sensors

MDPI

Article

Efficient Multi-Scale Stereo-Matching Network Using Adaptive Cost Volume Filtering

Suyeon Jeon [1] and Yong Seok Heo [1,2,*]

[1] Department of Artificial Intelligence, Ajou University, Suwon 16499, Korea; suyeon1804@ajou.ac.kr
[2] Department of Electrical and Computer Engineering, Ajou University, Suwon 16499, Korea
* Corresponding author: ysheo@ajou.ac.kr

Abstract: While recent deep learning-based stereo-matching networks have shown outstanding advances, there are still some unsolved challenges. First, most state-of-the-art stereo models employ 3D convolutions for 4D cost volume aggregation, which limit the deployment of networks for resource-limited mobile environments owing to heavy consumption of computation and memory. Although there are some efficient networks, most of them still require a heavy computational cost to incorporate them to mobile computing devices in real-time. Second, most stereo networks indirectly supervise cost volumes through disparity regression loss by using the softargmax function. This causes problems in ambiguous regions, such as the boundaries of objects, because there are many possibilities for unreasonable cost distributions which result in overfitting problem. A few works deal with this problem by generating artificial cost distribution using only the ground truth disparity value that is insufficient to fully regularize the cost volume. To address these problems, we first propose an efficient multi-scale sequential feature fusion network (MSFFNet). Specifically, we connect multi-scale SFF modules in parallel with a cross-scale fusion function to generate a set of cost volumes with different scales. These cost volumes are then effectively combined using the proposed interlaced concatenation method. Second, we propose an adaptive cost-volume-filtering (ACVF) loss function that directly supervises our estimated cost volume. The proposed ACVF loss directly adds constraints to the cost volume using the probability distribution generated from the ground truth disparity map and that estimated from the teacher network which achieves higher accuracy. Results of several experiments using representative datasets for stereo matching show that our proposed method is more efficient than previous methods. Our network architecture consumes fewer parameters and generates reasonable disparity maps with faster speed compared with the existing state-of-the art stereo models. Concretely, our network achieves 1.01 EPE with runtime of 42 ms, 2.92 M parameters, and 97.96 G FLOPs on the Scene Flow test set. Compared with PSMNet, our method is 89% faster and 7% more accurate with 45% fewer parameters.

Keywords: deep learning; stereo matching; knowledge distillation; cost volume filtering; lightweight network

Citation: Jeon, S.; Heo, Y.S. Efficient Multi-Scale Stereo-Matching Network Using Adaptive Cost Volume Filtering. *Sensors* **2022**, *22*, 5500. https://doi.org/10.3390/s22155500

Academic Editor: Carlos M. Travieso-González

Received: 26 June 2022
Accepted: 21 July 2022
Published: 23 July 2022

1. Introduction

Estimating depth from a stereo image has been a fundamental and classic computer vision problem for decades [1]. Stereo matching aims to estimate the correspondence between the pixels of a rectified stereo image pair. If a pixel at (x, y) in the reference left image matches a pixel at $(x - d, y)$ in the target right image, the horizontal difference d between the corresponding pixels is a disparity. Using disparity d, camera focal length f, and distance between cameras B, the depth of the pixel can be calculated as $\frac{fB}{d}$. Because 3D depth information is essential for various real-world applications, including robot navigation [2], augmented/virtual reality (AR/VR) [3,4], autonomous driving for vehicles [5], and network security domains [6,7], reliable real-time stereo matching processing in restricted hardware environments is important.

Since the introduction of the seminal work known as MC-CNN [8], convolutional neural networks (CNNs) have been used to learn strong feature representation [8,9] to compute the cost of corresponding patches in the input stereo image pair. Consequently, they achieve more significant increases in accuracy than traditional stereo matching algorithms. However, because most of them adopt handcrafted cost aggregation methods and do not realize the fully end-to-end networks, they suffer from errors for ambiguous regions such as textureless regions.

DispNet [10] presented the first end-to-end stereo-matching network by building a 3D cost volume with a correlation layer followed by 2D convolution layers to estimate a disparity map and showed significant improvement in accuracy compared with previous patch-wise CNNs [8,9]. Instead of the 3D cost volume with the correlation layer, refs. [11,12] built a 4D cost volume by concatenating the left and right feature maps along the disparity levels. Subsequently, by processing the cost volume through a series of 3D convolution layers, they achieved better accuracy than 3D cost-volume-based methods. Although these networks that process 4D cost volume with 3D convolutions achieved state-of-the-art accuracy, heavy computation and memory consumption prevent them from running in mobile environments and real-time processing.

To address this problem, various efficient stereo-matching networks [13–17] have recently been proposed. Although these networks have significantly improved performance in terms of efficiency compared with previous networks, they still require a heavy computational cost to incorporate them to mobile computing devices in real time. Even though some of them are sufficiently light to run in mobile environments, there is a significant decrease in accuracy.

On the other hand, since DispNet [11] proposed the softargmin function that robustly calculates continuous disparity values with subpixel precision from a cost distribution, it has been adopted in most stereo-matching networks to regress a continuous disparity map. Most of them trained their networks in an end-to-end manner by defining the loss function using the difference between the predictive disparity map and the ground truth disparity map. The networks learn the cost distribution in an indirect supervision manner by reducing this loss function. Because this indirect supervision results in many possibilities for unreasonable cost distributions as long as the result after regression is correct, learning with these flexible cost volumes leads to overfitting [18]. Thus, direct supervision is required for cost volume regularization with probability distribution which peaks at the true disparity especially at the pixels where a multi-modal probability distribution is predicted, such as the boundaries of the object [19].

There are a few works that handle this problem. AcfNet [18] adds a direct constraint to the predicted cost volume by generating a unimodal probability distribution that peaks at the ground truth disparity and filters the predicted cost volume through it. CDN [19] addresses the problem by directly taking the mode as a predicted disparity value without a regression function, such as the softargmin operation. However, these methods still perform insufficient regularization for the cost volume because the artificial cost distribution generated using only the ground truth disparity value does not contain information such as similarity among different disparities [20–23].

In this paper, to solve the problems mentioned above, we propose a new efficient network architecture called a Multi-scale Sequential Feature Fusion Network (MSFFNet) and a new loss function called adaptive cost-volume-filtering (ACVF) loss for direct cost volume supervision. MSFFNet consists of a proposed Multi-scale Sequential Feature Fusion (MSFF) module which connects SFF modules [17] with different scales in parallel for an efficient and accurate multi-scale network architecture. For higher efficiency, only the smallest scale in the MSFF module is processed for the entire disparity search range. The other scales are processed for the odd disparities among the entire disparity range. In addition, the cross-scale fusion [24] is adopted at the end of the MSFF module to compensate for the lack of scale information and allow each cost volume with different scale to interact

with each other. In addition, to effectively combine the multi-scale cost volumes generated from different scales, we present an interlaced cost concatenation method.

Moreover, the proposed ACVF loss adds a constraint to the estimated cost distribution, using both the ground truth disparity map and the probability distribution map generated from the teacher network with higher accuracy. Thus, a unimodal distribution is generated with the ground truth disparity value [18], and the distribution generated from the teacher network is utilized for knowledge distribution [21–23] which transfers the *dark knowledge* of the cumbersome teacher network to our network as student. However, there are some pixels for which the teacher network is not accurate and that negatively affect the student network. To avoid the negative effects of the distillation [22], the distribution from the teacher network is adaptively transferred using the ratio between the errors of the teacher and our proposed networks for each pixel. Because the proposed loss function does not require additional parameters and computations for inference, it allows us to consider the effectiveness of the direct supervision of the cost volume while maintaining the efficiency of our network. As shown in Figure 1, our proposed network predicts a reasonably accurate disparity map with an extremely fast runtime using small numbers of parameters compared with other efficient stereo-matching networks.

The main contributions of this paper are summarized as follows:

1. We propose an efficient multi-scale stereo-matching network, MSFFNet, that connects multi-scale SFF modules in parallel and effectively generates a cost volume using the proposed interlaced cost concatenation method.
2. We propose an adaptive cost-volume-filtering loss that adaptively filters the estimated cost volume using a ground truth disparity map and an accurate teacher network for direct supervision of the estimated cost volume.
3. We achieved competitive results on the Scene Flow [10] and KITTI-2015 [25] test sets. The proposed MSFFNet has only 2.92 M parameters with a runtime of 42 ms, which shows that our method is more efficient and reasonably accurate compared with other stereo-matching networks.

The following sections of this paper are organized as follows. We introduce several related works in Section 2. Section 3 introduces the details about methodology and implementation of our proposed method. Various experiments including comparative results with previous methods and ablation studies are demonstrated in Section 4. Finally we conclude the paper in Section 5.

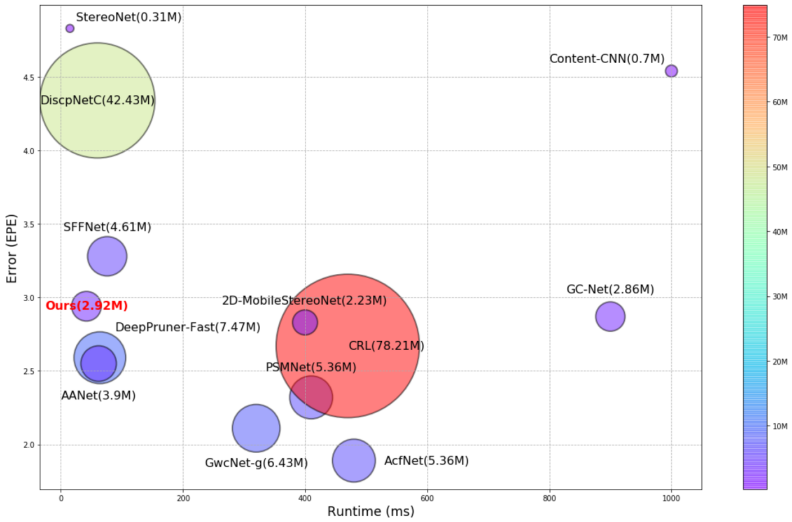

Figure 1. Visualization of runtime, accuracy, and number of parameters of stereo-matching networks, including Content-CNN [9], GC-Net [11], PSMNet [12], StereoNet [13], GwcNet-g [15], DeepPruner-Fast [26], AANet [16], AcfNet [18], DispNetC [10], CRL [27], SFFNet [17], 2D-MobileStereoNet [28], and our proposed method, on KITTI 2015 leader board [25].

2. Related Works

Our proposed method includes an efficient network architecture for stereo matching and a new loss function which filters the cost volume for increasing accuracy. Thus, in this section, the relevant works about efficient stereo-matching networks and cost volume filtering are separately discussed. Table 1 summarizes the representative related works in terms of types of cost volume, network architecture, cost volume filtering, and datasets used for training such as Scene Flow [10] and KITTI [25,29].

Table 1. A comparison of relevant stereo-matching networks.

Network	Cost Volume	Architecture	Cost Volume Filtering	Dataset
MC-CNN [8]	4D	Single scale	X	KITTI
DispNetC [10]	3D	Single scale	X	Scene Flow, KITTI
PSMNet [12]	4D	Single scale	X	Scene Flow, KITTI
AnyNet [14]	4D	Multi scale Sequential	X	Scene Flow, KITTI
AANet [16]	4D	Multi scale Parallel	X	Scene Flow, KITTI
Msmd-net [30]	4D	Multi scale Parallel	X	Scene Flow, KITTI
CompactStereoNet [31]	4D	Multi scale Sequential	X	Scene Flow, KITTI
UASNet [32]	4D	Multi scale Sequential	X	Scene Flow, KITTI
SFFNet [17]	3D	Single scale	X	Scene Flow, KITTI
AcfNet [18]	4D	Single scale	O	Scene Flow, KITTI
CDN [19]	4D	Single scale	O	Scene Flow, KITTI
2D-MobileStereoNet [28]	4D	Single scale	X	Scene Flow, KITTI

2.1. Efficient Stereo-Matching Networks

Since the first end-to-end training for a stereo-matching network [10] was proposed, stereo-matching networks usually generate a single-scale cost volume for matching cost computation. Refs. [10,27,33,34] constructed a 3D cost volume using a correlation operation between corresponding left and right features. Although a large amount of useful feature information for cost aggregation is lost through correlation, there is an advantage in terms of computational cost because the generated 3D cost volume is processed using 2D convolutions. Meanwhile, refs. [8,12] generated a 4D cost volume by concatenating two corresponding features and achieved higher accuracy compared with 3D cost-volume-based methods. However, 3D convolutions are required for processing the 4D cost volume, leading to a significant increase in computing resources and runtime.

There is a trade-off between efficiency and accuracy according to the choice of the dimension of the cost volume. Thus, various efficient stereo-matching networks have been studied extensively in recent years [14,16]. To efficiently process the 4D cost volume with 3D convolutions, several networks adopted a coarse to fine architecture that leverages multi-scale pyramid cost volumes. This method can enhance accuracy without expensive computational complexity and memory consumption. Moreover, multi-scale cost volumes are more sufficient to utilize spatial information of the stereo image pair. These networks using multi-scale cost volumes are classified as sequential method and parallel method according to the type of connecting multi-scale feature maps.

Sequential method networks [14,31,32] generate and aggregate the cost volume over the full disparity range only at the initial coarsest scale. Then, to correct the error of the disparity map generated from the previous scale, the succeeding networks warp the right feature map using the disparity map and process the residuals that cover only the offset range of each disparity search range. The rough disparity map from the network that covers the coarse scale is iteratively refined through the succeeding networks towards the bottom of the pyramid in the original resolution by adding the residual to the previous disparity result. Because the computational complexity of cost aggregation with 3D convolution increases cubically with resolution and linearly with disparity range, this sequential method considerably reduces the computational cost and memory consumption. However, the sequential method has the following disadvantages. The cost aggregation conducted only at the first stage with the smallest resolution is insufficient to estimate accurate disparity for sharp regions. In addition, the refinement using the networks of succeeding stages with respect to the offset range can propagate errors when the error is larger than the offset.

Parallel method networks [16,30] perform aggregation and estimate the disparity map at each scale of the multi-scale cost volume. These methods can capture both robust global features and detailed local feature representations by integrating all the different scales of cost volumes. Parallel method networks regularize cost volumes and predict disparity maps with high accuracy by utilizing these rich representations. Although there is an advantage in terms of accuracy, they still require a large amount of computation because the cost aggregation is performed at all scales, even though it has decreased compared with the single scale-based networks [10,17].

Our network efficiently combines the sequential and parallel methods. To make our network more efficient than other parallel networks, we process the entire disparity search range only at the initial coarsest scale similar to the sequential methods. The other higher scales are processed only half of the full disparity range. In addition, using a new interlaced concatenation method, we effectively combine the multi-scale cost volumes which include the entire disparity range.

2.2. Cost Volume Filtering

Most deep-learning-based stereo-matching networks use loss functions based on the difference between the true and predicted disparities because they consider stereo matching as a regression problem. However, there are only a few methods that supervise the cost volume. Because a disparity is the regression result from the cost volume, learning cost volume indirectly through the disparity regression loss easily causes an overfitting problem [18,19], especially around the edge regions.

To mitigate this problem, the cost volume filtering for direct constraints on the cost volume was first proposed in AcfNet [18]. Through the addition of a cost-volume-filtering module, which filters the estimated cost volume with a unimodal distribution that peaks at the ground truth disparity, direct supervision of the cost volume is conducted. CDN [19] also addresses this problem by proposing a new network architecture for stereo matching that can estimates arbitrary disparity values. Thus, the mode value can be directly chosen as a disparity without a regression step.

Unlike refs. [18,19] which uses the artificially generated cost distributions using the ground truth disparity map, we perform direct supervision to the cost volume through a loss function which includes two kinds of filtering using the cost volume from the ground truth disparity map and that estimated from the teacher network with higher accuracy via knowledge distillation.

3. Proposed Method

Figure 2 presents an overview of the proposed network. Given a rectified stereo image pair, each image is inserted into a U-Net [35] feature extractor to generate multi-scale feature maps. Using these feature maps as inputs, a cost aggregation module aggregates the matching costs and generates multi-scale cost volumes, where a MSFF module is proposed

to efficiently generate them by connecting the SFF [17] modules of different scales in parallel. The final cost volume for disparity regression is obtained by concatenating the cost volume using the proposed interlaced concatenation method. Finally, an initial disparity map is regressed with the softargmax function from the cost volume and hierarchically upsampled and refined with a refinement network [13] to produce a final disparity map. To train the network, we define a loss function that consists of a disparity regression loss and an adaptive cost-volume-filtering (ACVF) loss. Specifically, to supervise the cost volume, the proposed ACVF utilizes both the ground truth disparity map and probability distribution map generated from a teacher network with higher accuracy. Detailed explanations of each part are provided in the following subsections.

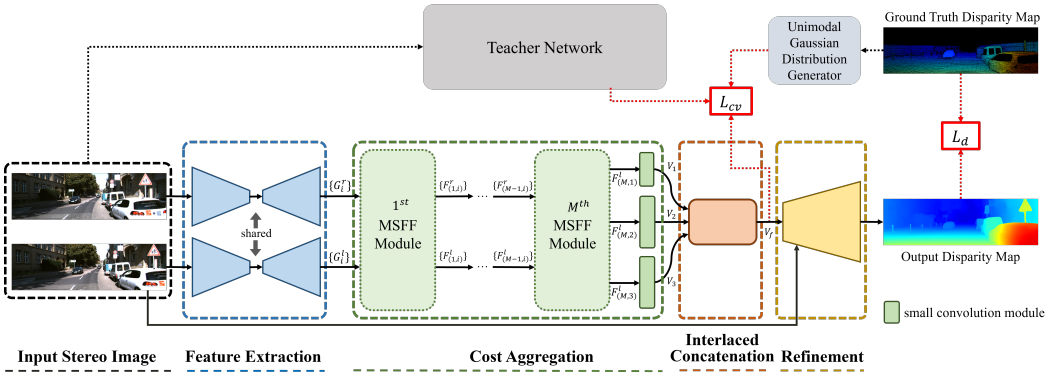

Figure 2. An overview of the proposed method. A path using dotted arrows is required only for the training phase.

3.1. Feature Extractor

Given a rectified stereo image pair $I_l \in \mathbb{R}^{W\times H\times 3}$ and $I_r \in \mathbb{R}^{W\times H\times 3}$, each image is inserted into a U-Net [35] feature extractor, to generate feature maps $\left\{G_i^l \in \mathbb{R}^{W_i\times H_i\times C_i}\right\}_{i=1}^{3}$ and $\left\{G_i^r \in \mathbb{R}^{W_i\times H_i\times C_i}\right\}_{i=1}^{3}$, respectively. Here, i represents the scale index, and W_i and H_i are $1/2^{(5-i)}$ times the width and height of the original input image, respectively. The channel size C_i is fixed to 32 for all scales. Similar to [14,16], G_i^l and G_i^r are extracted from the different positions of the decoder of the feature extractor network, where the corresponding spatial resolutions of the feature maps for each scale are 1/16, 1/8, and 1/4 times the original spatial resolution of the input image, respectively.

3.2. Cost Aggregation

From the multi-scale feature maps $\{G_i^l\}_{i=1}^3$ and $\{G_i^r\}_{i=1}^3$ obtained using the feature extractor module, the proposed cost aggregation module generates various cost volumes $\{V_i\}_{i=1}^3$ with different spatial resolutions. To this end, we propose an MSFF module that acts as a building block for the cost aggregation module. Concretely, the cost aggregation module consists of a series of M MSFF modules followed by a small convolution layer. As shown in Figure 3, each MSFF module consists of combined SFF modules [17] $\{S_i\}_{i=1}^3$ with different scales and a cross-scale fusion operation that combines them so that they share different scale information. Specifically, n^{th} MSFF module generates feature maps $\left\{F_{(n,i)}^l \in \mathbb{R}^{W_i\times H_i\times 32}\right\}_{i=1}^{3}$ and $\left\{F_{(n,i)}^r \in \mathbb{R}^{W_i\times H_i\times 32}\right\}_{i=1}^{3}$ which are inserted to the next $n+1^{th}$ MSFF module from $\{F_{(n-1,i)}^l\}_{i=1}^3$ and $\{F_{(n-1,i)}^r\}_{i=1}^3$, where the input of the first MSFF module is the feature maps $\{G_i^l\}_{i=1}^3$ and $\{G_i^r\}_{i=1}^3$, that is, $F_{(0,i)}^l = G_i^l$ and $F_{(0,i)}^r = G_i^r$. In the n^{th} MSFF module, S_i generates the intermediate feature maps $\tilde{F}_{(n,i)}^l$ and $\tilde{F}_{(n,i)}^r$ from $F_{(n,i)}^l$ and $F_{(n,i)}^r$. To prevent heavy computations and memory consumption, only S_1 with the smallest scale processes the entire disparity range, whereas the others, including S_2 and S_3, process only odd disparities of the full search range. In more detail, only S_1 processes the entire disparity range, which amounts to 1/16 of the maximum disparity range through two SFF

modules [17]. Thus, as shown in Figure 4, the input right feature $F^r_{(n,1)}$ is shifted by only one pixel to the right at a time in the SFF module. However, as shown in Figure 5, S_2 and S_3 process only odd disparities by shifting $F^r_{(n,2)}$ and $F^r_{(n,3)}$ 2 pixels to the right, respectively, unlike the original SFF module [17]. This reduces the computation and parameters of S_2 and S_3 by half compared with processing the entire disparity range at that scale.

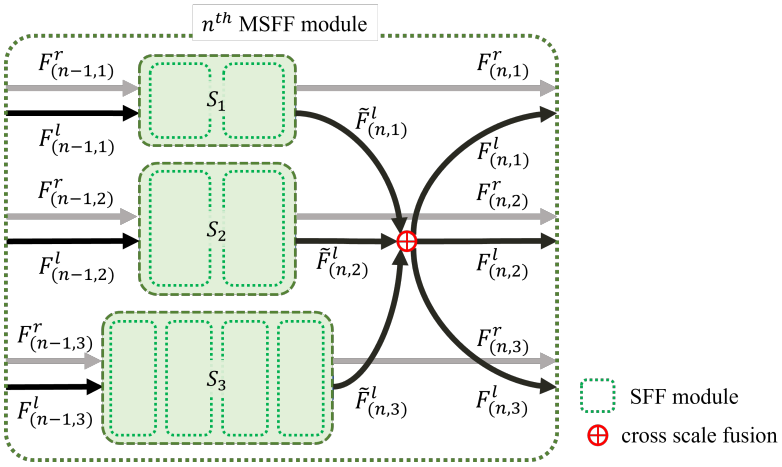

Figure 3. Details of the proposed MSFF module.

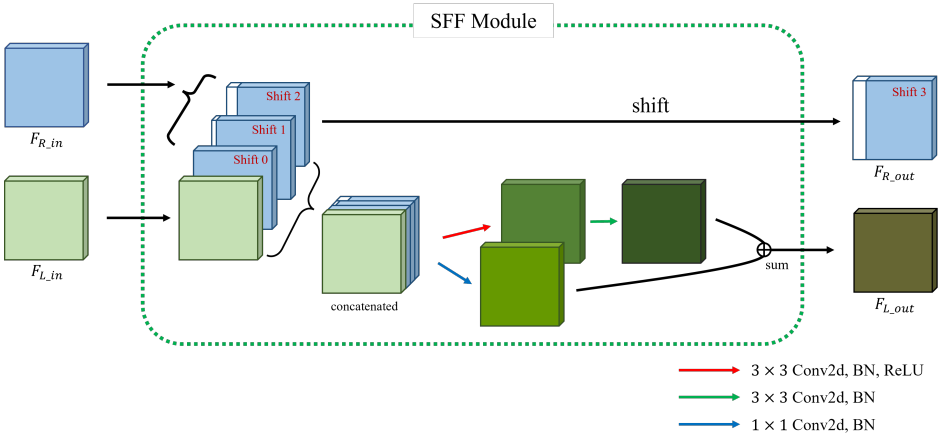

Figure 4. Details of a single SFF module in the S_1.

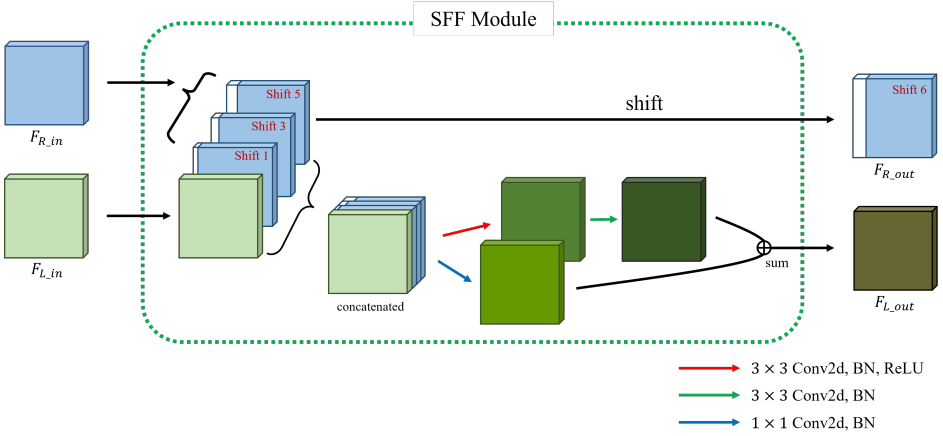

Figure 5. Details of a single SFF module in the S_2 and S_3.

Next, the output feature map $F^l_{(n,s)}$ of scale s in the nth MSFF module is obtained by fusing the intermediate feature maps $\{\tilde{F}^l_{(n,i)}\}^3_{i=1}$ with the cross-scale fusion function $f(\cdot)$, which is defined as

$$F^l_{(n,s)} = \sum_{i=1}^{3} f^s_i(\tilde{F}^l_{(n,i)}), \tag{1}$$

where $f^s_i(\cdot)$ is a function for the adaptive fusion of multi-scale features, similar to [24], that is defined as

$$f^s_i = \begin{cases} \mathrm{I}, & i = s, \\ (s-i) \cdot \mathit{strided}\ 3 \times 3\ \mathit{convs}, & i > s, \\ \mathit{upsampling}\ 1 \times 1\ \mathit{conv}, & i < s, \end{cases} \tag{2}$$

where I is an identity function, $(s-i) \cdot \mathit{strided}\ 3 \times 3\ \mathit{convs}$ represents a series of $(s-i)$ numbers of 3×3 convolutions with stride 2 for $2^{(s-i)}$ times downsampling to make the scale consistent, and $\mathit{upsampling}\ 1 \times 1\ \mathit{conv}$ represents bilinear upsampling for scale consistency, then 1×1 convolution is followed. It is noteworthy that $f^s_i(\cdot)$ fuses the features from different scales to share their information. By fusing these feature maps, some information about absent disparity is compensated using those of other scales. That is, this fusion is conducted so that each feature map obtains global and robust semantic information which covers broader spatial and disparity ranges from features of smaller scale. Furthermore, precise and local information for the fine detail is compensated from features of larger scale.

Figure 6 illustrates the disparity range covered in the first MSFF module, where bins with white color represent absent disparity values that are not included in the feature maps. A disparity value d in $\tilde{F}^l_{(n,1)}$ corresponds to $\{2d, 2d+1\}$ for $\tilde{F}^l_{(n,2)}$ and corresponds to $\{4d, 4d+1, 4d+2, 4d+3\}$ for $\tilde{F}^l_{(n,3)}$ owing to scale difference. For the first MSFF module, the feature map from S_1 module includes the disparity range of $[0, 5]$ on the $1/16$ scale because each SFF module in S_1 covers three disparity values, and two SFF modules are serially connected. Similarly, S_2 covers $[0, 11]$ disparity range in the $1/8$ scale using two SFF modules without even disparity values. Similarly, S_3 covers the disparity range of $[0, 23]$ on the $1/4$ scale with only odd disparity values. As shown in Figure 6, after the cross-scale fusion process, there is not absent disparity value for each feature map $F^l_{(n,i)}$, which compensates for the lack of even values of disparity information by fusing information from different scales.

Note that the left output feature map $F^l_{(M,i)}$ of the last MSFF module contains the aggregated cost distribution through the property of the SFF module [17]. Thus, the output cost volume V_i is obtained using a series of two 3×3 convolutions from $F^l_{(M,i)}$ to render the channel dimension of V_i equal to the disparity range at that scale. Specifically, the channel number of the output cost volume V_1 is the same as the full disparity range for its own scale, whereas those of V_2 and V_3 are half of the disparity range at those scales.

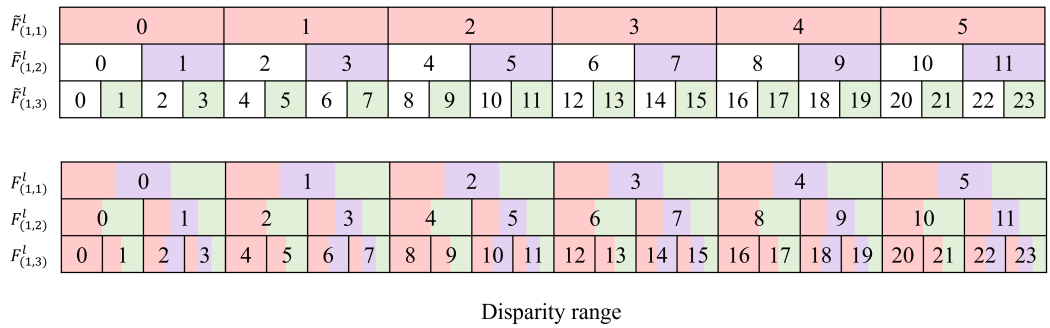

Figure 6. Details of multi-scale features before and after cross-scale fusion in the disparity domain.

3.3. Interlaced Cost Volume

We propose an interlaced concatenation method to combine the multi-scale cost volumes $\{V_i \in \mathbb{R}^{W_i \times H_i \times D_i}\}_{i=1}^{3}$ and produce the final cost volume V_f for disparity regression. Here, i represents the scale index, and W_i and H_i are $1/2^{(5-i)}$ times the width and height of the input image I_l, respectively. D_1, D_2, and D_3 are $1/16$, $0.5 \times (1/8)$, and $0.5 \times (1/4)$ times the maximum disparity search range, respectively. As shown in Figure 7, $\{V_i^{up} \in \mathbb{R}^{W_{i+1} \times H_{i+1} \times D_i}\}_{i=1}^{3}$ is obtained by upsampling the cost volume V_i only in the spatial domain to achieve the same spatial resolution as V_{i+1} while maintaining the channel dimension. Then, the upsampled V_i^{up} and V_{i+1} are interlaced along the channel direction to generate an interlaced cost volume IC_{i+1}, where the value of $IC_{i+1}(x, y, c)$ of (x, y) spatial position and c^{th} channel is defined by

$$IC_{i+1}(x, y, c) = \begin{cases} V_i^{up}(x, y, n), & c = 2n \\ V_{i+1}(x, y, n), & c = 2n + 1 \end{cases}, \tag{3}$$

where n denotes the channel index of V_i^{up} and V_{i+1}. When the interlaced concatenation is performed, the disparity channels from the lower scale are placed at the even number disparity. The final cost volume V_f, which is also IC_3, can be generated in the same manner using IC_2 and V_3.

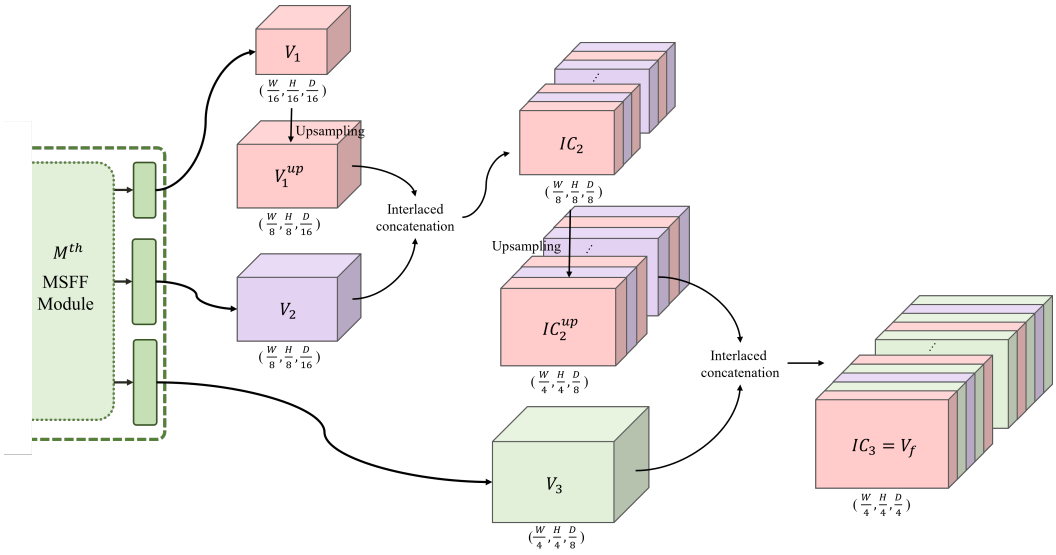

Figure 7. Process of interlaced concatenation.

3.4. Disparity Regression and Refinement

For each pixel, V_f contains a D_{max}-length vector that contains the matching costs of the disparity range D_{max}. This vector is converted to a probability vector using the softmax operation, where the probability $\hat{p}_d$ of disparity d is defined as

$$\hat{p}_d = \frac{\exp(c_d)}{\sum_{i=0}^{D_{\max}-1} \exp(c_i)}, \tag{4}$$

where c_d is the matching cost in V_f for disparity d. To estimate an initial disparity map $\hat{D}$ that contains a continuous disparity value for each pixel, the following softargmax function [11] is applied:

$$\hat{d} = \sum_{d=0}^{D_{max}-1} d \times \hat{p}_d, \tag{5}$$

where $\hat{p}_d$ is the probability corresponding to a candidate disparity d. The estimated disparity $\hat{d}$ is obtained using the weighted summation, as shown in Equation (5). This regression-based formulation can produce more robust disparity values with sub-pixel precision than classification-based stereo-matching methods [11].

To obtain a final disparity map D with increased accuracy, we use a refinement module for the initial disparity map $\hat{D}$ similar to [13]. In this module, the initial disparity map is upsampled to the same scale as I^l and concatenated with I^l as an input to the refinement module, where a series of convolution layers are followed to generate a final disparity map.

3.5. Loss Function

As shown in Figure 2, to train the network, we employ two loss functions: a disparity regression loss and an ACVF loss. The loss functions are described in detail in the following subsections.

3.5.1. Disparity Regression Loss

The first loss function is a disparity regression loss L_d, which is adopted in most networks [12,16,26] and is defined as

$$L_d = \frac{1}{N}\sum_{j=1}^{N} U_s(D^{gt}(j) - D(j)), \tag{6}$$

where $D^{gt}(j)$ and $D(j)$ are the ground truth disparity and the estimated disparity for jth pixel, respectively. N is the total number of pixels in D^{gt} and D. The smooth L_1 loss function U_s is widely used owing to its robustness and low sensitivity and is defined as

$$U_s(x) = \begin{cases} 0.5x^2, & if\ |x| \leqslant 1, \\ |x| - 0.5, & otherwise. \end{cases} \tag{7}$$

Note that L_d in Equation (6) is fully differentiable and useful for smooth disparity estimation.

3.5.2. Adaptive Cost-Volume-Filtering Loss

As shown in Figure 8, to directly supervise the estimated cost volume, we propose to filter the cost volume V_f using two types of cost volumes V_{gt}, generated using the ground truth disparity map, and V_t, generated from the teacher network with higher accuracy. The reason for using both types is that the ways cost volumes are generated are different, and the goal of usage is also different. V_{gt} is a unimodal distribution with a peak at the ground truth disparity value; thus, it can help filter the estimated cost volume for reducing the bias error aspect. In contrast, because V_{kd} is the cost volume estimated from the teacher network, it can be helpful in reducing variance errors [22].

Cost Volume Filtering Using Ground Truth Disparity

To filter the cost volume V_f using the ground truth disparity map, we generate a unimodal cost volume V_{gt} similar to [18]. Given a ground truth disparity map D^{gt}, we first downsample it four times to generate $\hat{D}^{gt}$ which has the same spatial resolution as V_f. Then, the unimodal distribution value $p_d^{gt}(j)$ for j^{th} pixel and disparity d is defined using the softmax operation as follows:

$$p_d^{gt}(j) = softmax(-\frac{|d - \hat{D}^{gt}(j)|}{\sigma}), \tag{8}$$

where $\hat{D}^{gt}(j)$ is the ground truth disparity of jth pixel, and σ is the variance related to the sharpness of the distribution. From Equation (8), the Laplacian unimodal distribution has a peak near the true disparity value.

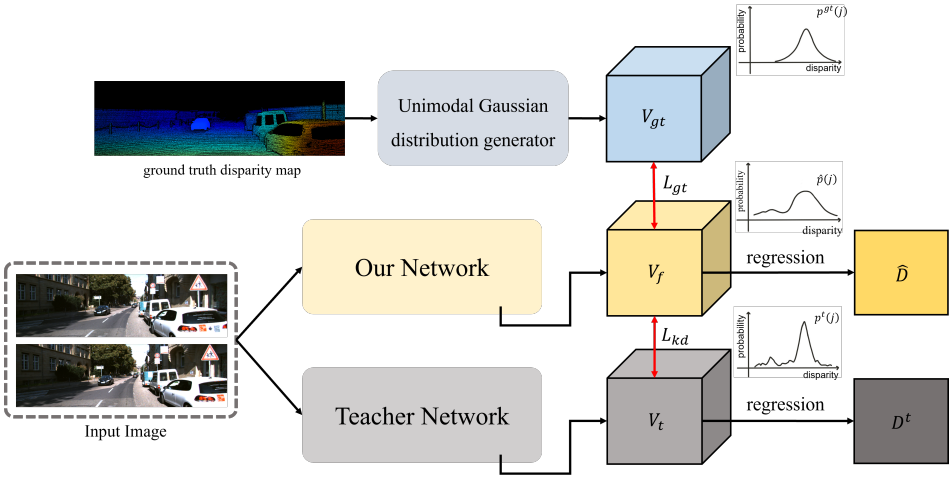

Figure 8. Process of adaptive cost volume filtering.

With the estimated probability distribution map $\hat{p}$ which is generated from the cost volume V_f on the softmax operation and the probability distribution p^{gt} generated from the ground truth disparity map, we define a unimodal cost-volume-filtering loss L_{gt} using the KL-divergence between them, as follows:

$$\begin{aligned} L_{gt} &= \frac{1}{N}\sum_{j=1}^{N} KL(p^{gt}(j), \hat{p}(j)) \\ &= \frac{1}{N}\sum_{j=1}^{N}\left(-\sum_{d=0}^{D_{max}-1} p_d^{gt}(j)\cdot\log\frac{\hat{p}_d(j)}{p_d^{gt}(j)}\right), \end{aligned} \tag{9}$$

where $p_d^{gt}(j)$ and $\hat{p}_d(j)$ are the probability value of p^{gt} and $\hat{p}$, respectively, for jth pixel and disparity d. N is the total number of pixels in p^{gt} and $\hat{p}$.

Cost Volume Filtering Using Knowledge Distillation

Using only the Laplacian unimodal, cost distribution from the ground truth disparity is not sufficient to effectively regularize the cost volume of the proposed network. For some pixels at the boundaries of objects, a multi-modal distribution would be more appropriate to help the network estimate accurate disparity [19]. However, the artificially generated distribution would not contain useful information for non-true disparities such as correlation among different disparities. Unlike [18,19] which uses artificially generated distributions, we perform knowledge distillation [21] to regularize the cost volume. The distribution generated from the teacher network with higher accuracy includes *dark knowledge* even for the *non-true disparities* which is helpful to further regularize the cost distributions for some pixels [20–23]. Thus, given a probability distribution map from the teacher network, the cost-volume-filtering loss L_{kd} using knowledge distillation via weighted KL-divergence is defined as follows:

$$\begin{aligned} L_{kd} &= \frac{1}{N}\sum_{j=1}^{N}\{\alpha(j)\cdot KL(p^t(j), \hat{p}(j))\} \\ &= \frac{1}{N}\sum_{j=1}^{N}\left\{\alpha(j)\cdot(-\sum_{d=0}^{D_{max}-1} p_d^t(j)\cdot\log\frac{\hat{p}_d(j)}{p_d^t(j)})\right\}, \end{aligned} \tag{10}$$

where $p_d^t(j)$ and $\hat{p}_d(j)$ are the probability values of the teacher network and the estimated cost volume at pixel j for disparity d, respectively. Here, $\alpha(j)$ is a weighting factor of pixel j that is defined similar to [22] as follows:

$$\begin{aligned} \alpha(j) &= (1 - \exp(-\frac{L^s(j)}{L^t(j)})), \\ L^s(j) &= \left|\hat{D}^{gt}(j) - \hat{D}(j)\right|, \\ L^t(j) &= \left|\hat{D}^{gt}(j) - \hat{D}^t(j)\right|, \end{aligned} \tag{11}$$

where $\hat{D}^t(j)$ and $\hat{D}(j)$ are disparity values for jth pixel of $\hat{D}^t$ and $\hat{D}$ obtained from V_t and V_f after performing the softargmax operation, respectively. The absolute differences $L^t(j)$ and $L^s(j)$ are calculated at pixel j with the ground truth disparity map $\hat{D}^{gt}$. The weight $\alpha(j)$ is determined according the ratio of the error of both networks. It is worthy to note that the role of this weighting factor is to avoid the negative effects of the distillation for pixels that the teacher network is less accurate than those of our network as student. At jth pixel, when the teacher network predicts a disparity with higher error than that of the proposed network, the weight $\alpha(j)$ decreases, and vice versa. Thus, the knowledge from the teacher network is adaptively transferred using the ratio between the errors of the teacher and our proposed networks for each pixel [22].

Now, the ACVF loss L_{cv} is defined by summing L_{gt} and L_{kd} as follows:

$$L_{cv} = \lambda_1 \cdot L_{gt} + \lambda_2 \cdot L_{kd}, \tag{12}$$

where λ_1 and λ_2 are the weighting factors for L_{gt} and L_{kd}, respectively. Note that L_{cv} is a combination of two cost-volume-filtering losses, L_{gt} and L_{kd}, where L_{gt} is the loss between V_{gt} and V_f, and L_{kd} is the loss between V_t and V_f. This L_{cv} loss is reminiscent of a loss function that uses the soft label with knowledge distillation scheme for the classification problem. Most classification methods usually combine a cross-entropy loss using a one-hot distribution with a ground truth label and a knowledge distillation loss using a teacher network to reduce both bias and variance errors [22]. Unlike in classification problems, to regress continuous disparity values, we use a unimodal distribution generated using ground truth disparity instead of a one-hot distribution. In addition, for each pixel, we adaptively combine the knowledge distillation loss based on the ratio between the accuracies of the teacher and student networks similar to [22].

3.5.3. Total Loss Function

To train the proposed network in an end-to-end manner, we employ two types of losses; the disparity regression loss described in Equation (6) and the ACVF loss in Equation (12). The final total loss L_{total} is a combination of these two losses, defined as

$$L_{total} = L_d + \beta \cdot L_{cv}, \tag{13}$$

where β is a hyperparameter used to balance the two different losses.

4. Experiments and Results

4.1. Datasets and Evaluation Metrics

To evaluate the performance of the proposed method, we conducted various comparative experiments using two stereo datasets, Scene Flow and KITTI 2015. In addition, ablation studies were conducted on the Scene Flow dataset with various settings.

1. **Scene Flow** [10]: A large-scale synthetic dataset with dense ground truth disparity maps. It contains 35,454 training image pairs and 4370 test image pairs to train the network directly. The end point error (EPE), which is the average disparity error in pixels of the estimated disparity map, was used as an evaluation metric.
2. **KITTI**: KITTI 2015 [25] and KITTI 2012 [29] are real-world datasets with outdoor views captured from a driving car. KITTI 2015 contains 200 training stereo image pairs with sparse ground truth disparity maps and 200 image pairs for testing. KITTI 2012 contains 194 training image pairs with sparse ground truth disparity maps

and 195 testing image pairs. The 3-Pixel-Error (3PE), the percentage of pixels with a disparity error larger than three pixels, was used as an evaluation metric.

4.2. Implementation Details

We trained the network using the PyTorch [36] framework in an end-to-end manner, with Adam [37] ($\beta_1 = 0.9$, $\beta_2 = 0.999$) as the optimizer. For the Scene Flow dataset, we trained the network with a batch size of 12 using all training sets (35,454 image pairs) from scratch on an NVIDIA GeForce RTX GPU for 120 epochs, and evaluated it using the test set (4370 image pairs). For the preprocessing step, the raw input images were randomly cropped to a size of 256 × 512 and normalized using ImageNet [38] statistics (mean:[0.485, 0.456, 0.406]; std:[0.229, 0.224, 0.225]). The learning rate started at 0.001 and decreased by half at the 20th, 40th, and 60th epochs. We used this dataset for ablation and pretraining for KITTI. For the KITTI dataset, we first fine-tuned the pretrained model for Scene Flow on the mixed KITTI 2012 and 2015 training sets for 1000 epochs. Then, an additional 1000 epochs were fine-tuned on the KITTI 2015 training set. For the mixed KITTI 2012 and KITTI 2015 training sets, input images were randomly cropped to a size of 336 × 960; the initial learning rate was 0.0001 and then decreased by half at the 200th, 400th, and 600th epochs. The same normalization method as the Scene Flow dataset was used for preprocessing. The loss weights for Equation (12) were set to $\lambda_1 = 5$ and $\lambda_2 = 1$, and β value for Equation (13) was set as $\beta = 1$. The maximum disparity D_{max} of the original input image size was set to 192. The number M of MSFF modules in the MSFFNet are set as $M = 2$ which covers the full disparity range of 192 for 1/4 of the input image size, because a single MSFF module covers a disparity range of 24. For the teacher network, GwcNet-g [15] was used, and the cost volume V_t was obtained from the teacher network before performing the softargmax regression for the resultant disparity map.

4.3. Comparative Result

4.3.1. Scene Flow Dataset

We compared the performance of our proposed method with that of other stereo-matching networks on the Scene Flow test set. Table 2 presents the comparative results of our network and various other 2D/3D convolution-based networks. "Ours" means the EPE of the predicted disparity map of the proposed network. As shown in Table 2, our MSFFNet outperforms all other 2D convolution-based models [10,17,27,28] and some 3D convolution-based models [12,13]. Among the 2D convolution networks, compared with SFFNet [17] which proposed the SFF module, the proposed MSFFNet shows better accuracy with $|1.04 - 1.01|/1.04 = 2.9\%$ lower EPE. Furthermore, among the 3D convolution networks, MSFFNet results in $|1.09 - 1.01|/1.09 = 7.3\%$ better accuracy compared with PSMNet [12] and $|1.10 - 1.01|/1.10 = 8.2\%$ better accuracy compared with StereoNet [13].

Table 2. Comparison of EPEs of various methods on the Scene Flow test set.

3D Conv. Models						**2D Conv. Models**				
PSMNet [12]	StereoNet [13]	Gwc-Net [15]	DeepPruner-Fast [26]	AANet [16]	AcfNet [18]	DispNet-C [10]	CRL [27]	SFFNet [17]	2D-MobileStereoNet [28]	Ours
1.09	1.10	0.79	0.97	0.87	0.92	1.68	1.32	1.04	1.14	1.01

Figure 9 shows a qualitative comparison between the MSFFNet and other networks [12,17,26,28] on the Scene Flow test set. This comparison demonstrates that our network predicts disparity maps comparable to other 2D/3D convolution networks for most regions.

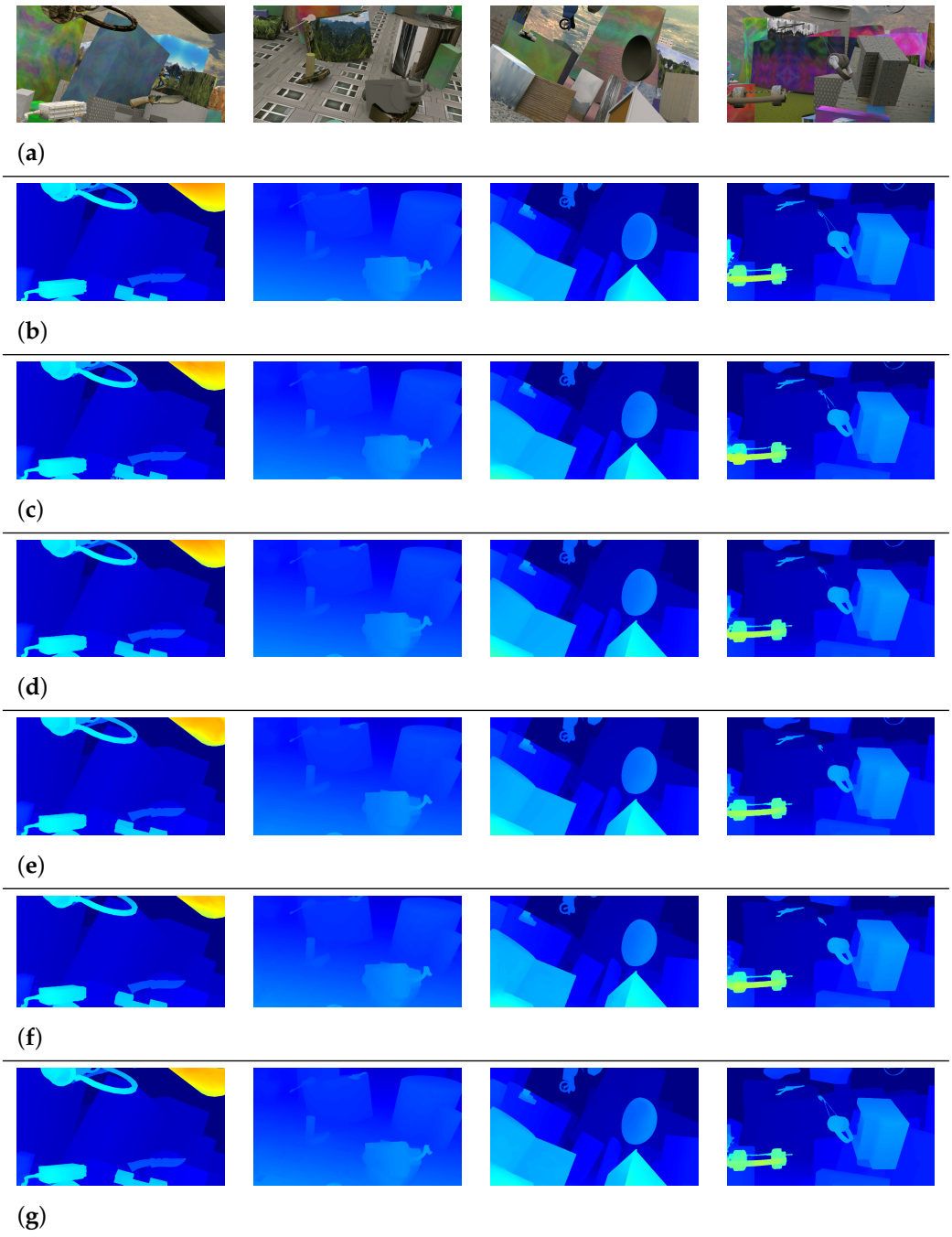

Figure 9. Qualitative comparison on the Scene Flow test set. (**a**) Input Image. (**b**) Ground-truth image. (**c**) PSMNet [12]. (**d**) DeepPruner-Fast [26]. (**e**) SFFNet [17]. (**f**) 2D-MobileStereoNet [28]. (**g**) Ours.

4.3.2. KITTI-2015 Dataset

We compared the performance of our method with other methods on the KITTI 2015 stereo benchmark [25]. Table 3 shows the comparison of various indicators including runtime, 3PE, number of parameters, and FLOPs. In this table, *bg*, *fg* and *all* indicate the percentage of error pixels averaged over the background pixels, foreground pixels, and all ground truth pixels, respectively. In addition, *Noc (%)* and *All (%)* represent the percentages of error pixels for non-occluded regions and all pixels, respectively. As shown in Table 3, our method achieves faster runtime and requires smaller number of parameters and smaller FLOPs with comparable accuracy compared with most 3D convolution-based networks. Specifically, compared with AANet [16] which connects multi-scale networks in parallel, our method has better efficiency with $|62 - 42|/62 = 32.3\%$ faster runtime

and $|3.9 - 2.92|/3.9 = 25.1\%$ smaller number of parameters. In addition, compared with AcfNet [18] which directly supervises the predicted cost volume, our method has better efficiency with $|480 - 42|/480 = 91.2\%$ faster runtime and $|5.36 - 2.92|/5.36 = 45.5\%$ smaller number of parameters. Table 3 shows that the proposed MSFFNet is the fastest network using the smallest FLOPs with comparable accuracy among the 2D convolution-based methods. Especially, compared with SFFNet [17], our method has better performance in all indicators with $|3.28 - 2.94|/3.28 = 10.4\%$ better accuracy, $|4.61 - 2.92|/4.61 = 36.7\%$ smaller number of parameters, and $|208.21 - 97.96|/208.21 = 52.9\%$ smaller FLOPs.

Figure 10 illustrates some of the qualitative comparison results of the MSFFNet and other networks [12,17,26,28] on the KITTI 2015 benchmark [25]. All figures are reported from the KITTI 2015 evaluation server. The images for each network represent the predicted disparity maps and error maps of the predicted disparity maps for the left input images. In the error maps, a pixel color closer to red indicates a region with a larger error. From this comparison, it is observed that our method generates comparable disparity maps with other methods for various scenes.

Table 3. Quantitative comparison results on the KITTI 2015 test set.

Method	Network	Runtime	Noc (%)			All (%)			Params	FLOPs
			bg	fg	All	bg	fg	All		
3D conv. method	Content-CNN [9]	1000 ms	3.32	7.44	4.00	3.73	8.58	4.54	0.70 M	978.19 G
	MC-CNN [8]	67,000 ms	2.48	7.64	3.33	2.89	8.88	3.89	0.15 M	526.28 G
	GC-Net [11]	900 ms	2.02	3.12	2.45	2.21	6.16	2.87	2.86 M	2510.96 G
	PSMNet [12]	410 ms	1.71	4.31	2.14	1.86	4.62	2.32	5.36 M	761.57 G
	StereoNet [13]	15 ms	-	-	-	4.30	7.45	4.83	0.31 M	-
	GwcNet-g [15]	320 ms	1.61	3.49	1.92	1.74	3.93	2.11	6.43 M	-
	GANet-15 [39]	1500 ms	1.40	3.37	1.73	1.55	3.82	1.93	-	-
	DeepPruner-Best [26]	182 ms	1.71	3.18	1.95	1.87	3.56	2.15	7.39 M	383.49 G
	DeepPruner-Fast [26]	64 ms	2.13	3.43	2.35	2.32	3.91	2.59	7.47 M	153.77 G
	AANet [16]	62 ms	1.80	4.93	2.32	1.99	5.39	2.55	3.9 M	-
	AcfNet [18]	480 ms	1.36	3.49	1.72	1.51	3.80	1.89	5.36 M	-
2D conv. method	DispNetC [10]	60 ms	4.11	3.72	4.05	4.32	4.41	4.34	42.43 M	93.46 G
	CRL [27]	470 ms	2.32	3.68	2.38	2.48	3.59	2.67	78.21 M	185.85 G
	SFFNet [17]	76 ms	2.50	5.44	2.99	2.69	6.23	3.28	4.61 M	208.21 G
	2D-MobileStereoNet [28]	400 ms	2.29	3.81	2.54	2.49	4.53	2.83	2.23 M	125.58 G
	Ours	42 ms	2.35	4.58	2.72	2.53	4. 99	2.94	2.92 M	97.96 G

4.4. Ablation Study

As shown in Table 4, we conducted various experiments with a controlled setup to verify the effectiveness of our proposed method on the Scene Flow dataset. First, we checked the effectiveness of the interlaced concatenation method in our network and then verified the effectiveness of the ACVF loss. All the methods list in Table 4 included the disparity regression loss L_d in Equation (6) for training networks as a baseline loss function.

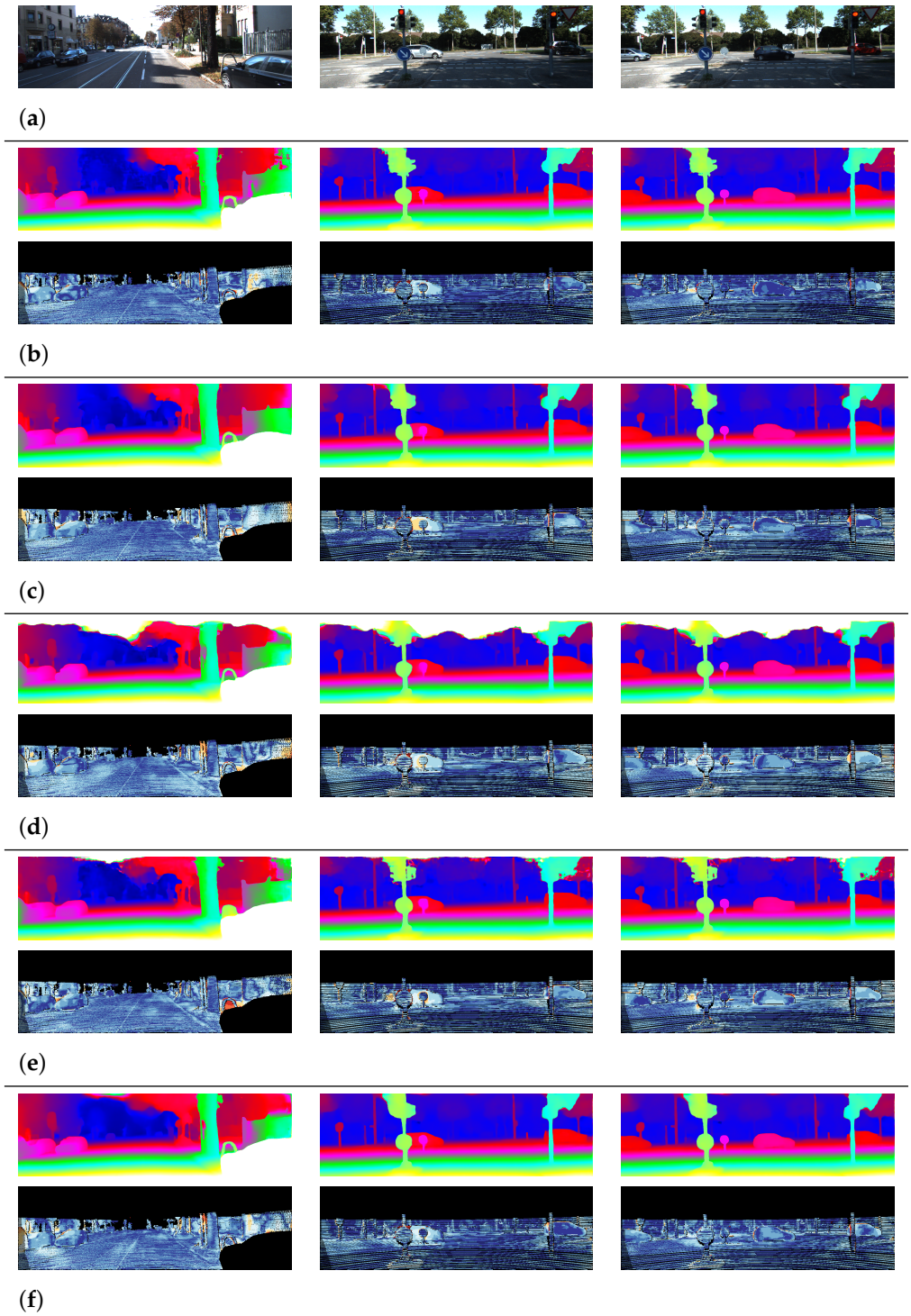

Figure 10. Qualitative comparison on the KITTI-2015 test set. (**a**) Input Image. (**b**) PSMNet [12]. (**c**) DeepPruner-Fast [26]. (**d**) SFFNet [17]. (**e**) 2D-MobileStereoNet [28]. (**f**) Ours.

Table 4. Ablation study results of the network architecture and the adaptive cost-volume-filtering loss on the Scene Flow test set.

Network Architecture		Adaptive Cost Volume Filtering		EPE
Concatenation	**Summation**	$+L_{gt}$	$+L_{kd}$	
	✓			1.174
✓				1.098
✓		✓		1.04
✓			✓	1.043
✓		✓	✓	1.011

4.4.1. Effect of Interlaced Cost Concatenation

To investigate the effectiveness of the proposed interlaced concatenation method defined in Equation (3), we conducted two experiments with the interlaced concatenation method denoted as "Concatenation" and the simple summation method denoted as "Summation" on the Scene Flow dataset, as listed in Table 4. Figure 11 compares the details of these two methods. The proposed concatenation method upsamples the cost volume V_i to obtain V_i^{up} only in the spatial domain without changing the channel dimension. Subsequently, with the upsampled V_i^{up} and V_{i+1}, interlaced concatenation is performed using Equation (3) to generate the fused cost volume IC_{i+1}. However, the naive summation method upsamples the cost volume V_i in both the spatial and disparity domains to produce $\hat{V}_i^{up}$ which has the same size as the cost volume V_{i+1}. Then, IS_{i+1} is obtained by simple summation defined as

$$IS_{i+1}(x,y,c) = \begin{cases} \hat{V}_i^{up}(x,y,c), & c = 2n \\ \hat{V}_i^{up}(x,y,c), + V_{i+1}(x,y,n), & c = 2n+1 \end{cases}, \tag{14}$$

where values for odd disparities in IS_{i+1} are computed by summing the elements of V_{i+1} and those of $\hat{V}_i^{up}$ for odd disparities. The cost values for even disparities of IS_{i+1} are established using the interpolated cost volume $\hat{V}_i^{up}$. The "Summation" method uses IS_{i+1} instead of IC_{i+1} for building the cost volumes.

Table 4 shows that the proposed concatenation method achieves a lower EPE (1.098) than the EPE (1.174) of the simple summation method on the Scene Flow test set. The interpolation of V_i along the disparity channel smoothens the cost of each odd disparity, which disturbs the cost of V_i through a summation. Thus, the proposed concatenation method which maintains the disparity channels of each cost volume generates more accurate results.

4.4.2. Effect of Adaptive Cost-Volume-Filtering Loss

The proposed ACVF loss in Equation (12) consists of two loss components such as L_{gt} using the ground truth disparity map and L_{kd} using the teacher network. To verify the effect of each component, we conducted experiments in which we added each loss L_{gt} and L_{kd} one at a time to the ACVF loss. Using the baseline method where MSFFNet is trained with only disparity regression loss L_d in Equation (6), we gradually added only one component of the ACVF loss and then finally added both of them. As shown in Table 4, the addition of each component improves the EPE for the Scene Flow test set compared with the baseline method. Because L_{gt} adds a constraint to the predicted cost volume in terms of reducing bias errors, the EPE of using only L_{gt} is reduced by 0.058. Meanwhile, L_{kd} is helpful for reducing variance errors, EPE is reduced by 0.055. The adaptive addition of both components achieves the best EPE of 1.011 which is reduced by 0.087 compared with the baseline method.

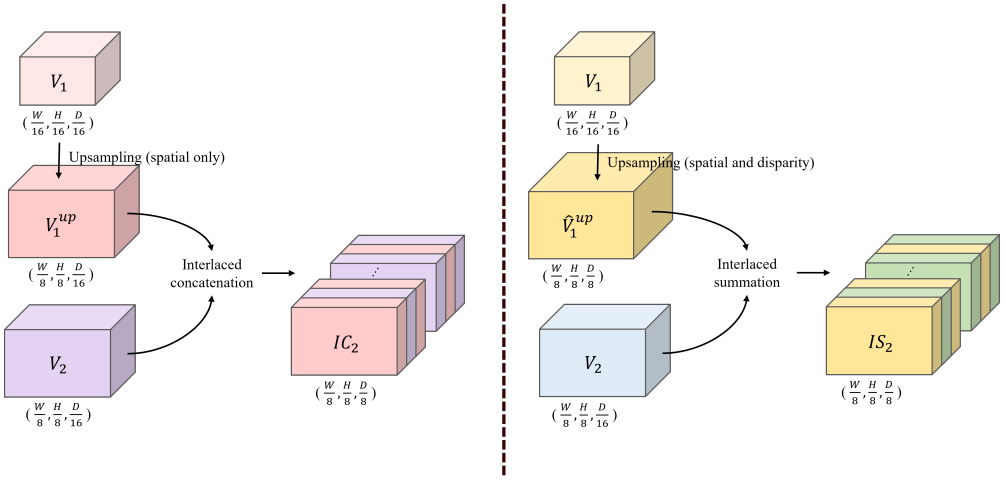

Figure 11. Details of interlaced concatenation and interlaced summation.

5. Conclusions

In this paper, we presented an efficient MSFFNet for stereo matching. The network consists of a series of MSFF modules that efficiently connect multi-scale SFF modules in parallel to generate multi-scale cost volumes. In addition, an interlaced concatenation method is presented to generate a final cost volume which contains the full disparity range by combining the multi-scale cost volumes that include information for only parts of the disparity range. To train the network, an ACVF loss function was proposed to regularize the estimated cost volume using both the ground truth disparity map and the probability distribution map generated from the teacher network with higher accuracy. This loss function adaptively utilizes them based on the error ratio of disparity maps generated from the teacher network and the proposed network. Results of various comparative experiments show that the proposed method is more efficient than previous methods because our network consumes fewer parameters and runs at a faster speed with reasonable accuracy. For example, our network achieves 1.01 EPE with 42 ms runtime, 2.92M parameters and 97.96G FLOPs on the Scene Flow test set. It is 89% faster and 7% more accurate with 45% fewer parameters than PSMNet [12] which is one of the representative 3D convolution-based methods on the Scene Flow test set. Compared with SFFNet [17] which is one of the representative 2D convolution-based methods, our method has better performance with 44% faster runtime, 10% better accuracy, and 36% fewer parameters.

Despite the performance improvements of various aspects, the proposed network has a limitation in that the accuracy of our method is relatively lower than some of heavy 3D convolution-based networks. In future work, the accuracy of our method should be improved by employing a more accurate teacher model to train our network. We believe that our method sheds light on developing a practical stereo-matching network that generates accurate depth information in real time. In addition, our method can be adopted in a variety of applications that require real-time 3D depth information.

Author Contributions: Conceptualization, S.J. and Y.S.H.; software, S.J.; validation, Y.S.H.; investigation, S.J.; writing—original draft preparation, S.J.; writing—review and editing, Y.S.H.; supervision, Y.S.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been supported in part by the BK21 FOUR program of the National Research Foundation of Korea funded by the Ministry of Education(NRF5199991014091) and in part by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(2022R1F1A1065702).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the anonymous reviewers for their constructive comments and recommendations.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolutional Neural Network
SFF module	Sequential Feature Fusion module
SFFNet	Sequential Feature Fusion Network
MSFF module	Multi-scale Sequential Feature Fusion module
MSFFNet	Multi-scale Sequential Feature Fusion Network
ACVF	Adaptive Cost Volume Filtering
EPE	End Point Error
3PE	3-Pixel Error
FLOPs	FLoating point Operations Per Second

References

1. Scharstein, D.; Szeliski, R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vis.* **2002**, *47*, 7–42. [CrossRef]
2. Huang, J.; Tang, S.; Liu, Q.; Tong, M. Stereo matching algorithm for autonomous positioning of underground mine robots. In Proceedings of the International Conference on Robots & Intelligent System, Changsha, China, 26–27 May 2018; pp. 40–43.
3. Zenati, N.; Zerhouni, N. Dense stereo matching with application to augmented reality. In Proceedings of the IEEE International Conference on Signal Processing and Communications, Dubai, United Arab Emirates, 24–27 November 2007; pp. 1503–1506.
4. El Jamiy, F.; Marsh, R. Distance estimation in virtual reality and augmented reality: A survey. In Proceedings of the IEEE International Conference on Electro Information Technology, Brookings, SD, USA, 20–22 May 2019; pp. 063–068.
5. Chen, C.; Seff, A.; Kornhauser, A.; Xiao, J. Deepdriving: Learning affordance for direct perception in autonomous driving. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 2722–2730.
6. Wong, A.; Mundhra, M.; Soatto, S. Stereopagnosia: Fooling stereo networks with adversarial perturbations. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; Volume 35, pp. 2879–2888.
7. Kwon, H.; Kim, Y. BlindNet backdoor: Attack on deep neural network using blind watermark. *Multimed. Tools Appl.* **2022**, *81*, 6217–6234. [CrossRef]
8. Zbontar, J.; LeCun, Y. Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches. *J. Mach. Learn. Res.* **2016**, *17*, 2287–2318.
9. Luo, W.; Schwing, A.G.; Urtasun, R. Efficient deep learning for stereo matching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 5695–5703.
10. Mayer, N.; Ilg, E.; Hausser, P.; Fischer, P.; Cremers, D.; Dosovitskiy, A.; Brox, T. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4040–4048.
11. Kendall, A.; Martirosyan, H.; Dasgupta, S.; Henry, P.; Kennedy, R.; Bachrach, A.; Bry, A. End-to-end learning of geometry and context for deep stereo regression. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 66–75.
12. Chang, J.R.; Chen, Y.S. Pyramid stereo-matching network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 5410–5418.
13. Khamis, S.; Fanello, S.; Rhemann, C.; Kowdle, A.; Valentin, J.; Izadi, S. Stereonet: Guided hierarchical refinement for real-time edge-aware depth prediction. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 573–590.
14. Wang, Y.; Lai, Z.; Huang, G.; Wang, B.H.; Van Der Maaten, L.; Campbell, M.; Weinberger, K.Q. Anytime stereo image depth estimation on mobile devices. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 5893–5900.
15. Guo, X.; Yang, K.; Yang, W.; Wang, X.; Li, H. Group-wise correlation stereo network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3273–3282.
16. Xu, H.; Zhang, J. Aanet: Adaptive aggregation network for efficient stereo matching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 1959–1968.

17. Jeong, J.; Jeon, S.; Heo, Y.S. An Efficient stereo-matching network Using Sequential Feature Fusion. *Electronics* **2021**, *10*, 1045. [CrossRef]
18. Zhang, Y.; Chen, Y.; Bai, X.; Yu, S.; Yu, K.; Li, Z.; Yang, K. Adaptive unimodal cost volume filtering for deep stereo matching. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 12926–12934.
19. Garg, D.; Wang, Y.; Hariharan, B.; Campbell, M.; Weinberger, K.Q.; Chao, W.L. Wasserstein distances for stereo disparity estimation. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 22517–22529.
20. Yuan, L.; Tay, F.E.; Li, G.; Wang, T.; Feng, J. Revisiting Knowledge Distillation via Label Smoothing Regularization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 3903–3911.
21. Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *arXiv* **2015**, arXiv:1503.02531.
22. Zhou, H.; Song, L.; Chen, J.; Zhou, Y.; Wang, G.; Yuan, J.; Zhang, Q. Rethinking soft labels for knowledge distillation: A bias-variance tradeoff perspective. *arXiv* **2021**, arXiv:2102.00650.
23. Zhao, B.; Cui, Q.; Song, R.; Qiu, Y.; Liang, J. Decoupled Knowledge Distillation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 11953–11962.
24. Sun, K.; Xiao, B.; Liu, D.; Wang, J. Deep high-resolution representation learning for human pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5693–5703.
25. Menze, M.; Geiger, A. Object scene flow for autonomous vehicles. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3061–3070.
26. Duggal, S.; Wang, S.; Ma, W.C.; Hu, R.; Urtasun, R. Deeppruner: Learning efficient stereo matching via differentiable patchmatch. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 4384–4393.
27. Pang, J.; Sun, W.; Ren, J.S.; Yang, C.; Yan, Q. Cascade Residual Learning: A Two-Stage Convolutional Neural Network for Stereo Matching. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Venice, Italy, 22–29 October 2017; pp. 887–895.
28. Shamsafar, F.; Woerz, S.; Rahim, R.; Zell, A. MobileStereoNet: Towards Lightweight Deep Networks for Stereo Matching. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–8 January 2022; pp. 2417–2426.
29. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for autonomous driving? the kitti vision benchmark suite. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 3354–3361.
30. Shen, Z.; Dai, Y.; Rao, Z. Msmd-net: Deep stereo matching with multi-scale and multi-dimension cost volume. *arXiv* **2020**, arXiv:2006.12797.
31. Gao, Q.; Zhou, Y.; Li, G.; Tong, T. Compact StereoNet: Stereo Disparity Estimation via Knowledge Distillation and Compact Feature Extractor. *IEEE Access* **2020**, *8*, 192141–192154. [CrossRef]
32. Mao, Y.; Liu, Z.; Li, W.; Dai, Y.; Wang, Q.; Kim, Y.T.; Lee, H.S. UASNet: Uncertainty Adaptive Sampling Network for Deep Stereo Matching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 6311–6319.
33. Liang, Z.; Feng, Y.; Guo, Y.; Liu, H.; Chen, W.; Qiao, L.; Zhou, L.; Zhang, J. Learning for disparity estimation through feature constancy. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 2811–2820.
34. Yang, G.; Zhao, H.; Shi, J.; Deng, Z.; Jia, J. Segstereo: Exploiting semantic information for disparity estimation. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 636–651.
35. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
36. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 8026–8037.
37. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the International Conference on Learning Representations (Poster), San Diego, CA, USA, 7–9 May 2015.
38. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
39. Zhang, F.; Prisacariu, V.; Yang, R.; Torr, P.H. Ga-net: Guided aggregation net for end-to-end stereo matching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 185–194.