Accurate and Efficient Stereo Matching via Attention Concatenation Volume

Gangwei Xu, Yun Wang, Junda Cheng, Jinhui Tang, *Member, IEEE*, Xin Yang, *Member, IEEE*

Abstract—Stereo matching is a fundamental building block for many vision and robotics applications. An informative and concise cost volume representation is vital for stereo matching of high accuracy and efficiency. In this paper, we present a novel cost volume construction method, named attention concatenation volume (ACV), which generates attention weights from correlation clues to suppress redundant information and enhance matching-related information in the concatenation volume. The ACV can be seamlessly embedded into most stereo matching networks, the resulting networks can use a more lightweight aggregation network and meanwhile achieve higher accuracy. We further design a fast version of ACV to enable real-time performance, named Fast-ACV, which generates high likelihood disparity hypotheses and the corresponding attention weights from low-resolution correlation clues to significantly reduce computational and memory cost and meanwhile maintain a satisfactory accuracy. The core idea of our Fast-ACV is volume attention propagation (VAP) which can automatically select accurate correlation values from an upsampled correlation volume and propagate these accurate values to the surroundings pixels with ambiguous correlation clues. Furthermore, we design a highly accurate network ACVNet and a real-time network Fast-ACVNet based on our ACV and Fast-ACV respectively, which achieve the state-of-the-art performance on several benchmarks (i.e., our ACVNet ranks the 2nd on KITTI 2015 and Scene Flow, and the 3rd on KITTI 2012 and ETH3D among all the published methods; our Fast-ACVNet outperforms almost all state-of-the-art real-time methods on Scene Flow, KITTI 2012 and 2015 and meanwhile has better generalization ability). The source code is available at https://github.com/gangweiX/ACVNet and https://github.com/gangweiX/Fast-ACVNet.

Index Terms—Stereo Matching, Cost Volume Construction, Attention Concatenation Volume, Attention Filtering.

✦

1 INTRODUCTION

STEREO matching, which estimates depth (or disparity) from a pair of rectified stereo images [1], [2], is a fundamental task for many robotics and computational photography applications, such as 3D reconstruction, robot navigation and autonomous driving. Despite a plethora of research works in the literature, stereo matching remains challenging due to difficulties in tackling repetitive structures, texture-less/transparent objects and occlusions. Meanwhile, how to concurrently achieve a high inference accuracy and efficiency is critical for practical applications yet remains challenging.

Recently, convolutional neural networks have exhibited great potentials in this field. State-of-the-art CNN stereo models typically consist of four steps, i.e. feature extraction, cost volume construction, cost aggregation and disparity regression. The cost volume which provides initial similarity measures for left image pixels and possible corresponding right image pixels is a crucial step of stereo matching. An informative and concise cost volume representation from this step is vital for the final accuracy and computational complexity. Learning-based methods explore different cost volume representations. DispNetC [3] computes a single-channel full correlation volume between the left and right feature maps along every disparity level. Such full correlation volume provides an efficient way for measuring simi-

larities, but it loses much content information. GC-Net [4] constructs a 4D concatenation volume by concatenating left and right feature maps along all disparity levels to provide abundant content information. However, the concatenation volume does not provide explicit similarity measurements, and thus requires extensive 3D convolutions for cost aggregation to learn similarity measurements from scratch. To tackle the above drawbacks, GwcNet [5] concatenates the group-wise correlation volume with a compact concatenation volume to encode both matching and content information in the final 4D cost volume. However, the data distribution and characteristics of a correlation volume and a concatenation volume are quite different, i.e. the former represents the similarity measurement obtained through dot product, and the latter is the concatenation of the unary features. Simply concatenating the two volumes and regularizing them via 3D convolutions can hardly exert the advantages of the two volumes to the full. As a result, GwcNet [5] still requires extensive 3D convolutions for cost aggregation. To further reduce the memory and computational complexity, several methods [6], [7], [8] employ the cascade cost volume which build a cost volume pyramid in a coarse-to-fine manner to progressively narrow down the target disparity range. However, these cascaded methods need to re-construct and re-aggregate a cost volume for each stage without reusing prior information in the probability volume of the previous stage, yielding a low utilization efficiency. In addition, these cascaded methods could suffer from irreversible cumulative errors as they directly discard disparities that are beyond the prediction range in the previous stages.

To balance efficiency and accuracy, several recent studies

• *G. Xu, Y. Wang, J. Cheng and X. Yang are with the Department of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail:{gwxu, wangyun, cjd, xinyang2014}@hust.edu.cn)*
• *J. Tang is with Nanjing University of Science and Technology (e-mail: jinhuitang@njust.edu.cn).*

Corresponding author: Xin Yang.

attempt to construct and aggregate only a low resolution (i.e., 1/8) 4D cost volume and restore a full resolution disparity map via up-sampling. For instance, StereoNet [9] constructs a 4D cost volume based on the differences between the left feature and right feature maps at only 1/8 resolution. Then 2D disparity maps are regressed and then up-sampled via bilinear interpolation form the 1/8 resolution cost volume. Similarly, BGNet [10] constructs a low-resolution 4D group-wise correlation volume and designs a parameter-free slicing layer based on a bilateral grid to obtain an edge-preserving high-resolution cost volume from the low-resolution cost volume. DeepPruner [11] develops a differentiable PatchMatch [12] module to efficiently construct a sparse representation of a low-resolution concatenation volume. The search space of each pixel is pruned by the predicted minimum and maximum disparities. Unfortunately, these efficiency-oriented methods typically degrade the accuracy greatly compared with the best-performing algorithms.

This work aims to explore a more efficient and effective form of cost volume, which can achieve the state-of-the-art accuracy with a high efficiency. We build our model based on two key observations: First, the correlation volume which measures feature similarities between left and right images can quickly provide rough geometric structure information. Second, the concatenation volume contains rich content and fine structure but redundant information. This suggests that utilizing the correlation volume which encodes geometric structure information can facilitate a concatenation volume to significantly suppress its redundant information and meanwhile maintain sufficient information for estimating the correct disparity.

With these intuitions in mind, we propose an Attention Concatenation Volume (ACV) which exploits a lightweight correlation volume to generate attention weights to filter a concatenation volume (see Fig. 1). The ACV can achieve a high accuracy and meanwhile significantly alleviate the burden of cost aggregation. Experimental results show that after replacing the combined volume of GwcNet with our ACV, only four 3D convolutions for cost aggregation can achieve better accuracy than GwcNet which employs twenty-eight 3D convolutions for cost aggregation. Our ACV is a general cost volume representation that can be seamlessly integrated into various 3D CNN stereo models for performance improvement. Results show that after applying our method, PSMNet [13] and GwcNet [5] can respectively achieve additional a 28% and 39% accuracy improvement.

We further design a faster version of ACV to enable real-time performance (i.e., inference time of stereo matching is less than 50ms), called Fast-ACV (see Fig. 2). The key of our Fast-ACV is a novel Volume Attention Propagation (VAP) module which can restore a high-quality correlation volume of high resolution from the low-resolution volume to significantly reduce the time cost of correlation volume construction and aggregation. Specifically, our VAP automatically identifies a set of pixels with reliable correlation values in an interpolated correlation volume of high resolution and propagates such information to their neighbors to progressively revise errors and reduce ambiguities in the high-resolution correlation volume. Pixels which have accurate regressed disparities and sharply-distributed disparity

probabilities (i.e. high confidence of regressed disparity) are considered to be reliable and can be used to guide the revision of correlation values for the neighboring pixels. To this end, we design an overall measure to evaluate the disparity estimation accuracy and confidence for each pixel in the interpolated correlation volume. In addition, we adopt a cross shape sampling pattern to enable a highly effective and efficient propagation path. Based on our VAP, we propose to utilize a "Fine-to-Important" sampling strategy which generates a set of disparity hypotheses with high likelihood and the corresponding attention weights to significantly suppress impossible disparities in the concatenation volume and in turn reduce time and memory cost.

Based on the advantages of the proposed ACV and Fast-ACV, we design an accurate stereo matching network ACVNet and its real-time version Fast-ACVNet. At the time of writing, our ACVNet ranks the 2nd on KITTI 2015 [14] and Scene Flow [3], and the 3rd on KITTI 2012 [15] and ETH3D [16] among all the published methods. It is noteworthy that our ACVNet is the only method that ranks top 3 concurrently on all four datasets above, demonstrating its good generalization ability to various scenes. Regarding the inference speed, our ACVNet is the fastest among the top 10 methods in the KITTI benchmarks. Meanwhile, our Fast-ACVNet outperforms almost all state-of-the-art real-time methods on Scene Flow [3], KITTI 2012 [15] and 2015 [14].

In summary, our main contributions are:

- We propose an Attention Concatenation Volume (ACV) and its real-time version Fast-ACV, which adopt a unified framework to construct highly informative and concise cost volume representation and can be seamlessly integrated into most existing stereo models to significantly reduce computational cost and improve accuracy.
- We propose a Volume Attention Propagation (VAP) module which is a key success factor of Fast-ACV. Our VAP can restore a high-quality correlation volume of high resolution from the low-resolution volume to enable great acceleration and meanwhile maintain satisfactory accuracy.
- We propose a highly accurate stereo matching model, ACVNet, and a real-time stereo matching model, Fast-ACVNet, which achieve state-of-the-art performance on several popular benchmarks.
- We release the source codes of ACVNet and Fast-ACVNet.

2 RELATED WORK

2.1 Cost Volume based Deep Stereo Matching

Recently, CNN-based stereo models [5], [13], [17], [18], [19], [20], [21] have achieved impressive performance on almost all the standard benchmarks. Most of them devote to improving the accuracy and efficiency of cost volume construction and cost aggregation, which are the two key steps of stereo matching.

Cost Volume Construction. Existing cost volume representation can be roughly categorized into three types: correlation volume, concatenation volume and combined volume by concatenating the two volumes. DispNetC [3] utilizes
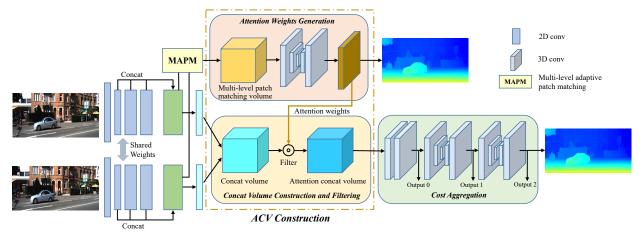
Fig. 1: The structure of our proposed ACVNet. The construction process of ACV consists of three steps: Attention Weights Generation, Initial Concatenation Volume Construction and Attention Filtering.

a correlation layer to directly measure the similarities of left and right image features to form a single-channel cost volume for each disparity level. Then, 2D convolutions are applied to aggregate contextual information. Such full correlation volume demands low memory and computational complexity, yet the encoded information is too limited (i.e. too much content information is lost in the channel dimension) to achieve a satisfactory accuracy. GC-Net [4] uses the concatenation volume, which concatenates the left and right CNN features to form a 4D cost volume for all disparities. Such 4D concatenation volume preserves abundant content information from all feature channels and thus outperforms the correlation volume in terms of accuracy. However, as the concatenation volume does not explicitly encodes similarity measures, it requires a deep stack of 3D convolutions to aggregate costs of all disparities from scratch. To overcome the above drawbacks, GwcNet [5] proposes the group-wise correlation volume and concatenates it with a compact concatenation volume to form a combined volume, which aims to combine the advantages of two volumes. However, directly concatenating two types of volumes without considering their respective characteristics yields an inefficient use of the complementary strengths in the two volumes. As a result, deep stacking 3D convolutions in the hourglass architecture are still demanded for cost aggregation in GwcNet [5]. Following the 4D combined cost volume, cascaded approaches [6], [7] build a 4D cost volume pyramid in a coarse-to-fine manner to progressively narrow down the target disparity range and refine the disparity map. However, these cascaded approaches need to reconstruct and aggregate the cost volume for every stage which also incur abundant 3D convolutions. In addition, they usually do not use the prior information in the probability volume of the previous stage, yielding a low utilization efficiency. Moreover, such coarse-to-fine strategy inevitably involves irreversible accumulated errors, i.e., refining only the peak disparity regressed from the previous stage may miss the true disparity when the disparity distribution contains multiple peaks.

Cost Aggregation. The goal of this step is to aggregate contextual information in the initial cost volume to derive accurate similarity measures. Many existing methods [5],

[13] exploit a deep 3D CNN to learn an effective similarity function from the cost volume. However, the computational and memory consumption is too high for time-constrained applications. To reduce the complexity, AANet [22] proposes an intra-scale and cross-scale cost aggregation algorithm to replace the conventional 3D convolutions which can achieve very fast inference speed with a sacrifice of nontrivial accuracy degradation. GANet [23] also tries to replace 3D convolutions with two guided aggregation layers, which achieves a higher accuracy using spatially dependent 3D aggregation at the cost of a higher aggregation time due to the two guided aggregation layers.

Cost volume construction and aggregation are two tightly-coupled modules which jointly determine the accuracy and efficiency of a stereo matching network. In this work, we propose a highly efficient yet informative cost volume representation, named attention concatenation volume, by using the similarity information encoded in the correlation volume to regularize the concatenation volume so that only a lightweight aggregation network is demanded to achieve an overall high efficiency and accuracy.

2.2 Real-time Stereo Matching

Several recent studies [9], [10], [11], [24] focus on lightweight stereo networks based on 4D cost volumes to achieve real-time performance and meanwhile maintain satisfactory accuracy. These methods typically construct and aggregate a 4D cost volume at low resolution to significantly reduce computational cost. To compensate information loss at low resolution, StereoNet [9] proposes an edge-preserving refinement network, which utilizes the left images as a guidance to recover high frequency details. DeepPruner [11] adopts the idea of PatchMatch [12] to first build a sparse representation of the cost volume, and then prune the search space based on the predicted minimum and maximum disparities. The predicted disparities are further refined under the guidance of low-level image feature maps. BGNet [10] proposes an up-sampling module based on the learned bilateral grid to restore a 4D cost volume of high resolution from a low-resolution cost volume. HITNet [25] represents image tiles as planar patches and integrates image warping, spatial propagation and a fast high resolution initialization
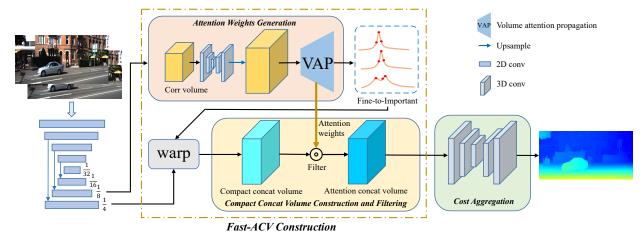
Fig. 2: The structure of our proposed Fast-ACVNet. We first exploit a correlation volume to generate disparity hypotheses with high likelihood and the corresponding attention weights. Then we use the attention weights to filter the compact concatenation volume constructed based on disparity hypotheses, deriving our Fast-ACV.

step into the network, to reduce computational cost. However, HITNet [25] requires extra propagation loss, slant loss and confidence loss for training, which could lead to a poor generalization ability in unseen scenes with different characteristics from the training data. In comparison, our Fast-ACV has better generalization ability and can be applied to many 4D cost volume based stereo models as plugins, which are more convenient.

Motivated by existing real-time methods, our real-time version of ACV (i.e., Fast-ACV) employs a novel propagation strategy (via the VAP module) to efficiently obtain interpolated correlation values which can better tolerate noises in the original low-resolution correlation volume and meanwhile effectively recover thin structures and sharp boundaries. Meanwhile, different from existing real-time methods which directly regress disparities from the interpolated correlation volume, we use the interpolated correlation values to generate disparity hypotheses with high likelihood and the corresponding attention weights to regularize the concatenation volume. Also, our Fast-ACV is different from the cascaded approaches which narrow down the disparity search space via a coarse-to-fine strategy, our method preserves all disparity hypotheses with high likelihood and adjusts the attention weights of hypotheses (i.e., a fine-to-important strategy) to avoid irreversible cumulative errors.

3 METHOD

In this section, we first describe the basic design of our ACV (Sec. 3.1) and Fast-ACV (Sec. 3.2). Then in Secs. 3.3 and 3.4 we present details of network architectures of ACVNet and Fast-ACVNet respectively. Finally, in Sec. 3.5, we explain the loss functions used to train our ACVNet and Fast-ACVNet.

3.1 Attention Concatenation Volume

The construction process of attention concatenation volume consists of three steps: attention weights generation, initial concatenation volume construction and attention filtering.
1) Attention Weights Generation. The attention weights aim to filter the initial concatenation volume so as to emphasize useful information and suppress irrelevant information.

To this end, we generate attention weights by extracting geometric information from correlations between a pair of stereo images. Conventional correlation volume is obtained by computing pixel-to-pixel similarity which becomes unreliable for textureless regions due to lack of sufficient matching clues. To address this problem, we propose a more robust correlation volume construction method via multi-level adaptive patch matching (MAPM). We obtain feature maps at three different levels l_1, l_2 and l_3 from the feature extraction module, and the number of channels for l_1, l_2 and l_3 is 64, 128 and 128 respectively. For each pixel at a particular level, we utilize an atrous patch with a predefined size and adaptively learned weights to calculate the matching cost. By controlling the dilation rate, we ensure that the patch's scope is related to the feature map level and meanwhile maintains the same number of pixels in similarity calculation for the center pixel. The similarity of two corresponding pixels is then a weighted sum of correlations between corresponding pixels within in the patch.

We split features into groups and compute correlation maps group by group [5]. Three levels feature maps of l_1, l_2 and l_3 are concatenated to form N_f-channel unary feature maps (N_f=320). We equally divide N_f channels into N_g groups (N_g=40), and accordingly the first 8 groups are from l_1, the middle 16 groups are from l_2, and the last 16 groups are from l_3. Feature maps of different levels will not interfere with each other. We denote the gth feature group as $\mathbf{f}_l^g$, $\mathbf{f}_r^g$, and multi-level patch matching volume $\mathbf{C}_{patch}$ is computed as,

$$\mathbf{C}_{patch}^{l_k}(g,d,x,y) = \frac{1}{N_f/N_g} \sum_{(i,j)\in\Omega^k} \omega_{ij}^{k,g} \cdot C_{ij}^g(d,x,y)$$
$$C_{ij}^g(d,x,y) = \langle \mathbf{f}_l^g(x-i,y-j), \mathbf{f}_r^g(x-i-d,y-j)\rangle,$$

(1)

where $\mathbf{C}_{patch}^{l_k}$ ($k\in(1,2,3)$) represents the matching cost of the feature level k, $\langle\cdot,\cdot\rangle$ is the inner product, (x,y) represents the pixel's location, and d denotes a disparity level. $\Omega^k=(i,j)$ ($i,j\in(-k,0,k)$) is a nine-point coordinate set, defining the scope of the patch on the k-level feature maps ($k\in(1,2,3)$). ω_{ij}^k represents the weight of a pixel (i,j) in the patch on the k-level feature maps and is learned adaptively during the training process. The final multi-level

patch matching volume is then obtained by concatenating matching costs $\mathbf{C}_{patch}^{l_k}$ ($k \in (1, 2, 3)$) of all levels,

$$\mathbf{C}_{patch} = \text{Concat} \left\{ \mathbf{C}_{patch}^{l_1}, \mathbf{C}_{patch}^{l_2}, \mathbf{C}_{patch}^{l_3} \right\}, \qquad (2)$$

we denote the derived multi-level patch matching volume as $\mathbf{C}_{patch} \in \mathbb{R}^{N_g \times D/4 \times H/4 \times W/4}$. We then apply two 3D convolutions and a 3D hourglass network [5] to regularize $\mathbf{C}_{patch}$, and then use another convolution layer to compress the channels to 1 and derive the attention weights, i.e. $\mathbf{A} \in \mathbb{R}^{1 \times D/4 \times H/4 \times W/4}$.

To obtain accurate attention weights of different disparities to filter the initial concatenation volume, we use the ground truth disparity to supervise $\mathbf{A}$. Specifically, we use the $soft\ argmin$ function [4] to obtain the disparity estimation $\mathbf{d}_{att}$ from $\mathbf{A}$. We compute smooth L1 loss between $\mathbf{d}_{att}$ and the disparity ground truth to guide network learning.

2) Initial Concatenation Volume Construction. Given an input stereo image pair whose size is $H \times W \times 3$, for each image, we obtain unary feature maps $\mathbf{f}_l$ and $\mathbf{f}_r$ for the left and right images respectively from CNN feature extraction. The size of feature maps of $\mathbf{f}_l$ ($\mathbf{f}_r$) is $N_c \times H/4 \times W/4$ ($N_c = 32$). The initial concatenation volume is then formed by concatenating the $\mathbf{f}_l$ and $\mathbf{f}_r$ for each disparity level as,

$$\mathbf{C}_{concat}(\cdot, d, x, y) = \text{Concat} \left\{ \mathbf{f}_l(x, y), \mathbf{f}_r(x - d, y) \right\}, \quad (3)$$

the accordingly size of $\mathbf{C}_{concat}$ is $2N_c \times D/4 \times H/4 \times W/4$, D denotes the maximum of disparity.

3) Attention Filtering. After obtaining the attention weights $\mathbf{A}$, we use it to eliminate redundant information in the initial concatenation volume and in turn enhance its representation ability. The attention concatenation volume $\mathbf{C}_{ACV}$ at channel i is computed as,

$$\mathbf{C}_{ACV}(i) = \mathbf{A} \odot \mathbf{C}_{concat}(i), \qquad (4)$$

where $\odot$ represents element-wise product, and the attention weights $\mathbf{A}$ are applied to all channels of the initial concatenation volume.

3.2 Fast Attention Concatenation Volume

Compared with ACV, our Fast-ACV accelerate the process from two aspects. First, Fast-ACV employs a novel volume attention propagation (VAP) module which enables it to perform the majority of calculations for generating disparity hypotheses with high likelihood and their attention weights at a low resolution. Second, our Fast-ACV constructs and filters a compact concatenation volume based on only the high likelihood disparity hypotheses to further reduce the time and memory cost. In the following, we present details of our VAP, compact concatenation volume construction and filtering.

3.2.1 Volume Attention Propagation

As shown in Fig. 2, we construct and aggregate a 4D correlation volume at a low resolution and then form an initial high-resolution correlation volume via bilinear interpolation. In order to obtain a set of seed pixels which are likely to have accurate matches (i.e. a high accuracy for disparity prediction) and meanwhile have sharply-distributed cost values along all the disparity levels (i.e. high confidence
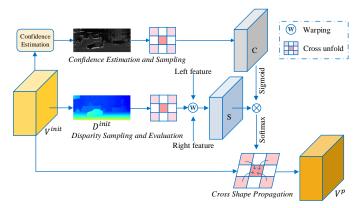

Fig. 3: Volume Attention Propagation.

for disparity prediction), we conduct joint screening of neighboring disparity from two complementary perspectives: 1) **Disparity Sampling and Evaluation** to obtain matching scores, which represent the correlation degree of the surrounding disparities; 2) **Confidence Estimation and Sampling** to obtain confidence scores, which represent the degree of reliability of the surrounding disparities. Finally, the two evaluation scores are integrated through the **Cross Shape Propagation** which follows a principle that only the disparity with high correlation and high reliability simultaneously is the first choice for propagating to the current location. We detail each of the three components below.

1) Disparity Sampling and Evaluation. Given an initial high-resolution correlation volume $\mathbf{V}^{init} \in \mathbb{R}^{D/4 \times H/4 \times W/4}$ (as shown in Fig. 3) up-sampled from a low resolution correlation volume via bilinear interpolation, we first regress an initial disparity map $\mathbf{D}^{init} \in \mathbb{R}^{H/4 \times W/4}$ from $\mathbf{V}^{init}$, and then sample disparities of adjacent pixels through convolution with a pre-defined one-hot filter pattern. To fully exploit neighboring information, the scope of sampling is related to the up-sampling factor. For example, when the up-sampling factor is 2, we set the size of the sampling block to 3×3. Meanwhile, for efficiency, we sample disparities around each pixel in a cross shape instead of a square shape. As shown in Fig. 3, the candidate disparities at pixel i after sampling is $\mathbf{D}_m^{init}(i)$, $m = 1, 2, 3, 4, 5$. Matching score at pixel i is computed as,

$$\mathbf{S}_m(i) = \left\langle \mathbf{F}_l(i), \mathbf{F}_r \left(i - \mathbf{D}_m^{init}(i) \right) \right\rangle, m = 1, 2, 3, 4, 5, \quad (5)$$

where $\langle \cdot, \cdot \rangle$ is the inner product, $\mathbf{F}_l, \mathbf{F}_r \in \mathbb{R}^{C \times H/4 \times W/4}$ are left and right feature maps at $1/4$ resolution from feature extraction and provide details for calculating matching scores for VAP to alleviate the blurred edges problem.

2) Confidence Estimation and Sampling. With the high-resolution correlation volume $\mathbf{V}^{init}$, we obtain a probability volume $\mathbf{P}^{init} \in \mathbb{R}^{D/4 \times H/4 \times W/4}$ by $softmax$. Due to the disparity discontinuity in the edge regions, the problem of multi-peak distribution is easily caused by linear interpolation in the edge regions, as shown in Fig. 4, which causes low confidence and high prediction error. Thus, pixels with multi-peak distribution are unreliable and should not be used for propagation. To this end, we propose to employ uncertainty estimation to evaluate the pixel-level confidence of the current estimation. The uncertainty is estimated by the variance of the distribution. The uncertainty $\mathbf{U}(i)$ at

Fig. 4: Visualization of confidence map. For edge regions, the disparity probability distribution is multi-peak due to interpolations from different objects, so the variance-based uncertainty is larger, which represents low confidence. The bright regions represent low confidence.

pixel i is calculated as:

$$\mathbf{U}(i) = \sum_{d=0}^{D/4-1} \mathbf{P}_d^{init}(i) \times \left(d - \mathbf{D}^{init}(i)\right)^2, \qquad (6)$$

where $\mathbf{P}_d^{init}(i)$ means the dth probability value at pixel i. With uncertainty $\mathbf{U}(i)$, the confidence at pixel i is defined as,

$$\mathbf{C}(i) = \alpha + \beta \times \mathbf{U}(i), \qquad (7)$$

where α, β are learned parameters. The greater the uncertainty is, the lower the confidence will be. For the pixel i, we sample confidence around it in a cross shape to obtain the candidate confidence scores which are $\mathbf{C}_m(i)$, $m = 1, 2, 3, 4, 5$, as shown in Fig. 3.

3) Cross Shape Propagation. Combining matching score and confidence score of each neighboring pixel, we can obtain an overall propagation weight as,

$$\mathbf{W}_m(i) = \mathbf{S}_m(i) \times sigmoid\left(\mathbf{C}_m(i)\right). \qquad (8)$$

Along the spatial dimension, we unfold the initial correlation volume $\mathbf{V}^{init}$ in the cross shape, and then obtain the unfolded correlation volume $\mathbf{V}^u \in \mathbb{R}^{M \times D/4 \times H/4 \times W/4}$. At spatial position i and disparity d of $\mathbf{V}^u$, we can obtain the vector $\mathbf{V}^u(i, d) \in \mathbb{R}^M$. The cross shape propagation operation is defined as,

$$\mathbf{V}^p(i, d) = \sum_{m=1}^{M} \mathbf{V}_m^u(i, d) \times softmax(\mathbf{W}_m(i)), \qquad (9)$$

$\mathbf{V}^p \in \mathbb{R}^{D/4 \times H/4 \times W/4}$ is the correlation volume after propagation (see Fig. 3).

3.2.2 Compact Concatenation Volume Construction and Filtering.

Fast-ACV only preserves high likelihood hypotheses for constructing a compact concatenation volume and their corresponding attention weights for filtering the compact concatenation volume, which can greatly reduce computational cost. With the correlation volume $\mathbf{V}^p$, we obtain the disparity probability distribution volume $\mathbf{P} \in \mathbb{R}^{D/4 \times H/4 \times W/4}$ by applying the $softmax$ function to it. Ideally, the disparity probability distribution $\mathbf{P}$ should be unimodal peaked at true disparities. However, the actual probability distribution

could be multimodal at some pixels, such as ill-posed areas, texture-less regions, and occlusions (see Fig. 5). To concurrently preserves high likelihood hypotheses and reduce computational cost, we use the Top-K probability values of $\mathbf{P}$ at every pixel as attention weights $\mathbf{A}^F \in \mathbb{R}^{K \times H/4 \times W/4}$, and use their corresponding disparities as high likelihood hypotheses $\mathbf{D}^{hyp} \in \mathbb{R}^{K \times H/4 \times W/4}$,

$$\mathbf{A}^F = max_{i=1}^K \{\mathbf{P}\}, \qquad (10)$$

$$\mathbf{D}^{hyp} = arg\ max_{i=1}^K \{\mathbf{P}\}, \qquad (11)$$

The compact concatenation volume is then constructed by concatenating the $\mathbf{f}_l^F$ and $\mathbf{f}_r^F$ based on only the high likelihood disparity hypotheses as,

$$\mathbf{C}_{concat}^{compact}(\cdot, \mathbf{D}_{xy}^{hyp}, x, y) = Concat\left\{\mathbf{f}_l^F(x, y), \mathbf{f}_r^F(x - \mathbf{D}_{xy}^{hyp}, y)\right\} \qquad (12)$$

The unary feature maps $\mathbf{f}_l^F$ and $\mathbf{f}_r^F$ for the left and right images respectively is from CNN feature extraction. Finally, we use attention weights $\mathbf{A}^F$ to filter the compact concatenation volume to enhance its representation ability. The fast attention concatenation volume $\mathbf{C}_{F-ACV}$ at channel i is computed as,

$$\mathbf{C}_{F-ACV}(i) = \mathbf{A}^F \odot \mathbf{C}_{concat}^{compact}(i). \qquad (13)$$

3.3 ACVNet Architecture

Based on the ACV, we design an accurate and efficient end-to-end stereo matching network, named ACVNet. Fig. 1 shows the architecture of our ACVNet which consists of four steps of Feature Extraction, Attention Concatenation Volume Construction, Cost Aggregation and Disparity Prediction. In the following, we introduce each step in details.

Feature Extraction. We adopt the three-level ResNet-like architecture in [5]. For the first three layers, three convolutions of 3×3 kernel are used to downsample the input images. Then, 16 residual layers [26] are followed to produce unary features at 1/4 resolution, i.e., l_1. After that, we apply 6 residual layers with more channels to obtained large receptive fields and semantic information, i.e., l_2 and l_3. Finally, all feature maps (l_1, l_2, l_3) at 1/4 resolution are concatenated to form 320-channel feature maps for the generation of attention weights. Then two convolutions are applied to compress the 320-channel feature maps to 32-channel feature maps for construction of the initial concatenation volume, which are denoted as $\mathbf{f}_l$ and $\mathbf{f}_r$.

Attention Concatenation Volume Construction. We take the 320-channels feature maps for attention weights generation, and $\mathbf{f}_l$ and $\mathbf{f}_r$ for initial concatenation volume construction. Then attention weights are used to filter the initial concatenation volume to produce a 4D cost volume for all disparities, as described in Sec. 3.1.

Cost Aggregation. We process the ACV using a pre-hourglass module which consists of four 3D convolutions with batch normalization and ReLU, and two stacked 3D hourglass networks [5]. Each hourglass network consists of four 3D convolutions and two 3D deconvolutions stacked in an encoder-decoder architecture.

Disparity Prediction. We obtain three outputs from cost aggregation. For each output, following GwcNet [5], we convolve it using two 3D convolutions to output a 1-channel

Fig. 5: Visualization of the Fine-to-Important sampling strategy. Ideally, the disparity probability distribution should be unimodal peaked at true disparities. However, the actual probability distribution could be either unimodal or multimodal at some pixels. The red and blue triangles in the middle figure show two points with multimodal (left figure) and unimodal distributions (right figure) at the coarse stage respectively. The cascaded methods only sample the disparities near the regressed disparity of the coarse stage (denoted by blue lines), however, which could miss the true disparity for points with multimodal distributions. In contrast, Our sampling strategy (red dots) can preserve true disparities for points with both multimodal and unimodal distributions.

4D volume. Then we up-sample and convert it into a probability volume by the $softmax$ function along the disparity dimension. Finally, the predicted value is computed by the $soft\ argmin$ function [4]. The three predicted disparity maps are denoted as $\mathbf{d}_0$, $\mathbf{d}_1$, $\mathbf{d}_2$.

3.4 Fast-ACVNet Architecture

We construct a real-time stereo model, named Fast-ACVNet, based on our Fast-ACV. Fig. 2 shows the architecture of our Fast-ACVNet which consists of four steps of Multi-scale Feature Extraction, Fast Attention Concatenation Volume Construction, Cost Aggregation and Disparity Prediction.

Multi-scale Feature Extraction. Given an input stereo image pair whose size is $H \times W \times 3$, we use the MobileNetV2 pre-trained on ImageNet [27] to obtain four scales of feature maps whose resolutions are 1/4, 1/8, 1/16, and 1/32 of the original resolution respectively. Then we use three up-sampling blocks with skip-connections to increase the size of low-resolution feature maps of $H/32 \times W/32$, $H/16 \times W/16$ and $H/8 \times W/8$ resolution (see Fig.2). Finally, we obtain $H/8 \times W/8$ resolution feature maps for generation of attention weights, and $H/4 \times W/4$ resolution feature maps for construction of the compact concatenation volume.

Fast Attention Concatenation Volume Construction. We take 96-channels feature maps at 1/8 resolution for attention weights generation, and $\mathbf{f}_l^F$ and $\mathbf{f}_r^F$ at 1/4 resolution for compact concatenation volume construction. We divide the 96 channels of feature maps at 1/8 resolution into 12 groups, each group contains 8 channels of feature maps. Then we construct a group-wise correlation volume as [5]. We utilize a lightweight guided hourglass network same as CoEx [28] to regularize the low-resolution group-wise correlation volume. Then we apply VAP to the regularized correlation volume to obtain the volume $\mathbf{V}^p$ at 1/4 resolution. We further convert $\mathbf{V}^p$ into a probability volume to get the attention weights $\mathbf{A}^F \in \mathbb{R}^{K \times H/4 \times W/4}$ by using the Top-K probability values of $\mathbf{V}^p$ at every pixel. The attention weights are used to filter the compact concatenation volume to produce Fast-ACV, as described in Sec. 3.2.

Cost Aggregation. We aggregate the Fast-ACV using a guided hourglass network, which consists of six 3D convolutions and two 3D deconvolutions stacked in an encoder-decoder architecture. Following CoEx [28], we utilize extracted feature maps from the left image as guidance for cost aggregation.

Disparity Prediction. We take out only the top 2 values at every pixel of the aggregated cost volume and perform $softmax$ on these values to compute the expected disparity values. Finally, we up-sample the output disparity prediction to the original input image resolution by "superpixel" weights surrounding each pixel as [29].

3.5 Loss Function

For ACVNet, the final loss is given by,

$$L = \lambda_{att} \cdot \text{Smooth}_{L_1}(\mathbf{d}_{att} - \mathbf{d}^{gt}) + \sum_{i=0}^{i=2} \lambda_i \cdot \text{Smooth}_{L_1}(\mathbf{d}_i - \mathbf{d}^{gt}), \quad (14)$$

where $\mathbf{d}_{att}$ is obtained by attention weights $\mathbf{A}$ in Sec. 3.1. λ_{att} represents the coefficient for the predicted $\mathbf{d}_{att}$, λ_i represents the coefficient for the ith predicted disparity and $\mathbf{d}^{gt}$ denotes the ground-truth disparity map. The Smooth_{L_1} is the smooth L1 loss.

For Fast-ACVNet, the final loss is given by,

$$L^F = \lambda_{att}^F \cdot \text{Smooth}_{L_1}(\mathbf{d}_{att}^F - \mathbf{d}^{gt}) + \lambda^F \cdot \text{Smooth}_{L_1}(\mathbf{d}^F - \mathbf{d}^{gt}), \quad (15)$$

where $\mathbf{d}_{att}^F$ is obtained by compact attention weights for Fast-ACV, $\mathbf{d}^F$ is final output of Fast-ACVNet.

4 EXPERIMENT

4.1 Datasets and Evaluation Metrics

Scene Flow is a collection of synthetic stereo datasets which provides 35,454 training image pairs and 4,370 testing image pairs with the resolution of 960×540. This dataset provides dense disparity maps as ground truth. For Scene Flow [3] dataset, we utilized the widely-used evaluation metrics end-point error (EPE) and percentage of disparity outliers D1 as the evaluation metrics. The outliers are defined as the pixels whose disparity errors are greater than $\max(3px, 0.05d^*)$, where d^* denotes the ground-truth disparity.

TABLE 1: Ablation study of the ACV on Scene Flow [3].

Model	Attention Weights	Hourglass for Attention	Supervise for Attention	>1px (%)	>2px (%)	>3px (%)	D1 (%)	EPE (px)	FLOPs (G)	Params. (M)
GwcNet [5]				8.03	4.47	3.30	2.71	0.76	976.4	6.91
Gwc-att	✓			6.14	3.39	2.49	2.03	0.57	982.1	7.02
Gwc-att-hg	✓	✓		5.67	3.09	2.23	1.87	0.52	1071.9	7.40
Gwc-att-hg-s	✓	✓	✓	4.89	2.69	1.98	1.55	0.46	1071.9	7.40

KITTI includes KITTI 2012 [15] and KITTI 2015 [14], which are datasets for real-world driving scenes. KITTI 2012 contains 194 training pairs and 195 testing pairs, and KITTI 2015 contains 200 training pairs and 200 testing pairs. Both datasets provide sparse ground-truth disparities obtained with LIDAR. For KITTI 2012, we report the percentage of pixels with errors larger than x disparities in both non-occluded (x-noc) and all regions (x-all), as well as the overall EPE in both non occluded (EPE-noc) and all the pixels (EPE-all). For KITTI 2015, we report the percentage of pixels with EPE larger than 3 pixels in background regions (D1-bg), foreground regions (D1-fg), and all (D1-all).

ETH3D [16] is a collection of grayscale stereo pairs from indoor and outdoor scenes. It contains 27 training and 20 testing image pairs with sparse labeled ground-truth. Its disparity range is between 0 and 64. The percentage of pixels with errors larger than 2 pixels (bad 2.0) and 1 pixel (bad 1.0) are reported.

Middlebury [30] is an indoor dataset with 15 training image pairs and 15 testing image pairs with full, half, and quarter resolutions. Bad 2.0 (percentage of the points with absolute error larger than 2 pixel) are reported.

4.2 Implementation Details

We implement our methods with PyTorch and perform our experiments using NVIDIA RTX 3090 GPUs. For all the experiments, we use the Adam [31] optimizer, with $\beta_1 = 0.9$, $\beta_2 = 0.999$. For ACVNet, the coefficients of four outputs are set as λ_{att}=0.5, λ_0=0.5, λ_1=0.7, λ_2=1.0. For Fast-ACVNet, the coefficients of two outputs are set as λ_{att}^F=0.5, λ^F=1.0. For ACVNet on Scene Flow, we first train attention weights generation network for 64 epochs and then train the remaining network for another 64 epochs. Finally we train complete network for 64 epochs. The initial learning rate is set to 0.001 decayed by a factor of 2 after epoch 20, 32, 40, 48 and 56. For Fast-ACVNet on Scene Flow, we first train attention weights generation network for 24 epochs, and then train complete network for another 24 epochs. The initial learning rate is set to 0.001 decayed by a factor of 2 after epoch 10, 15, 18, and 21. For KITTI, we finetune the pre-trained Scene Flow model on the mixed KITTI 2012 and KITTI 2015 training sets for 500 epochs. The initial learning rate is 0.001 and decreases to 0.0001 half at the 300th epoch.

4.3 Ablation Study

In this section, we investigate different designs and configuration settings for ACV and Fast-ACV.

4.3.1 Attention Concatenation Volume.

We evaluate different strategies for constructing the ACV on Scene Flow [3]. We take GwcNet [5] for example and for other existing stereo matching methods based on cost

TABLE 2: Ablation study of VAP and Top-K sampling strategy of Fast-ACV on Scene Flow.

Base Model	VAP	Top-K for Att	D1 (%)	EPE (px)	Runtime (ms)
Fast-ACVNet (baseline)		48	2.51	0.66	49
	✓	48	2.32	0.62	50
	✓	32	2.40	0.63	43
	✓	24	2.49	0.64	39
	✓	16	2.60	0.69	35

TABLE 3: Performance of ACVNet when using different number of parameters in aggregation on Scene Flow [3]

Model	ACV	Hourglass Number	D1 (%)	EPE (px)	Params. (M)
GwcNet [5]		3	2.71	0.76	6.91
Gwc-acv-3	✓	3	1.55	0.46	7.40
Gwc-acv-1	✓	1	1.79	0.53	5.04
Gwc-acv-0	✓	0	2.08	0.59	3.86
ACVNet (Gwc-acv-2)	✓	2	1.59	0.48	6.22

volumes, the trend is similar and thus are omitted in this experiment. We replace the combined volume of GwcNet with our ACV and keep the subsequent aggregation and disparity prediction modules the same. Fig. 6 shows three different ways of constructing an ACV. Fig. 6 (a) directly averages the multi-level patch matching volume along the channel dimension and multiply it with the concatenation volume, denoted as Gwc-att. As shown in Tab. 1, just this simple approach can dramatically improve the accuracy of GwcNet. Apparently, when using the correlation volume to filter the concatenation volume, the accuracy of correlation volume is crucial and largely affects the final performance of the network, so we use an hourglass architecture of 3D convolutions to aggregate it, which is denoted as Gwc-att-hg shown in Fig. 6 (b). The results in Tab. 1 show that Gwc-att-hg improves D1 and EPE by 9.8% and 8.7% respectively compared with Gwc-att. To further explicitly constrain the correlation volume during training, we use the *softmax* and *soft argmin* functions for regression to obtain the predicted disparity, and use the ground truth to supervise the disparity, denoted as Gwc-att-hg-s shown in Fig. 6 (c). Compared with the Gwc-att-hg, Gwc-att-hg-s improves D1 and EPE by 17.1% and 11.5% respectively

TABLE 4: Performance of Fast-ACV on Scene Flow [3]

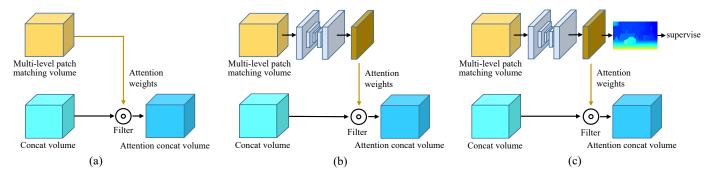| Model	Attention Filter	Hourglass Number	D1 (%)	EPE (px)	Runtime (ms)
Fast-ACV		1	0.83	3.56	39
		2	0.78	3.23	46
		3	0.74	2.92	52
	✓	1	0.64	2.49	39
	✓	2	0.62	2.33	46
	✓	3	0.60	2.20	52

Fig. 6: Illustration of different ways of constructing attention concatenation volume (ACV).

TABLE 5: Comparisons with the original cost volume and cascade cost volume.

Method	>1px (%)	>2px (%)	>3px (%)	EPE (px)	FLOPs (G)	Params. (M)	Runtime (ms)
PSMNet [13]	9.46	5.19	3.80	0.88	961.9	5.22	310
PSM+CAS [7]	7.44	4.61	3.50	0.72	1134.9	10.3	200
PSM+ACV	**7.35**	**4.12**	**3.01**	**0.63**	1052.4	5.53	355
PSM+Fast-ACV	7.41	4.22	3.10	0.65	**436.8**	**4.52**	**85**
GwcNet [5]	8.03	4.47	3.30	0.76	976.4	6.91	180
Gwc+CAS [7]	7.46	4.16	3.04	0.64	1248.8	10.7	220
Gwc+ACV	**4.89**	**2.69**	**1.98**	**0.46**	1071.9	7.40	200
Gwc+Fast-ACV	7.09	3.96	2.87	0.61	**456.5**	**5.35**	**81**

TABLE 6: Evaluation of VAP module embedded into other networks on Scene Flow. The image size is 960×512.

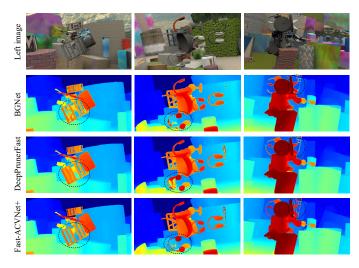| Model	>3px (%)	EPE (px)	FLOPs (G)
PSMNet [13]	3.80	0.88	961.9
PSMNet-VAP	**3.48**	**0.71**	964.2
GwcNet [5]	3.30	0.76	976.4
GwcNet-VAP	**2.52**	**0.59**	979.2

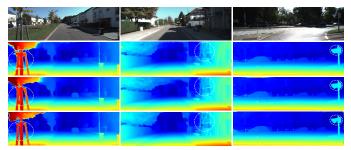Fig. 7: Qualitative results on Scene Flow.

Fig. 8: Qualitative results on KITTI 2012 and KITTI 2015. The second, third and fourth row shows the results of BGNet, DeepPrunerFast and Fast-ACVNet+ respectively.

with no computational cost increase in the inference stage. Overall, by replacing the combined volume in GwcNet with our ACV, our Gwc-att-hg-s model achieves 42.8% and 39.5% improvement for D1 and EPE compared with GwcNet, demonstrating the effectiveness of ACV.

4.3.2 Fast Attention Concatenation Volume.

We evaluate VAP and explore the Top-K sampling strategy on the Scene Flow dataset. In this ablation study, we use Fast-ACVNet without using VAP but using only bilinear upsample to increase the resolution of a correlation volume as our baseline. In the baseline model, we set the number of values sampled in the correlation volume $\mathbf{V}^p$ at each pixel (i.e., Top-K) as 48. Results in Tab. 2 show that our VAP can improve EPE from 0.66 to 0.62. We further explore the influence of the number of K in Top-K for the final accuracy and runtime. As shown in Tab. 2, we find that K=24 can achieve the best results in terms of accuracy and runtime. When K increases from 24 to 32 or 48, there is little increase in accuracy, but a large increase in runtime. Finally, we set Top-K to 24 for the final model of Fast-ACVNet.

4.4 Performance of ACV and Fast-ACV

ACV An ideal cost volume should require few parameters for subsequent aggregation network and meanwhile enable a satisfactory disparity prediction accuracy. We analyze the complexity of ACV in terms of the number of parameters demanded in the subsequent aggregation network and the corresponding accuracy. We use GwcNet [5] as the baseline. In original GwcNet, it uses three stacked hourglass networks for cost aggregation. We first replace the combined volume in the original GwcNet with our ACV with other parts remain the same. The corresponding model is denoted as Gwc-acv-3 in Tab. 3. The results show that compared with GwcNet, Gwc-acv-3 improves D1 and EPE by 42.8% and 39.5% respectively. We further reduce the number of hourglass networks of the aggregation network from 3 to 2, 1, and 0, the correspondingly derived models are denoted

TABLE 7: Comparison of ACVNet with state-of-the-art accuracy oriented methods on Scene Flow [3] and ETH3D [16]. **Bold**: Best, <u>Underscore</u>: Second best.

Model	Scene Flow EPE (px)	ETH3D Bad 1.0 (%)	ETH3D Bad 2.0 (%)
PSMNet [13]	1.09	5.02	1.09
GANet [23]	0.84	6.56	1.10
CFNet [6]	0.97	3.31	<u>0.77</u>
LEAStereo [32]	0.78	-	-
HITNet [25]	**0.43**	<u>2.79</u>	0.80
ACVNet (ours)	<u>0.48</u>	**2.58**	**0.57**

TABLE 8: Comparison of Fast-ACVNet with the state-of-the-art efficiency oriented methods on Scene Flow [3]

Model	EPE (px)	Runtime (ms)
DeepPrunerFast [11]	0.97	61
AANet [22]	0.87	62
BGNet [10]	1.17	28
DecNet [24]	0.84	50
CoEx [28]	0.69	33
Fast-ACVNet (ours)	<u>0.64</u>	39
Fast-ACVNet+ (ours)	**0.59**	45

as Gwc-acv-2, Gwc-acv-1 and Gwc-acv-0. The results in Tab. 3 show that, as the number of parameters reduced in the aggregation network, the prediction errors slightly increase. To achieve a both high accuracy and efficiency, we choose Gwc-acv-2 as our final model, and we denote it as ACVNet. **Fast-ACV** We demonstrate the effectiveness of our attention filtering and powerful expressive ability of our Fast-ACV. As shown in Tab. 4, our attention filter can improve EPE from 0.83 to 0.64 when there is only one hourglass network for aggregation. The Fast-ACV without attention filter, which is similar to the cascade cost volume, requires more hourglass aggregation networks to achieve performance gains due to the limited representation ability of the cost volume. While our Fast-ACV with only one hourglass aggregation network can achieve better accuracy than Fast-ACV without the attention filter and with three hourglass aggregation networks. Cascade cost volume methods reduce the memory and computational complexity of cost volume, however, for every stage, the cost volume needs to be re-constructed and re-aggregated with abundant 3D convolutions. In contrast, our Fast-ACV has higher representation ability and thus requires only much fewer aggregation networks.

4.5 Universality and Superiority of Our Methods

To demonstrate the universality and the superiority of our method, we integrate our cost volume into two state-of-the-art models, i.e. PSMNet [13] and GwcNet [5], and compare the performance of the original models with those after using our method. We denote the model after applying our method as PSM+ACV, PSM+Fast-ACV, Gwc+ACV and Gwc+Fast-ACV respectively, as shown in Tab. 5. We also experimentally compare our ACV and Fast-ACV with the cascaded approach. We apply the two-stage cascaded method proposed by [7] to PSMNet and GwcNet, the corresponding model is denoted as PSM+CAS and Gwc+CAS. The results in Tab. 5 show that, both our ACV and Fast-ACV can obviously improve the accuracy compared with the original model and and meanwhile outperform the cascaded cost volume [7]. In addition, our Fast-ACV can significantly

reduce the number of FLOPs, the number of parameters and runtime compared with the original model and [7]. We think the superior performance of Fast-ACV to the cascaded cost volume in terms of both accuracy and runtime is because that the latter need to re-aggregate a cost volume for each stage without reusing the prior information from the probability volume of the previous stage. As a result, it needs three hourglass networks for aggregation. In comparison, our Fast-ACV is more informative and expressive, which only need an hourglass network for aggregation.

It is noteworthy that our VAP module can also be easily integrated into stereo networks based on linear up-sampling cost volume in disparity prediction stage, such as PSMNet and GwcNet. As shown in Tab. 6, we integrate our VAP into PSMNet and GwcNet to derive PSMNet-VAP and GwcNet-VAP respectively. We observe that our VAP can achieve significant performance improvements with little extra computation.

4.6 Comparisons with State-of-the-art

Scene Flow and ETH3D As shown in Tab. 7, our ACVNet achieves the state-of-the-art performance. We can observe that our ACVNet improves EPE accuracy by 38.4% on Scene Flow with the state-of-the-art method LEAStereo [33]. On ETH3D, our ACVNet also achieve the state-of-the-art results. On Scene Flow, our Fast-ACVNet+ achieve the remarkable EPE of 0.59, which ourperforms all other real-time methods [10], [11], [22], [24], [28] at time of writing. Representative competitors are reported in Tab. 8. Qualitative results are shown in Fig. 7. Compared with Fast-ACVNet, Fast-ACVNet+ construct correlation volume at 1/4 resolution.

KITTI As shown in Tab. 9, our ACVNet ourperforms most exitsing published methods and achieves comparable accuracy with LEAStereo, but is faster than it, i.e., 200ms vs 300ms. Meanwhile, our Fast-ACVNet+ outperforms all the published real-time methods (i.e., inference time is smaller than 50ms) on the KITTI 2012 and 2015 benchmarks. Qualitative results are shown in Fig. 8. Fast-ACVNet+ also achieves comparable accuracy with HITNet [25], but with a faster inference speed, i.e., 45ms vs 54ms. In order to ensure the fairness of the comparison, the runtime of HITNet is tested on our hardware (RTX 3090) using the open-source models in PyTorch.

Runtime analysis We also report the runtime breakdown of our ACVNet, Fast-ACVNet and Fast-ACVNet+ based on images of KITTI 2015 whose image size is is 1242 × 375. Compared with Fast-ACVNet, Fast-ACVNet+ requires an additional 6ms due to constructing a 1/4 resolution correlation volume for Fast-ACV.

4.7 Generalization Performance

In addition to impressive performance on Scene Flow, ETH3D and KITTI, we also evaluate the generalization ability of our methods on the half-resolution training set of the Middlebury 2014 dataset [30] and the full-resolution training set of the KITTI 2012 and 2015. In this evaluation, all the comparison methods are only trained on the Scene Flow. Qualitative results are shown in Fig. 9. As shown in Tab. 11, our Fast-ACVNet achieves the state-of-the-art performance and outperforms other real-time methods.

TABLE 9: Quantitative evaluation on the test sets of KITTI 2012 [15] and KITTI 2015 [14]. We split the state-of-the-art methods into two parts according to the running time whether exceeds 100ms. The results of reference methods are obtained from the official declaration. ∗ denotes the runtime is tested on our hardware (RTX 3090).

Target	Method	KITTI 2012 [15]						KITTI 2015 [14]			Runtime (ms)
		3-noc	3-all	4-noc	4-all	EPE noc	EPE all	D1-bg	D1-fg	D1-all	
Accuracy	GCNet [4]	1.77	2.30	1.36	1.77	0.6	0.7	2.21	6.16	2.87	900
	PSMNet [13]	1.49	1.89	1.12	1.42	0.5	0.6	1.86	4.62	2.32	310*
	GwcNet [5]	1.32	1.70	0.99	1.27	0.5	0.5	1.74	3.93	2.11	180*
	GANet-deep [23]	1.19	1.60	0.91	1.23	0.4	0.5	1.48	3.46	1.81	180
	CFNet [6]	1.23	1.58	0.92	1.18	0.4	0.5	1.54	3.56	1.88	180
	LEAStereo [33]	**1.13**	**1.45**	**0.83**	**1.08**	0.5	0.5	<u>1.40</u>	**2.91**	**1.65**	300
	ACVNet (ours)	**1.13**	<u>1.47</u>	<u>0.86</u>	<u>1.12</u>	0.4	0.5	**1.37**	<u>3.07</u>	**1.65**	200
Speed	DispNetC [3]	4.11	4.65	2.77	3.20	0.9	1.0	4.32	4.41	4.34	60
	DeepPrunerFast [11]	-	-	-	-	-	-	2.32	3.91	2.59	50*
	AANet [22]	1.91	2.42	1.46	1.87	0.5	0.6	1.99	5.39	2.55	62
	DecNet [24]	-	-	-	-	-	-	2.07	3.87	2.37	50
	BGNet [10]	1.77	2.15	-	-	0.6	0.6	2.07	4.74	2.51	28*
	BGNet+ [10]	1.62	2.03	1.16	1.48	0.5	0.6	1.81	4.09	2.19	35*
	CoEx [28]	1.55	1.93	1.15	1.42	0.5	0.5	1.79	3.82	2.13	33*
	HITNet [25]	**1.41**	<u>1.89</u>	<u>1.14</u>	<u>1.53</u>	0.4	0.5	<u>1.74</u>	3.20	**1.98**	54*
	Fast-ACVNet+ (ours)	<u>1.45</u>	**1.85**	**1.06**	**1.36**	0.5	0.5	**1.70**	<u>3.53</u>	<u>2.01</u>	45
	Fast-ACVNet (ours)	1.68	2.13	1.23	1.56	0.5	0.6	1.82	3.93	2.17	39

TABLE 10: Runtime (ms) analysis of ACVNet, Fast-ACVNet and Fast-ACVNet+ on KITTI 2015. The size of input stereo images is 1242×375.

Module	ACVNet	Fast-ACVNet	Fast-ACVNet+
Feature Extraction	24	16	16
ACV Construction	58	8	14
Cost Aggregation	85	9	9
Disparity Prediction	33	6	6

TABLE 11: Generalization evaluation on KITTI and Middlebury training sets. All models are only trained on Scene Flow and tested on training images of two real datasets.

Model	KITTI 2012 D1-all (%)	KITTI 2015 D1-all (%)	Middlebury Bad 2.0 (%)
PSMNet [13]	15.1	16.3	30.25
DeepPrunerFast [11]	16.8	15.9	30.83
BGNet [10]	24.8	20.1	37.00
CoEx [28]	13.5	11.6	25.51
Fast-ACVNet (ours)	**12.4**	**10.6**	**20.13**

5 CONCLUSION

In this paper, we propose a novel cost volume construction method, named attention concatenation volume (ACV), and its real-time version Fast-ACV, which generates attention weights based on similarity measures to filter the concatenation volume. Based on ACV and Fast-ACV, we design a highly accurate network ACVNet and real-time network Fast-ACVNet, which achieve the state-of-the-art performance on several benchmarks (i.e., our ACVNet ranks the 2nd on KITTI 2015 and Scene Flow, and the 3rd on KITTI 2012 and ETH3D; our Fast-ACVNet outperforms almost all the state-of-the-art real-time methods on Scene Flow, KITTI 2012 and 2015 and meanwhile has the best generalization ability among all real-time methods). Our ACV and Fast-ACV are general cost volume representation that can be integrated into many existing stereo matching models based on 4D cost volumes for performance improvement. In addition, the idea of our Fast-ACV, which exploiting prior information in the probability volume of the previous
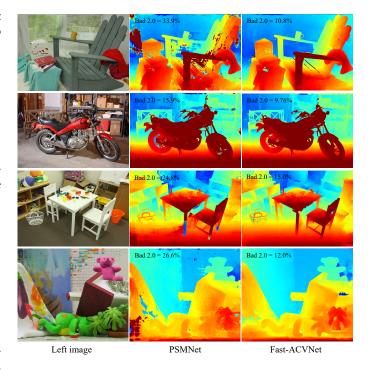

Fig. 9: Qualitative results of generalization performance evaluation on the Middlebury 2014 dataset.

stage as attention weights to filter the cost volume of the current stage, can be also applied to the cascade cost volume methods to achieve accuracy gain and greatly reduce the amount of computation and parameters.

REFERENCES

[1] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE TPAMI*, vol. 30, no. 2, pp. 328–341, 2007.

[2] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *IJCV*, vol. 47, no. 1, pp. 7–42, 2002.

[3] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *CVPR*, 2016, pp. 4040–4048.

[4] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry, "End-to-end learning of geometry and context for deep stereo regression," in *ICCV*, 2017, pp. 66–75.

[5] X. Guo, K. Yang, W. Yang, X. Wang, and H. Li, "Group-wise correlation stereo network," in *CVPR*, 2019, pp. 3273–3282.

[6] Z. Shen, Y. Dai, and Z. Rao, "Cfnet: Cascade and fused cost volume for robust stereo matching," in *CVPR*, 2021, pp. 13 906–13 915.

[7] X. Gu, Z. Fan, S. Zhu, Z. Dai, F. Tan, and P. Tan, "Cascade cost volume for high-resolution multi-view stereo and stereo matching," in *CVPR*, 2020, pp. 2495–2504.

[8] S. Cheng, Z. Xu, S. Zhu, Z. Li, L. E. Li, R. Ramamoorthi, and H. Su, "Deep stereo using adaptive thin volume representation with uncertainty awareness," in *CVPR*, 2020, pp. 2524–2534.

[9] S. Khamis, S. Fanello, C. Rhemann, A. Kowdle, J. Valentin, and S. Izadi, "Stereonet: Guided hierarchical refinement for edge-aware depth prediction," in *ECCV*, 2018.

[10] B. Xu, Y. Xu, X. Yang, W. Jia, and Y. Guo, "Bilateral grid learning for stereo matching networks," in *CVPR*, 2021, pp. 12 497–12 506.

[11] S. Duggal, S. Wang, W.-C. Ma, R. Hu, and R. Urtasun, "Deeppruner: Learning efficient stereo matching via differentiable patchmatch," in *ICCV*, 2019, pp. 4384–4393.

[12] F. Besse, C. Rother, A. Fitzgibbon, and J. Kautz, "Pmbp: Patch-match belief propagation for correspondence field estimation," *IJCV*, vol. 110, no. 1, pp. 2–13, 2014.

[13] J.-R. Chang and Y.-S. Chen, "Pyramid stereo matching network," in *CVPR*, 2018, pp. 5410–5418.

[14] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *CVPR*, 2015, pp. 3061–3070.

[15] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *CVPR*. IEEE, 2012, pp. 3354–3361.

[16] T. Schops, J. L. Schonberger, S. Galliani, T. Sattler, K. Schindler, M. Pollefeys, and A. Geiger, "A multi-view stereo benchmark with high-resolution images and multi-camera videos," in *CVPR*, 2017, pp. 3260–3269.

[17] Z. Liang, Y. Guo, Y. Feng, W. Chen, L. Qiao, L. Zhou, J. Zhang, and H. Liu, "Stereo matching using multi-level cost volume and multi-scale feature constancy," *IEEE TPAMI*, 2019.

[18] X. Cheng, P. Wang, and R. Yang, "Learning depth with convolutional spatial propagation network," *IEEE TPAMI*, vol. 42, no. 10, pp. 2361–2379, 2019.

[19] L. Wang, Y. Guo, Y. Wang, Z. Liang, Z. Lin, J. Yang, and W. An, "Parallax attention for unsupervised stereo correspondence learning," *IEEE TPAMI*, 2020.

[20] Z. Wu, X. Wu, X. Zhang, S. Wang, and L. Ju, "Semantic stereo matching with pyramid cost volumes," in *ICCV*, 2019, pp. 7484–7493.

[21] Y. Zhang, Y. Chen, X. Bai, S. Yu, K. Yu, Z. Li, and K. Yang, "Adaptive unimodal cost volume filtering for deep stereo matching," in *AAAI*, vol. 34, no. 07, 2020, pp. 12 926–12 934.

[22] H. Xu and J. Zhang, "Aanet: Adaptive aggregation network for efficient stereo matching," in *CVPR*, 2020, pp. 1959–1968.

[23] F. Zhang, V. Prisacariu, R. Yang, and P. H. Torr, "Ga-net: Guided aggregation net for end-to-end stereo matching," in *CVPR*, 2019, pp. 185–194.

[24] C. Yao, Y. Jia, H. Di, P. Li, and Y. Wu, "A decomposition model for stereo matching," in *CVPR*, 2021, pp. 6091–6100.

[25] V. Tankovich, C. Hane, Y. Zhang, A. Kowdle, S. Fanello, and S. Bouaziz, "Hitnet: Hierarchical iterative tile refinement network for real-time stereo matching," in *CVPR*, 2021, pp. 14 362–14 372.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[27] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*. Ieee, 2009, pp. 248–255.

[28] A. Bangunharcana, J. W. Cho, S. Lee, I. S. Kweon, K.-S. Kim, and S. Kim, "Correlate-and-excite: Real-time stereo matching via guided cost volume excitation," in *IROS*. IEEE, 2021, pp. 3542–3548.

[29] F. Yang, Q. Sun, H. Jin, and Z. Zhou, "Superpixel segmentation with fully convolutional networks," in *CVPR*, 2020, pp. 13 964–13 973.

[30] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nešić, X. Wang, and P. Westling, "High-resolution stereo datasets with subpixel-accurate ground truth," in *GCPR*. Springer, 2014, pp. 31–42.

[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[32] X. Cheng, Y. Zhong, M. Harandi, Y. Dai, X. Chang, T. Drummond, H. Li, and Z. Ge, "Hierarchical neural architecture search for deep stereo matching," *arXiv preprint arXiv:2010.13501*, 2020.

[33] X. Cheng, Y. Zhong, M. Harandi, Y. Dai, X. Chang, H. Li, T. Drummond, and Z. Ge, "Hierarchical neural architecture search for deep stereo matching," *NeurIPS*, vol. 33, pp. 22 158–22 169, 2020.

Gangwei Xu is a first-year master at the Department of Electronic Information and Communications at Huazhong University of Science and Technology. He is supervised by Prof. Xin Yang. He received the B.Eng. degree from Huazhong University of Science and Technology in 2021. His research interest focus on stereo matching and 3D vision. He has published one paper in CVPR.

Yun Wang is a first-year master at the Department of Electronic Information and Communications at Huazhong University of Science and Technology. She is supervised by Prof. Xin Yang. She received the B.Eng. degree from Huazhong University of Science and Technology in 2020. Her research interest focus on stereo matching and 3D vision.

Junda Cheng is a second-year master at the Department of Electronic Information and Communications at Huazhong University of Science and Technology. She is supervised by Prof. Xin Yang. She received the B.Eng. degree from Huazhong University of Science and Technology in 2020. Her research interest focus on stereo matching and 3D vision. He has published one paper in CVPR.

Jinhui Tang is a Professor at the Nanjing University of Science and Technology. He received the B.Eng. and Ph.D. degrees from the University of Science and Technology of China in 2003 and 2008, respectively. He has authored over 150 papers in top-tier journals and conferences, with more than 10,000 citations in Google Scholar. His research interests include multimedia analysis and computer vision. He was a recipient of the best paper awards in ACM MM 2007, PCM 2011 and ICIMCS 2011, the Best Paper Runner-up in ACM MM 2015, and the best student paper awards in MMM 2016 and ICIMCS 2017. He has served as an Associate Editor for the IEEE TNNLS, IEEE TKDE, IEEE TMM and IEEE TCSVT.

Xin Yang is a Professor at the Department of Electronic Information and Communications at Huazhong University of Science and Technology. She received her Ph.D. degree in the Department of Electrical Computer Engineering at the University of California, Santa Barbara (UCSB). Her research interests include medical image analysis and 3D vision. She is the recipient of the National Natural Science Fund of China for Excellent Youth Scholar and China Society of Image and Graphics Qingyun Shi Female Scientist Award. She has published over 90 technical papers and held 12 patents. She serves as an Associate Editor of IEEE-TMI and Multimedia System, an Area Chair of MICCAI'19-21 and ACM MM'18, and a PC member of CVPR, ECCV and ICCV. She is also a reviewer of top journals such as IEEE-TPAMI, IEEE-TNNLS, MedIA, etc.