \mathbf{q})$  is the aggregation weight and  $C(d, \mathbf{q})$  is the raw matching cost at  $\mathbf{q}$  for disparity  $d$ . Despite the widespread and successful applications of local methods, they still have several important limitations. First and foremost, the fundamental assumption made by local methods is that all the pixels in the matching window have similar disparities. However, this assumption does not hold at disparity discontinuities, causing the well-known edge-fattening issue in object boundaries and thin structures [29, 23]. As a consequence, the weighting function  $w$  needs to be designed carefully to eliminate the influence of pixels that violate the smoothness assumption [12, 40]. While learning based methods automatically learn the aggregation weights from data, they still suffer from the inherent drawback of regular convolutions: weights are spatially shared, thus making themselves content-agnostic. Moreover, a large window size is often required to obtain high-quality results [24, 23], leading to high computational cost. Some works have been proposed to address the limitations of fixed rectangular window, e.g., using varying window size [26], multiple windows [11], or unconstrained shapes [2].

Different from existing methods, we propose a new sparse points based representation for cost aggregation. This representation is also different from [23], in which sparse points inside the matching window are regularly sampled to reduce the computational complexity. In contrast, our proposed sampling mechanism is completely unconstrained and adaptive, providing more flexibility than the regular sampling in [23]. We also learn additional content-adaptive weights to enable position-specific weighting in contrast to the spatial sharing nature of regular convolutions.

**Cross-Scale Cost Aggregation.** Traditional cross-scale cost aggregation algorithm [44] reformulates local cost aggregation from a unified optimization perspective, and shows that by enforcing multi-scale consistency on cost volumes, the final cost volume is obtained through the adaptive combination of the results of cost aggregation performed at different scales. Details are provided in the supplementary material. We approximate this conclusion with neural network layers in an end-to-end manner. Different from existing coarse-to-fine approaches [33, 39, 30], we build multi-scale cost volumes in parallel and allow adaptive multi-scale interactions. Our cross-scale aggregation architecture is also different from the very recent work [35], in which

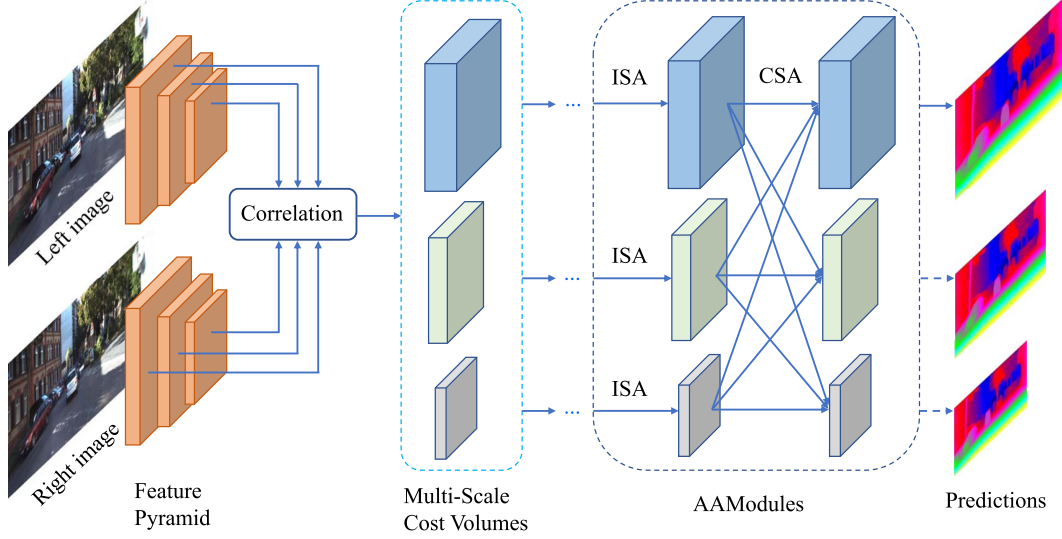


Figure 2: Overview of our proposed Adaptive Aggregation Network (AANet). Given a stereo pair, we first extract down-sampled feature pyramid at 1/3, 1/6 and 1/12 resolutions with a shared feature extractor. Then multi-scale cost volumes are constructed by correlating left and right features at corresponding scales. The raw cost volumes are aggregated by six stacked Adaptive Aggregation Modules (AAModules), where an AAModule consists of three Intra-Scale Aggregation (ISA, Sec. 3.1) modules and a Cross-Scale Aggregation (CSA, Sec. 3.2) module for three pyramid levels. Next multi-scale disparity predictions are regressed. Note that the dashed arrows are only required for training and can be removed for inference. Finally the disparity prediction at 1/3 resolution is hierarchically upsampled/refined to the original resolution. For clarity, the refinement modules are omitted in this figure, see Sec. 3.3 for details.

multi-scale cost volumes are also constructed. However, [35] fuses the cost volumes from the lowest level to the higher ones hierarchically, while ours aggregates all scale cost volumes simultaneously based on the analysis in [44].

**Stereo Matching Networks.** Existing end-to-end stereo matching networks can be broadly classified into two categories: 2D and 3D convolution based methods. They mainly differ in the way that cost volume is constructed. 2D methods [20, 18, 33] generally adopt a correlation layer [20] while 3D methods [14, 4, 25, 43, 3] mostly use direct feature concatenation [14]. An exception to concatenation based 3D methods is [8], in which group-wise correlation is proposed to reduce the information loss of full correlation [20]. In terms of performance, 3D methods usually outperform 2D methods by a large margin on popular benchmarks (e.g., Scene Flow [20] and KITTI [22]), but the running speed is considerably slower. In this paper, we aim at significantly speeding up existing top-performing methods while maintaining comparable performance. The very recent work, DeepPruner [6], shares a similar goal with us to build efficient stereo models. They propose to reduce the disparity search range by a differentiable PatchMatch [1] module, and thus a compact cost volume is constructed. In contrast, we aim at reducing the sampling complexity and improving the sampling flexibility in cost aggregation, which works on different aspects, and both methods can be

complementary to each other.

**Deformable Convolution.** Deformable convolution [5, 45] is initially designed to enhance standard convolution’s capability of modeling geometric transformations, and commonly used as backbone for object detection and semantic/instance segmentation tasks. We instead take a new perspective of traditional stereo methods and propose an adaptive sampling scheme for efficient and flexible cost aggregation. Since the resulting formulation is similar to deformable convolution, we adopt it in our implementation.

### 3. Method

Given a rectified image pair  $I_l$  and  $I_r$ , we first extract downsampled feature pyramid  $\{F_l^s\}_{s=1}^S$  and  $\{F_r^s\}_{s=1}^S$  with a shared feature extractor, where  $S$  denotes the number of scales,  $s$  is the scale index, and  $s = 1$  represents the highest scale. Then multi-scale 3D cost volumes  $\{C^s\}_{s=1}^S$  are constructed by correlating left and right image features at corresponding scales, similar to DispNetC [20]:

$$C^s(d, h, w) = \frac{1}{N} \langle F_l^s(h, w), F_r^s(h, w - d) \rangle, \quad (2)$$

where  $\langle \cdot, \cdot \rangle$  denotes the inner product of two feature vectors and  $N$  is the channel number of extracted features.  $C^s(d, h, w)$  is the matching cost at location  $(h, w)$  for disparity candidate  $d$ . The raw cost volumes  $\{C^s\}_{s=1}^S$  are

then aggregated with several stacked Adaptive Aggregation Modules (AAModules), where an AAModule consists of  $S$  adaptive Intra-Scale Aggregation (ISA) modules and an adaptive Cross-Scale Aggregation (CSA) module for  $S$  pyramid levels. Finally, the predicted low-resolution disparity is hierarchically upsampled to the original resolution with the refinement modules. All disparity predictions are supervised with ground truth when training, while only the last disparity prediction is required for inference. Fig. 2 provides an overview of our proposed Adaptive Aggregation Network (AANet). In the following, we introduce the ISA and CSA modules in detail.

### 3.1. Adaptive Intra-Scale Aggregation

To alleviate the well-known edge-fattening issue at disparity discontinuities, we propose a sparse points based representation for efficient and flexible cost aggregation. Since the resulting formulation is similar to deformable convolution, we adopt it in our implementation.

Specifically, for cost volume  $C \in \mathbb{R}^{D \times H \times W}$  at a certain scale, where  $D, H, W$  represents the maximum disparity, height and width, respectively, the proposed cost aggregation strategy is defined as

$$\tilde{C}(d, \mathbf{p}) = \sum_{k=1}^{K^2} w_k \cdot C(d, \mathbf{p} + \mathbf{p}_k + \Delta \mathbf{p}_k), \quad (3)$$

where  $\tilde{C}(d, \mathbf{p})$  denotes the aggregated cost at pixel  $\mathbf{p}$  for disparity candidate  $d$ ,  $K^2$  is the number of sampling points ( $K = 3$  in our paper),  $w_k$  is the aggregation weight for  $k$ -th point,  $\mathbf{p}_k$  is the fixed offset to  $\mathbf{p}$  in window based cost aggregation approaches. Our key difference from previous stereo works is that we learn additional offset  $\Delta \mathbf{p}_k$  to regular sampling location  $\mathbf{p} + \mathbf{p}_k$ , thus enabling adaptive sampling for efficient and flexible cost aggregation, leading to high-quality results in object boundaries and thin structures.

However, in the context of learning, the spatial sharing nature of regular convolution weights  $\{w_k\}_{k=1}^{K^2}$  makes themselves content-agnostic. We further learn position-specific weights  $\{m_k\}_{k=1}^{K^2}$  (i.e., modulation in [45], they also have effects of controlling the relative influence of the sampling points) for each pixel location  $\mathbf{p}$  to achieve content-adaptive cost aggregation:

$$\tilde{C}(d, \mathbf{p}) = \sum_{k=1}^{K^2} w_k \cdot C(d, \mathbf{p} + \mathbf{p}_k + \Delta \mathbf{p}_k) \cdot m_k. \quad (4)$$

We implement Eq. (4) with deformable convolution [45], both  $\Delta \mathbf{p}_k$  and  $m_k$  are obtained by a separate convolution layer applied over the input cost volume  $C$ . The original formulation of deformable convolution assumes the offsets  $\Delta \mathbf{p}_k$  and weights  $m_k$  are shared by each channel (i.e., disparity candidate  $d$  in this paper), we further evenly divide all

disparity candidates into  $G$  groups, and share  $\Delta \mathbf{p}_k$  and  $m_k$  within each group. Dilated convolution [41] is also used for deformable convolution to introduce more flexibility. We set  $G = 2$  and the dilation rate to 2 in this paper.

We build an Intra-Scale Aggregation (ISA) module with a stack of 3 layers and a residual connection [9]. The three layers are  $1 \times 1$ ,  $3 \times 3$  and  $1 \times 1$  convolutions, where the  $3 \times 3$  convolution is a deformable convolution. This design is similar to the bottleneck in [9], but we always keep the channels constant (equals to the number of disparity candidates). That is, we keep reasoning about disparity candidates, similar to traditional cost aggregation methods.

### 3.2. Adaptive Cross-Scale Aggregation

In low-texture or textureless regions, searching the correspondence at the coarse scale can be beneficial [21], as the texture information will be more discriminative under the same patch size when an image is downsampled. A similar observation has also been made in [36]. Therefore, multi-scale interactions are introduced in traditional cross-scale cost aggregation algorithm [44].

The analysis in [44] shows that the final cost volume is obtained through the adaptive combination of the results of cost aggregation performed at different scales (details are given in the supplementary material). We thus approximate this algorithm with

$$\hat{C}^s = \sum_{k=1}^S f_k(\tilde{C}^k), \quad s = 1, 2, \dots, S, \quad (5)$$

where  $\hat{C}$  is the resulting cost volume after cross-scale cost aggregation,  $\tilde{C}^k$  is the intra-scale aggregated cost volume at scale  $k$ , for example, with the algorithm in Sec. 3.1, and  $f_k$  is a general function to enable the adaptive combination of cost volumes at each scale. We adopt the definition of  $f_k$  from HRNet [32], a recent work for human pose estimation, which depends on the resolutions of cost volumes  $\tilde{C}^k$  and  $\hat{C}^s$ . Concretely, for cost volume  $\hat{C}^s$ ,

$$f_k = \begin{cases} \mathcal{I}, & k = s, \\ (s - k) \text{ stride-2 } 3 \times 3 \text{ convs}, & k < s, \\ \text{upsampling } \oplus 1 \times 1 \text{ conv}, & k > s, \end{cases} \quad (6)$$

where  $\mathcal{I}$  denotes the identity function,  $s - k$  stride-2  $3 \times 3$  convolutions are used for  $2^{s-k}$  times downsampling to make the resolution consistent, and  $\oplus$  means bilinear up-sampling to the same resolution first, then following a  $1 \times 1$  convolution to align the number of channels. We denote this architecture as Cross-Scale Aggregation (CSA) module.

Although our CSA module is similar to HRNet[32], they have two major differences. First, we are inspired by traditional cross-scale cost aggregation algorithm [44] and aiming at approximating the geometric conclusion with neural network layers, while HRNet is designed for learning



rich feature representations. Moreover, the channel number (corresponding to the disparity dimension) of lower scale cost volume is *halved* in our approach due to the smaller search range in coarser scales, while HRNet doubles, indicating our architecture is more efficient than HRNet.

### 3.3. Adaptive Aggregation Network

The proposed ISA and CSA modules are complementary and can be integrated, resulting in our final Adaptive Aggregation Module (AAModule, see Fig. 2). We stack six AAModules for cost aggregation, while for the first three AAModules, we simply use regular 2D convolutions for intra-scale aggregation, thus a total of nine deformable convolutions are used for cost aggregation in this paper.

Our feature extractor adopts a ResNet-like [9] architecture (40 layers in total), in which six regular 2D convolutions are replaced with their deformable counterparts. We use Feature Pyramid Network [19] to construct feature pyramid at 1/3, 1/6 and 1/12 resolutions. Two refinement modules proposed in StereoDRNet [3] are used to hierarchically upsample the 1/3 disparity prediction to the original resolution (i.e., upsample to 1/2 resolution first, then to original resolution). Combining all these components leads to our final Adaptive Aggregation Network (AANet).

### 3.4. Disparity Regression

For each pixel, we adopt the *soft argmin* mechanism [14] to obtain the disparity prediction  $\tilde{d}$ :

$$\tilde{d} = \sum_{d=0}^{D_{\max}-1} d \times \sigma(c_d), \quad (7)$$

where  $D_{\max}$  is the maximum disparity range,  $\sigma$  is the soft-max function, and  $c_d$  is the aggregated matching cost for disparity candidate  $d$ .  $\sigma(c_d)$  can be seen as the probability of disparity being  $d$ . This regression based formulation can produce sub-pixel precision and thus is used in this paper.

### 3.5. Loss Function

Our AANet is trained end-to-end with ground truth disparities as supervision. While for KITTI dataset, the high sparsity of disparity ground truth may not be very effective to drive our learning process. Inspired by the knowledge distillation in [10], we propose to leverage the prediction results from a pre-trained stereo model as pseudo ground truth supervision. Specifically, we employ a pre-trained model to predict the disparity maps on the training set, and use the prediction results as pseudo labels in pixels where ground truth disparities are not available. We take the pre-trained GA-Net [43] model as an example to validate the effectiveness of this strategy.

For disparity prediction  $D_{\text{pred}}^i, i = 1, 2, \dots, N$ , it is first bilinearly upsampled to the original resolution. The

corresponding loss function is defined as

$$L_i = \sum_{\mathbf{p}} V(\mathbf{p}) \cdot \mathcal{L}(D_{\text{pred}}^i(\mathbf{p}), D_{\text{gt}}(\mathbf{p})) + (1 - V(\mathbf{p})) \cdot \mathcal{L}(D_{\text{pred}}^i(\mathbf{p}), D_{\text{pseudo}}(\mathbf{p})), \quad (8)$$

where  $V(\mathbf{p})$  is a binary mask to denote whether the ground truth disparity for pixel  $\mathbf{p}$  is available,  $\mathcal{L}$  is the smooth L1 loss [4],  $D_{\text{gt}}$  is the ground truth disparity and  $D_{\text{pseudo}}$  is the pseudo ground truth.

The final loss function is a combination of losses over all disparity predictions

$$L = \sum_{i=1}^N \lambda_i \cdot L_i, \quad (9)$$

where  $\lambda_i$  is a scalar for balancing different terms.

## 4. Experiments

### 4.1. Datasets and Evaluation Metrics

We conduct extensive experiments on three popular stereo datasets: Scene Flow, KITTI 2012 and KITTI 2015. The Scene Flow dataset [20] is a large scale synthetic dataset and provides dense ground truth disparity maps. The end-point error (EPE) and 1-pixel error are reported on this dataset, where EPE is the mean disparity error in pixels and 1-pixel error is the average percentage of pixel whose EPE is bigger than 1 pixel. The KITTI 2012 [7] and KITTI 2015 [22] are real-world datasets in the outdoor scenario, where only sparse ground truth is provided. The official metrics (e.g., D1-all) in the online leader board are reported.

### 4.2. Implementation Details

We implement our approach in PyTorch [27] and using Adam [16] ( $\beta_1 = 0.9, \beta_2 = 0.999$ ) as optimizer. For Scene Flow dataset, we use all training set (35454 stereo pairs) for training and evaluate on the standard test set (4370 stereo pairs). The raw images are randomly cropped to  $288 \times 576$  as input. We train our model on 4 NVIDIA V100 GPUs for 64 epochs with a batch size of 64. The learning rate starts at 0.001 and is decreased by half every 10 epochs after 20th epoch. For KITTI dataset, we use  $336 \times 960$  crop size, and first fine-tune the pre-trained Scene Flow model on mixed KITTI 2012 and 2015 training sets for 1000 epochs. The initial learning rate is 0.001 and decreased by half at 400th, 600th, 800th and 900th epochs. Then another 1000 epochs are trained on the separate KITTI 2012/2015 training set for benchmarking, with an initial learning rate of 0.0001 and same schedule as before. But only the last disparity prediction is supervised with ground truth following a similar strategy in [13]. For all datasets, the input images are normalized with ImageNet mean and standard deviation

Method	Scene Flow		KITTI 2015	
	EPE	> 1px	EPE	D1-all
w/o ISA & CSA	1.10	10.9	0.75	2.63
w/o ISA	0.97	10.1	0.70	<b>2.22</b>
w/o CSA	0.99	10.1	0.69	2.31
AANet	<b>0.87</b>	<b>9.3</b>	<b>0.68</b>	2.29

Table 1: Ablation study of ISA and CSA modules. The best performance is obtained by integrating these two modules.

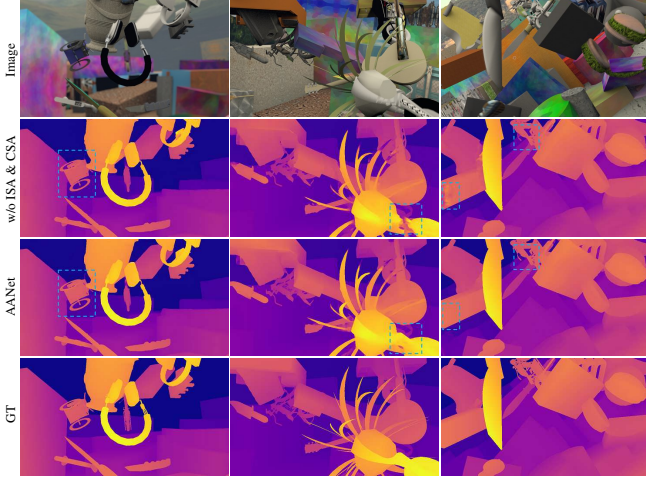


Figure 3: Visual comparisons of ablation study on Scene Flow test set. Our AANet produces sharper results in thin structures and better predictions in textureless regions.

statistics. We use random color augmentation and vertical flipping, and set the maximum disparity as 192 pixels. From highest scale to lowest, the loss weights in Eq. 8 are set to  $\lambda_1 = \lambda_2 = \lambda_3 = 1.0$ ,  $\lambda_4 = 2/3$ ,  $\lambda_5 = 1/3$ .

### 4.3. Analysis

To validate the effectiveness of each component proposed in this paper, we conduct controlled experiments on Scene Flow test set and KITTI 2015 validation set (the KITTI 2015 training set is split into 160 pairs for training and 40 pairs for validation).

**Ablation Study.** As shown in Tab. 1, removing the proposed ISA or CSA module leads to clear performance drop. The best performance is obtained by integrating these two modules, which are designed to be complementary in principle. Fig. 3 further shows the visual comparison results. Our full model produces better disparity predictions in thin structures and textureless regions, demonstrating the effectiveness of the proposed method.

**Sampling Points Visualization.** To better understand our proposed adaptive intra-scale cost aggregation algo-

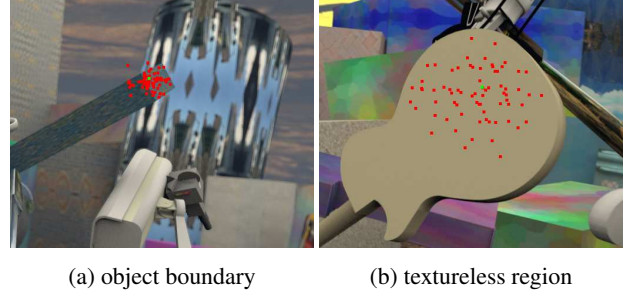


Figure 4: Visualization of sampling points (red points) in two challenging regions (green points). In object boundary (a), the sampling points tend to focus on similar disparity regions. While for large textureless region (b), they are more discretely distributed to sample from a large context.

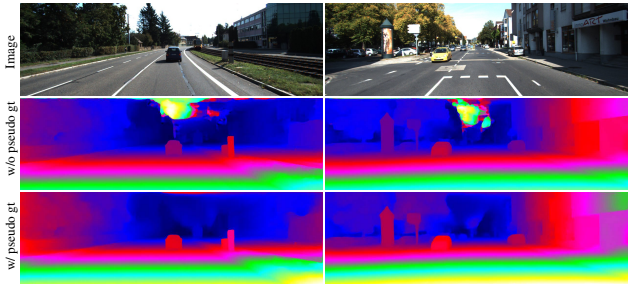


Figure 5: Visualization of disparity prediction results on KITTI 2015 validation set. Leveraging pseudo ground truth as additional supervision helps reduce the artifacts in regions where ground truth disparities are not available, e.g., the sky region.

rithm, we visualize the sampling locations in two challenging regions. As illustrated in Fig. 4, for pixel in object boundary (Fig. 4a), the sampling points tend to focus on similar disparity regions. While for large textureless region (Fig. 4b), a large context is usually required to obtain reliable matching due to lots of local ambiguities. Our method can successfully adapt the sampling locations to these regions, validating that the proposed adaptive aggregation method can not only dynamically adjust the sampling locations, but also enables sampling from a large context.

**Pseudo Ground Truth Supervision.** Fig. 5 shows the visual results on KITTI 2015 validation set. We empirically find that leveraging the prediction results from a pre-trained GA-Net [43] model helps reduce the artifacts in regions where ground truth disparities are not available, e.g., the sky region. Quantitatively, the D1-all error metric decreases from 2.29 to 2.15, while the EPE increases from 0.68 to 0.69. The possible reason might be that the validation set is too small to make the results unstable. Similar phenomenon has also been noticed in [8]. However, the qualitative results indicate that our proposed strategy can be

Method	#3D Convs	#DConvs	#CSA	EPE	> 1px	Params	FLOPs	Memory	Time (ms)
StereoNet [15]	4	0	0	1.10	-	0.62M	106.89G	1.41G	23
StereoNet-AA	0	4	0	<b>1.08</b>	12.9	<b>0.53M</b>	<b>88.17G</b>	<b>1.38G</b>	<b>17</b>
GC-Net [14]	19	0	0	2.51	16.9	2.85M	1754.10G	21.52G	3731
GC-Net-AA	0	9	6	<b>0.98</b>	<b>10.8</b>	<b>2.15M</b>	<b>212.59G</b>	<b>1.97G</b>	<b>91</b>
PSMNet [4]	25	0	0	1.09	12.1	5.22M	613.90G	4.08G	317
PSMNet-AA	0	9	6	<b>0.97</b>	<b>10.2</b>	<b>4.15M</b>	<b>208.73G</b>	<b>1.58G</b>	<b>77</b>
GA-Net [43]	15	0	0	<b>0.84</b>	9.9	4.60M	1439.57G	6.23G	2211
GA-Net-AA	0	14	6	0.87	<b>9.2</b>	<b>3.68M</b>	<b>119.64G</b>	<b>1.63G</b>	<b>57</b>

Table 2: Comparisons with four representative stereo models: StereoNet, GC-Net, PSMNet and GA-Net. We replace the 3D convolutions in cost aggregation stage with our proposed architectures and denote the resulting model with suffix AA. Our method not only obtains clear performance improvements (except GA-Net has lower EPE), but also shows fewer parameters, less computational cost and memory consumption, while being significantly faster than top-performing models ( $41\times$  than GC-Net,  $4\times$  than PSMNet and  $38\times$  than GA-Net). The comparison with StereoNet indicates that our method can also be a valuable way to improve the performance of existing fast stereo models. “DConvs” is short for deformable convolutions.

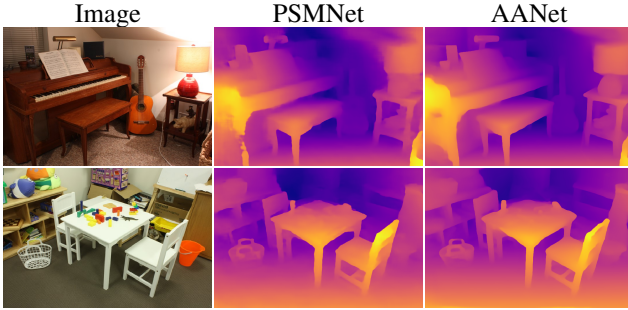


Figure 6: Generalization on Middlebury 2014 dataset. Our AANet produces sharper object boundaries and better preserves the overall structures than PSMNet.

an effective way to handle highly sparse ground truth data.

**Generalization.** We further test the generalization ability of our method on Middlebury 2014 dataset [28]. Specifically, we directly use our KITTI fine-tuned model to predict the disparity map, no additional training is done on Middlebury. Fig. 6 shows the results. Compared with the popular PSMNet [4] model, our AANet produces sharper object boundaries and better preserves the overall structures.

#### 4.4. Comparison with 3D Convolutions

To demonstrate the superiority of our proposed cost aggregation method over commonly used 3D convolutions, we conduct extensive experiments on the large scale Scene Flow dataset.

**Settings.** We mainly compare with four representative stereo models: the first 3D convolution based model GC-Net [14], real-time model StereoNet [15], previous and current state-of-the-art models PSMNet [4] and GA-Net [43].

For fair comparisons, we use similar feature extractors with them. Specifically, StereoNet uses  $8\times$  downsampling for fast speed while we use  $4\times$ ; five regular 2D convolutions in GA-Net are replaced with their deformable counterparts; for GC-Net and PSMNet, the feature extractors are exactly the same. We replace the 3D convolutions in cost aggregation stage with our proposed AAModules, and denote the resulting model with suffix AA. We integrate all these models in a same framework and measure the inference time with  $576 \times 960$  resolution on a single NVIDIA V100 GPU.

**Results.** Tab. 2 shows the comprehensive comparison metrics/statistics. To achieve fast speed, StereoNet [15] uses  $8\times$  downsampling to build a very low-resolution cost volume, while at the cost of sacrificing accuracy. But thanks to our efficient adaptive aggregation architecture, we are able to directly aggregate the  $1/4$  cost volume with even less computation while being more accurate and faster, indicating that our method can be a valuable way to improve the performance of existing fast stereo models. Compared with top-performing stereo models GC-Net [14], PSMNet [4] and GA-Net [43], we not only obtain clear performance improvements (except GA-Net has lower EPE than ours), but also show fewer parameters, less computational cost and memory consumption, while being significantly faster ( $41\times$  than GC-Net,  $4\times$  than PSMNet and  $38\times$  than GA-Net), demonstrating the high efficiency of our method compared with commonly used 3D convolutions.

**Complexity Analysis.** 2D stereo methods use simple feature correlation to build a 3D cost volume ( $D \times H \times W$ ) while 3D methods use concatenation thus a 4D cost volume is built ( $C \times D \times H \times W$ ), where  $C, D, H, W$  denotes channels after feature concatenation, maximum disparity, height and width, respectively.  $C$  usually equals to 64 for



Method	GC-Net [14]	PSMNet [4]	GA-Net [43]	DeepPruner-Best [6]	DispNetC [20]	StereoNet [15]	AANet	AANet*
EPE	2.51	1.09	0.84	0.86	1.68	1.10	0.87	<b>0.83</b>
Time (s)	0.9	0.41	1.5	0.182	0.06	<b>0.015</b>	0.068	0.160

Table 3: Evaluation results on Scene Flow test set. Our method not only achieves state-of-the-art performance but also runs significantly faster than existing top-performing methods.

Method	KITTI 2012		KITTI 2015		Time (s)
	Out-Noc	Out-All	D1-bg	D1-all	
MC-CNN [42]	2.43	3.63	2.89	3.89	67
GC-Net [14]	1.77	2.30	2.21	2.87	0.9
PSMNet [4]	1.49	1.89	1.86	2.32	0.41
DeepPruner-Best [6]	-	-	1.87	2.15	0.182
iResNet-i2 [18]	1.71	2.16	2.25	2.44	<b>0.12</b>
HD <sup>3</sup> [39]	1.40	1.80	1.70	2.02	0.14
GwcNet [8]	<b>1.32</b>	<b>1.70</b>	1.74	2.11	0.32
GA-Net [43]	1.36	1.80	<b>1.55</b>	<b>1.93</b>	1.5
AANet*	1.71	2.21	1.78	2.24	0.142
StereoNet [15]	4.91	6.02	4.30	4.83	<b>0.015</b>
MADNet [33]	-	-	3.75	4.66	0.02
DispNetC [20]	4.11	4.65	4.32	4.34	0.06
DeepPruner-Fast [6]	-	-	2.32	2.59	0.061
AANet	<b>1.91</b>	<b>2.42</b>	<b>1.99</b>	<b>2.55</b>	0.062

Table 4: Benchmark results on KITTI 2012 and KITTI 2015 test sets. Our deeper model AANet\* achieves competitive results among existing top-performing methods while maintaining fast inference speed. Note that HD<sup>3</sup> has more than 6× parameters than ours. Compared with other fast models, our AANet is much more accurate.

3D convolutions based methods and  $D = 64$  for 1/3 resolution cost volume. Supposing the output cost volume has the same size as input and the kernel size of a convolution layer is  $K$  ( $K = 3$  usually), then the computational complexity of a 3D convolution layer is  $\mathcal{O}(K^3 C^2 D H W)$ . In contrast, the complexity of a deformable convolution layer is  $\mathcal{O}(K^2 D^2 H W + 3K^4 D H W + 3K^2 D H W)$ . Therefore, the computational complexity of a deformable convolution layer is less than 1/130 of a 3D convolution layer.

#### 4.5. Benchmark Results

For benchmarking, we build another model variant AANet\* that uses higher-resolution (1/2) cost volume and deeper (61 layers) feature extractor. Tab. 3 shows the evaluation results on Scene Flow test set. Our method not only achieves state-of-the-art results, but also runs significantly faster than existing top-performing methods. The evaluation results on KITTI 2012 and KITTI 2015 benchmarks are shown in Tab. 4. Compared with other fast models, our AANet is much more accurate. The deeper version model AANet\* achieve competitive results while maintaining fast inference speed. We also note that HD<sup>3</sup>[39] has more than

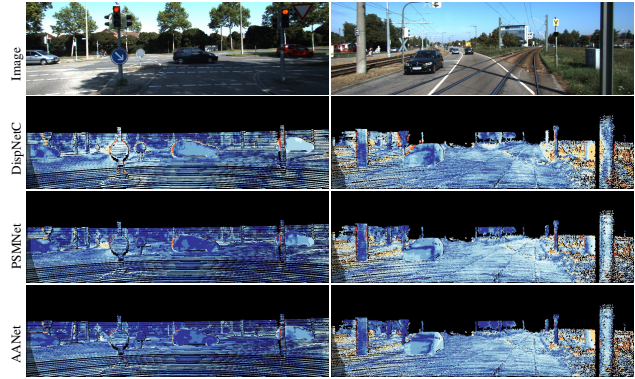


Figure 7: Visualization of disparity prediction error on KITTI 2015 test set (red and yellow denote large errors). Our method produces better results in object boundaries. Best viewed enlarged.

6× parameters than our AANet\* (39.1M vs. 5.9M), and our AANet\* performs much better than iResNet-i2[18] on the more challenging KITTI 2015 dataset, demonstrating that our method achieves a better balance between accuracy and speed. Fig. 7 further visualizes the disparity prediction error on KITTI 2015 test set. Our method produces better results in object boundaries, validating the effectiveness of our proposed adaptive aggregation algorithm.

## 5. Conclusion

We have presented an efficient architecture for cost aggregation, and demonstrated its superiority over commonly used 3D convolutions by high efficiency and competitive performance on both Scene Flow and KITTI datasets. Extensive experiments also validate the generic applicability of the proposed method. An interesting future direction would be extending our method to other cost volume based tasks, e.g., high-resolution stereo matching [37], multi-view stereo [38] and optical flow estimation [30]. We also hope our lightweight design can be beneficial for downstream tasks, e.g., stereo based 3D object detection [34].

**Acknowledgements.** We thank anonymous reviewers for their constructive comments. This work was supported by the National Natural Science Foundation of China (No. 61672481) and Youth Innovation Promotion Association CAS (No. 2018495).



## References

- [1] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. In *ACM Transactions on Graphics (ToG)*, volume 28, page 24. ACM, 2009.
- [2] Yu Boykov, Olga Veksler, and Ramin Zabih. A variable window approach to early vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1283–1294, 1998.
- [3] Rohan Chabra, Julian Straub, Christopher Sweeney, Richard Newcombe, and Henry Fuchs. Stereodrnnet: Dilated residual stereonet. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11786–11795, 2019.
- [4] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5410–5418, 2018.
- [5] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017.
- [6] Shivam Duggal, Shenlong Wang, Wei-Chiu Ma, Rui Hu, and Raquel Urtasun. Deepprunner: Learning efficient stereo matching via differentiable patchmatch. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4384–4393, 2019.
- [7] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012.
- [8] Xiaoyang Guo, Kai Yang, Wukui Yang, Xiaogang Wang, and Hongsheng Li. Group-wise correlation stereo network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3273–3282, 2019.
- [9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [10] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.
- [11] Heiko Hirschmüller, Peter R Innocent, and Jon Garibaldi. Real-time correlation-based stereo vision with reduced border errors. *International Journal of Computer Vision*, 47(1-3):229–246, 2002.
- [12] Asmaa Hosni, Christoph Rhemann, Michael Bleyer, Carsten Rother, and Margrit Gelautz. Fast cost-volume filtering for visual correspondence and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(2):504–511, 2012.
- [13] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. FlowNet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470, 2017.
- [14] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 66–75, 2017.
- [15] Sameh Khamis, Sean Fanello, Christoph Rhemann, Adarsh Kowdle, Julien Valentin, and Shahram Izadi. Stereonet: Guided hierarchical refinement for real-time edge-aware depth prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 573–590, 2018.
- [16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [17] Vladimir Kolmogorov and Ramin Zabih. Computing visual correspondence with occlusions using graph cuts. In *Eighth IEEE International Conference on Computer Vision*, volume 2, pages 508–515, 2001.
- [18] Zhengfa Liang, Yiliu Feng, Yulan Guo, Hengzhu Liu, Wei Chen, Linbo Qiao, Li Zhou, and Jianfeng Zhang. Learning for disparity estimation through feature constancy. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2811–2820, 2018.
- [19] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.
- [20] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas 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*, pages 4040–4048, 2016.
- [21] Michael D Menz and Ralph D Freeman. Stereoscopic depth processing in the visual cortex: a coarse-to-fine mechanism. *Nature neuroscience*, 6(1):59–65, 2003.
- [22] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3061–3070, 2015.
- [23] Dongbo Min, Jiangbo Lu, and Minh N Do. A revisit to cost aggregation in stereo matching: How far can we reduce its computational redundancy? In *2011 International Conference on Computer Vision*, pages 1567–1574. IEEE, 2011.
- [24] Dongbo Min and Kwanghoon Sohn. Cost aggregation and occlusion handling with wls in stereo matching. *IEEE Transactions on Image Processing*, 17(8):1431–1442, 2008.
- [25] Guang-Yu Nie, Ming-Ming Cheng, Yun Liu, Zhengfa Liang, Deng-Ping Fan, Yue Liu, and Yongtian Wang. Multi-level context ultra-aggregation for stereo matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3283–3291, 2019.
- [26] Masatoshi Okutomi and Takeo Kanade. A locally adaptive window for signal matching. *International Journal of Computer Vision*, 7(2):143–162, 1992.
- [27] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming

- Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019.
- [28] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *German conference on pattern recognition*, pages 31–42. Springer, 2014.
- [29] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1-3):7–42, 2002.
- [30] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8934–8943, 2018.
- [31] Jian Sun, Nan-Ning Zheng, and Heung-Yeung Shum. Stereo matching using belief propagation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (7):787–800, 2003.
- [32] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5693–5703, 2019.
- [33] Alessio Tonioni, Fabio Tosi, Matteo Poggi, Stefano Mattoccia, and Luigi Di Stefano. Real-time self-adaptive deep stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 195–204, 2019.
- [34] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8445–8453, 2019.
- [35] Zhenyao Wu, Xinyi Wu, Xiaoping Zhang, Song Wang, and Lili Ju. Semantic stereo matching with pyramid cost volumes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7484–7493, 2019.
- [36] Qingshan Xu and Wenbing Tao. Multi-scale geometric consistency guided multi-view stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5483–5492, 2019.
- [37] Gengshan Yang, Joshua Manela, Michael Happold, and Deva Ramanan. Hierarchical deep stereo matching on high-resolution images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5515–5524, 2019.
- [38] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 767–783, 2018.
- [39] Zhichao Yin, Trevor Darrell, and Fisher Yu. Hierarchical discrete distribution decomposition for match density estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6044–6053, 2019.
- [40] Kuk-Jin Yoon and In So Kweon. Adaptive support-weight approach for correspondence search. *IEEE transactions on pattern analysis & machine intelligence*, (4):650–656, 2006.
- [41] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 472–480, 2017.
- [42] Jure Zbontar and Yann LeCun. Computing the stereo matching cost with a convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1592–1599, 2015.
- [43] Feihu Zhang, Victor Prisacariu, Ruigang Yang, and Philip HS Torr. Ga-net: Guided aggregation net for end-to-end stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 185–194, 2019.
- [44] Kang Zhang, Yuqiang Fang, Dongbo Min, Lifeng Sun, Shiqiang Yang, Shuicheng Yan, and Qi Tian. Cross-scale cost aggregation for stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1590–1597, 2014.
- [45] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9308–9316, 2019.