Domain-invariant Stereo Matching Networks

Feihu Zhang[1], Xiaojuan Qi[2], Ruigang Yang[3], Victor Prisacariu[1]
Benjamin Wah[4], Philip Torr[1]

[1] University of Oxford
[2] University of Hong Kong
[3] Baidu Research
[4] Chinese University of Hong Kong

Abstract. State-of-the-art stereo matching networks have difficulties in generalizing to new unseen environments due to significant domain differences, such as color, illumination, contrast, and texture. In this paper, we aim at designing a domain-invariant stereo matching network (DSMNet) that generalizes well to unseen scenes. To achieve this goal, we propose i) a novel "domain normalization" approach that regularizes the distribution of learned representations to allow them to be invariant to domain differences, and ii) an end-to-end trainable structure-preserving graph-based filter for extracting robust structural and geometric representations that can further enhance domain-invariant generalizations. When trained on synthetic data and generalized to real test sets, our model performs significantly better than all state-of-the-art models. It even outperforms some deep neural network models (*e.g.* MC-CNN and DispNet) fine-tuned with test-domain data. The code is available at https://github.com/feihuzhang/DSMNet.

1 Introduction

Stereo reconstruction is a fundamental problem in computer vision, robotics and autonomous driving. It aims to estimate 3D geometry by computing disparities between matching pixels in a stereo image pair. Recently, many end-to-end deep neural network models (*e.g.* [5,19,63]) have been developed for stereo matching that achieve impressive accuracy on several datasets or benchmarks.

However, state-of-the-art stereo matching networks (supervised [5,19,63] and unsupervised [51,68]) cannot generalize well to unseen data without fine-tuning or adaptation. Their difficulties lie in the large domain differences (such as color, illumination, contrast and texture). As illustrated in Fig. 1, the pre-trained models on one specific dataset produce poor results on other real and unseen scenes.

Domain adaptation and transfer learning methods (*e.g.* [3,12,51]) attempt to transfer or adapt from one source domain to another new domain. Typically, a large number of stereo images from the new domain are required for the adaptation. However, these cannot be easily obtained in many real scenarios. Yet, we still need a good method for disparity estimation even without data from the new domain for adaptation.

We focus on the more challenging but crucial domain generalization [1] problem that assumes no access to target information for adaptation or fine-tuning. Namely, we are trying to design a model that can generalize well to unseen data without any re-training or adaptation. The difficulties in developing such

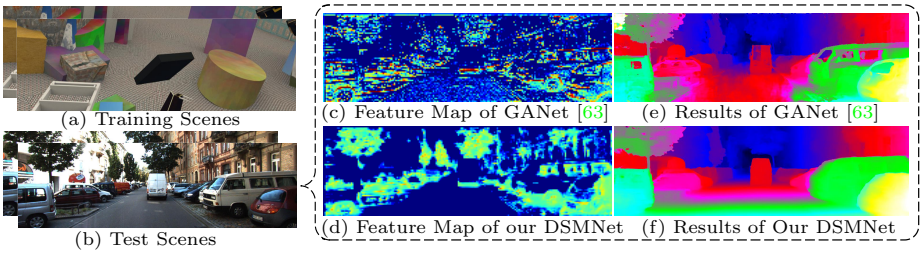


Fig. 1: Visualization of the feature maps and disparity results. GANet [63] is used for comparisons. The features used for matching (outputs of the feature extraction networks) are visualized in (c) and (d). Models are trained on synthetic data (Sceneflow [32]) and tested on novel real scenes (KITTI [33]). The feature maps from GANet has many artifacts (*i.e.* noise). Our DSMNet mainly captures the structure and shape information as robust features, and there is no distortions or artifacts in the feature map. It can produce accurate disparity estimations in the novel test scenes.

a domain-invariant stereo matching network (DSMNet) come from the significant domain differences (Fig. 1(a)-(b)) which can be roughly categorized as i) image-level styles (*e.g.* color, illumination), ii) local variations (*e.g.* contrast), iii) texture patterns, details and noise conditions and iv) other complicated domain shifts (*e.g.* uncommon/non-linear contents). They can be approximated by:

$$f(p) = \alpha_I(\alpha_p \cdot \phi(p) + \beta_p) + \beta_I. \quad (1)$$

Here, p is the feature of each pixel (e.g. RGB). Without domain shifts, $f(p) = p$ for different datasets. In practice, domain shifts are varying in different datasets. The i) image-level style differences can be represented as α_I and β_I. The ii) local variations (e.g. contrasts) are α_p. β_p represents the iii) image details/noise. Pixels of an image have the same α_I and β_I. The local shifts α_p and β_p are varying in different regions/pixels. And ϕ is the expression of iv) other uncommon domain differences that cannot be easily formulated as specific models.

Fig. 1 visualizes the features learned by state-of-the-art stereo matching model [63]. Such domain differences make the learned features unstable, distorted and noisy, leading to many wrong matching results (Fig. 1(e)) when applied to the novel test data (Fig. 1(c)).

In this paper, we propose two novel trainable neural network layers for constructing the DSMNet for cross-domain generalization without fine-tuning or adaptation. The proposed novel **domain normalization (DN)** layer fully regulates the distribution of the feature in both the image-level spatial (height and width) and the pixel-level channel dimensions. It can therefore reduce the domain shifts/differences of i) image-level styles (α_I and β_I in Eq. (1)) and ii) local contrast variations (α_p in Eq. (1)) between different datasets/scenes. Our non-local **structure-preserving graph-based filtering (SGF)** layer can further smooth and reduce the iii) domain-sensitive local details/noise (β_p in Eq.(1)). It also helps capture more robust structural and geometric representations (*e.g.* shape and structure, as in Fig. 1(d)) that are more robust to iv) many other complicated domain differences (ϕ in Eq. (1)) for stereo reconstruction.

We formulate our method as an end-to-end deep neural network and train it only with synthetic data. In experiments, without any fine-tuning or adaptation on the real test data, our DSMNet far outperforms: 1) almost all state-of-the-art stereo matching models (*e.g.* GANet [63]) trained on the same synthetic dataset, 2) most of the traditional methods (*e.g.* Cosfter filter, SGM [14] *et al.*), 3) most of the unsupervised/self-supervised models trained on the target test domains. Our model even surpasses some of the fine-tuned (on the target domains) supervised neural network models (*e.g.* MC-CNN [61], content-CNN [31], DispNetC [32]). Also, it doesn't sacrifice fine-tuned accuracy for generalization. After fine-tuning on the target scenes, it can achieve state-of-the-art accuracy (*e.g.* on KITTI benchmark). Moreover, our method can be easily extended to the optical flow task. It also significantly improves the generalization abilities of the optical flow networks (*e.g.* FlowNew2 [17], PwcNet [48]).

2 Related Work

2.1 Deep Neural Networks for Stereo Matching

In recent years, deep neural networks have seen great success in stereo matching [5, 19, 32, 44, 63]. These models can be categorized into three types: 1) learning better features for traditional stereo matching algorithms, 2) correlation-based deep neural networks, 3) cost-volume based stereo matching networks.

In the first category, deep neural networks have been used to compute patch-wise similarity scores as the matching costs [61, 64]. The costs are then fed into the traditional cost aggregation and disparity computation/refinement methods [14] to get the final disparity maps. The models are, however, limited by the traditional matching cost aggregation step and often produce wrong predictions in occluded regions, large textureless/reflective regions and around object edges.

DispNetC [32], a typical method in the second category, computes the correlations by warping between stereo views and attempts to predict the per-pixel disparity by minimizing a regression training loss. Many other sate-of-the-art methods, including iResNet [28], CRL [38], SegStereo [57], EdgeStereo [47], HD^3 [60], and MADNet [51], are all based on color or feature correlations between the left and right views for disparity estimation.

The recently developed cost-volume based models explicitly learn feature extraction, cost volume, and regularization function all end to end. Examples include GC-Net [19], PSM-Net [5], StereoNet [20], AnyNet [55], GANet [63] and EMCUA [36]. They all utilize a similarity cost as the third dimension to build the 4D cost volume in which the real geometric context is maintained.

Others, like [13], combine the correlation and cost volume strategies.

The common feature of these models is that they all require a large number of training samples with ground truths. More importantly, a model trained on one domain cannot generalize well to new scenes without fine-tuning or retraining.

2.2 Adaptation and Self-supervised Learning

Self-supervised Learning: A recent trend of training stereo matching networks in an unsupervised manner relies on image reconstruction losses that are achieved

by warping left and right views [67,68]. However, they cannot solve the occlusions and reflective regions where there is no correspondence between the left and the right views. Also, they cannot generalize well to other new domains.

Domain Adaptation: Some methods pre-train the models on synthetic data and then explore the cross-domain knowledge to adapt [12, 39] for a new domain. Others focus on the online or offline adaptations [41, 49–51]. For example, MAD-Net [51] is proposed to adapt the pre-trained model online and in real time. But, it has poor accuracy even after the adaptation. Moreover, the domain adaptation approaches require a large number of stereo images from the target domain for adaptations. However, these cannot be easily obtained in many real scenarios. And, in this case, we still need a good method for disparity estimation even without data from the new domain for adaptation.

2.3 Cross-domain Generalization

In contrast to domain adaptation, domain generalization [1, 11] is a much harder problem that assumes no access to target information for adaptation or fine-tuning. There are many approaches that explore the idea of domain-invariant feature learning. Previous approaches focus on developing data-driven strategies to learn invariant features from different source domains [11, 22, 34]. Some recent methods utilize meta-learning that takes variations in multiple source domains to generalize to novel test distributions [1, 23]. Other approaches [24, 25] employ an invariant adversarial network to learn domain-invariant representations for image recognition. Choy *et al*. [7] develop a universal feature learning framework for visual correspondences using deep metric learning.

In contrast to the above approaches, there are methods that try to improve the batch or instance normalization in order to improve the generalization and robustness for style transfer or image recognition [35, 37].

In summary, for stereo matching, work is seldom done to improve the generalization ability of the end-to-end deep neural network models, especially when developing the domain-invariant stereo matching networks.

3 Proposed DSMNet

To address the challenges of domain shifts (Eq.(1)), we propose 1) a novel domain normalization (DN) to remove the influence of the image-level domain shifts (α_I and β_I: *e.g.* color, style, illuminance) and the local contrast variations (α_p in Eq.(1)), as well as 2) the trainable structure-preserving graph-based filtering (SGF) layer to smooth the domain-sensitive local noise/details (β_p) and capture the structural and geometric context as robust features for domain-invariant stereo reconstruction.

3.1 Domain Normalization

Batch normalization (BN) has become the default feature normalization operation for constructing end-to-end deep stereo matching networks [5, 19, 32, 47, 51, 63]. Although it can reduce the internal covariate shift effects in training

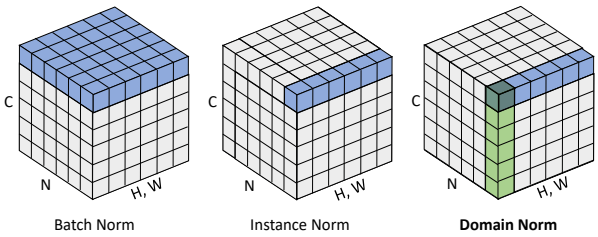
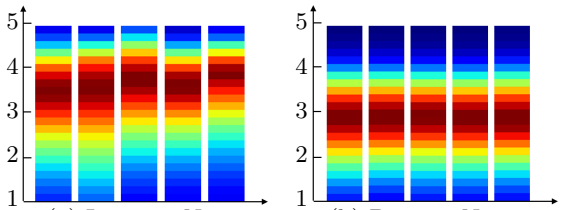


Fig. 2: Normalization methods. Each subplot shows a feature map tensor, with N as the batch axis, C as the channel axis, and (H, W) as the spatial axes. The blue elements in set S are normalized by the same mean and variance. The proposed domain normalization consists of image-level normalization (blue, Eq. (2)) and pixel-level normalization of each C-channel feature vector (green, Eq. (4)).



(a) Instance Norm (b) Domain Norm

Fig. 3: Norm (α_p of Eq.(1)) distributions of the features for different datasets (left to right: synthetic SceneFlow, KITTI, Middlebury, CityScapes and ETH 3D). The output of the feature extraction network is used for the study. The norm (α_p) of the feature vector at each pixel is counted. Instance normalization can only reduce the image-level differences, but does not normalize the C-channel feature vectors at pixel level.

deep networks, it is domain-dependent and has negative influence on the cross-domain generalization ability.

BN normalizes the features as follows:

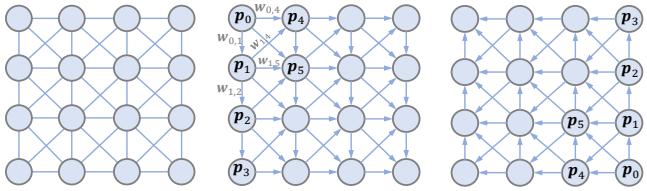
$$\hat{x}_i = \frac{1}{\sigma}(x_i - \mu_i). \tag{2}$$

Here x and $\hat{x}$ are the input and output features, respectively, and i indexes elements in a tensor (*i.e.* feature maps, as illustrated in Fig. 2) of size $N \times C \times H \times W$ (N: batch size, C: channels, H: spatial height, W: spatial width). μ_i and σ_i are the corresponding channel-wise mean and standard deviation (std) and are computed by:

$$\mu_i = \frac{1}{m}\sum_{k \in S_i} x_k, \quad \sigma_i = \sqrt{\frac{1}{m}\sum_{k \in S_i}(x_k - \mu_i)^2 + \epsilon}, \tag{3}$$

where S_i is the set of elements in the same channel as element i (Fig. 2), and ϵ is a small constant to avoid dividing by zeros.

Mean μ and standard deviation σ are computed per batch in the training phase, and the accumulated values of the training set are utilized for inference. However, different domains may have different μ and σ caused by color shifts, contrast, and illumination. (Fig. 1(a)–(b)). Thus μ and σ computed for one dataset are not transferable to others.



(a) 8-connected graph (b) directed graph G_1 (c) directed graph G_2

Fig. 4: Illustration of the graph construction. The 8-way connected graph is separated into two directed graphs G_1 and G_2.

Instance normalization (IN) [35, 40] overcomes the dependency on dataset statistics by normalizing each sample separately, where elements in S_i are confined to be from the same sample as illustrated in Fig. 2. In theory, IN is domain-invariant, and normalization across the spatial dimensions (H, W) reduces image-level style variations.

However, matching of stereo views is realized at the pixel level by finding an accurate correspondence for each pixel using its C-channel feature vector. Any inconsistence of the feature norm and scaling will significantly influence the matching cost and similarity measurements.

Fig. 3 illustrates that IN cannot regulate the norm distribution of pixel-wise feature vectors that vary in datasets/domains.

We propose in Fig. 2 our **domain-invariant normalization (DN)**. Our method normalizes features along the spatial axis (H, W) to induce style-invariant representations similar to IN as well as along the channel dimension (C) to enhance the local invariance.

Our DN is realized as follows:

$$\hat{x}'_i = \frac{\hat{x}_i}{\sqrt{\sum_{i \in S'_i} |\hat{x}_i|^2 + \epsilon}}, \tag{4}$$

where S'_i (green region in Fig. 2) includes C elements from the same example (N axis) and the same spatial location (H, W axis). $\hat{x}_i$ is computed as Eq. (2) and (3) with elements in S_i from the same channel and sample (blue in Fig. 2).

In our DN, besides normalization across spatial dimension, we also employ L_2 normalization to normalize features along the channel axis. They collaborate with each other to address the sensitivity to both image-level domain shift (α_I and β_I in Eq.(1)) and the local contrast variations (α_p). As illustrated in Fig. 3, it helps regulate the norm (α_p) distribution of the features in different datasets and improves the robustness to local contrast variations.

Finally, the trainable per-channel scale γ and shift β are added to enhance the discriminative representation ability as BN and IN. The final formulation is:

$$y_i = \gamma_i \ \hat{x}'_i + \beta_i. \tag{5}$$

3.2 Structure-preserving Graph-based Filtering

We propose a trainable Structure-preserving Graph-based Filter (SGF) that exploits contextual information and avoid solely memorizing local domain-sensitive texture patterns, details or noise (see Fig. 1(c)) for robust stereo matching.

Our inspiration comes from traditional graph-based filters that are remarkably effective in employing structural and geometric information for structure-preserving texture and detail removing/smoothing [62], denoising [6, 62], as well as depth-aware estimation and enhancement [29, 58].

Formulation For a 2D image/feature map I, we construct an 8-connected graph by connecting pixel $\mathbf{p}$ to its eight neighborhoods (see Fig. 4). To avoid loops and achieve fast information aggregation over the graph, we split it into two reverse directed graphs G_1, G_2 (see Fig. 4(b) and 4(c)).

We assign weight ω_e to each edge $e \in G$, and a feature (or color) vector $C(\mathbf{p})$ to each node $\mathbf{p} \in G$. We also allow $\mathbf{p}$ to propagate information to itself with weight $\omega_e(\mathbf{p}, \mathbf{p})$. For graph G_i $(i = 0, 1)$, our SGF is defined as follows:

$$C_i^A(\mathbf{p}) = \frac{\sum_{\mathbf{q}\in G_i} W(\mathbf{q},\mathbf{p})\cdot C(\mathbf{q})}{\sum_{\mathbf{q}\in G_i} W(\mathbf{q},\mathbf{p})}, \qquad W(\mathbf{q},\mathbf{p}) = \sum_{l_{\mathbf{q},\mathbf{p}}\in G_i} \prod_{e\in l_{\mathbf{q},\mathbf{p}}} \omega_e. \tag{6}$$

Here, $l_{\mathbf{q},\mathbf{p}}$ is a feasible path from $\mathbf{q}$ to $\mathbf{p}$. Note that $e(\mathbf{q}, \mathbf{q})$ is included in the path and counts for the start node $\mathbf{q}$. Unlike traditional geodesic filters, we consider all valid paths from source node $\mathbf{q}$ to target node $\mathbf{p}$. The propagation weight along path $l_{\mathbf{q},\mathbf{p}}$ is the product of all edge weights ω_e along the path. Here weight $W(\mathbf{q}, \mathbf{p})$ is defined as the sum of the weights of all feasible paths from $\mathbf{q}$ to $\mathbf{p}$, which determines how much information is diffused to $\mathbf{p}$ from $\mathbf{q}$.

For the edge weight $\omega_{(\mathbf{q},\mathbf{p})}$, we define it in a self-regularized manner as follows:

$$\omega_e(\mathbf{q}, \mathbf{p}) = \frac{\mathbf{x_p}^T \mathbf{x_q}}{\|\mathbf{x_p}\|_2 \cdot \|\mathbf{x_q}\|_2}, \tag{7}$$

where $\mathbf{x_p}$ and $\mathbf{x_q}$ represent the feature vectors of $\mathbf{p}$ and $\mathbf{q}$, respectively.

Compared to other local filters, such as Gaussian filter, median filter, and bilateral filter that can only propagate information in a local region determined by the filter kernel size, our SGF allows the propagation of long-range information over the whole image. More importantly, the filtering weights is defined as a spatial accumulation along all feasible paths in a graph. Similar to Geodesic filter [29] and tree filter [46, 59], this path-based filtering kernel helps better preserve the structures of the feature maps.

For stable training and to avoid extreme values, we further add a normalization constraint to the weights associated with $\mathbf{p}$ in the graph G_i as:

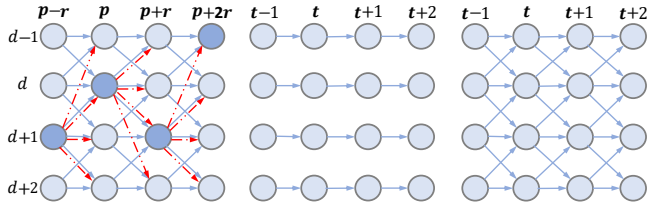
$$\sum_{\mathbf{q}\in N_\mathbf{p}} \omega_{e(\mathbf{q},\mathbf{p})} = 1. \tag{8}$$

Here, $N_\mathbf{p}$ is the set of the connected neighbors of $\mathbf{p}$ (including itself), and $e(\mathbf{q}, \mathbf{p})$ is the directed edge connecting $\mathbf{q}$ and $\mathbf{p}$. For example, in Fig. 4(b), for node $\mathbf{p}_0$, $\omega_{e(\mathbf{p}_0,\mathbf{p}_0)} = 1$; and for node $\mathbf{p}_4$, $\omega_{0,4} + \omega_{1,4} + \omega_{e(\mathbf{p}_4,\mathbf{p}_4)} = 1$.

If Eq. (8) holds, we can further derive $\sum_{\mathbf{q}\in G_i} W(\mathbf{q}, \mathbf{p}) = 1$*.Eq. (6) can then be simplified as follows:

$$C_i^A(\mathbf{p}) = \sum_{\mathbf{q}\in G_i} W(\mathbf{q},\mathbf{p}) \cdot C(\mathbf{q}), \qquad W(\mathbf{q},\mathbf{p}) = \sum_{l_{\mathbf{q},\mathbf{p}}\in G_i} \prod_{e\in l_{\mathbf{q},\mathbf{p}}} \omega_e. \tag{9}$$

*Proof is in the supplementary material: https://github.com/feihuzhang/DSMNet



(a) SGA [63] (b) one-way [30] (c) three-way [30]

Fig. 5: Special cases of our graph-based filter. (a) Semi-global aggregation (SGA) layer [63]. The dark blue node represents the maximum of each column. (b) and (c) are the affinity-based spatial propagations [30]. They aggregate from column t to $t+1$.

Such a transformation not only increases the robustness in training but also reduces the computational costs.

Linear Implementation Eq. (9) can be realized as an iterative linear aggregation, where the node is sequentially updated following the direction of the graph (*e.g.* top to bottom, then left to right in G_1). In each step, $\mathbf{p}$ is updated as:

$$\begin{aligned} C_i^A(\mathbf{p}) &= \omega_{e(\mathbf{p},\mathbf{p})} \cdot C(\mathbf{p}) + \sum_{\mathbf{q}\in N_\mathbf{p}, \mathbf{q}\neq\mathbf{p}} \omega_{e(\mathbf{q},\mathbf{p})} \cdot C_i^A(\mathbf{q}) \\ s.t. &\sum_{\mathbf{q}\in N_\mathbf{p}} \omega_{e(\mathbf{q},\mathbf{p})} = 1. \end{aligned} \tag{10}$$

Finally, we repeat the aggregation process for G_1 and G_2 where the updated representation with G_1 is used as the input for aggregation with G_2. The aggregation of Eq. (10) is a linear process with time complexity of $O(n)$ (with n nodes in the graph). During training, backpropagation can be realized by reversing the propagation which is also a linear process (*refer to the supplementary material*).

Relations to Existing Approaches We show that the recently proposed semi-global aggregation (SGA) layer [63] and affinity-based propagation approach [30] are special cases of our SGF (Eq. (9)). In addition, we compare it with non-local strategies, [54,56], graph neural networks [65] and the attention mechanism [16].

a) Semi-global Aggregation (SGA) [63] is proposed as a differentiable approximation of SGM [14] and can be presented as follows:

$$C_\mathbf{r}^A(\mathbf{p},d) = \text{sum}\begin{cases} \omega_0(\mathbf{p},\mathbf{r}) \cdot C(\mathbf{p},d) \\ \omega_1(\mathbf{p},\mathbf{r}) \cdot C_\mathbf{r}^A(\mathbf{p}-\mathbf{r},d) \\ \omega_2(\mathbf{p},\mathbf{r}) \cdot C_\mathbf{r}^A(\mathbf{p}-\mathbf{r},d-1) \\ \omega_3(\mathbf{p},\mathbf{r}) \cdot C_\mathbf{r}^A(\mathbf{p}-\mathbf{r},d+1) \\ \omega_4(\mathbf{p},\mathbf{r}) \cdot \max\limits_i C_\mathbf{r}^A(\mathbf{p}-\mathbf{r},i) \end{cases} s.t. \sum_{i=0,1,2,3,4} \omega_i(\mathbf{p},\mathbf{r}) = 1 \tag{11}$$

The aggregations are in four directions, namely $\mathbf{r} = \{(0,1),(0,-1),(1,0),(-1,0)\}$. Taking the right to left propagation ($\mathbf{r} = (0,1)$) as an example, we can construct a propagation graph in Fig. 5(a). The y-coordinate represents disparity d, and the x-coordinate is the indexes of the pixels/nodes. Compared to our graph in Fig. 4(b), edges connecting top and bottom nodes are removed, and the maximum of each column is densely connected to every node of the next column (red edges). Eq. (11) can then be realized by our SGF of Eq. (9). Here, $(\mathbf{p}-\mathbf{r}, d\pm1)$ are the neighborhood nodes of $\mathbf{p}$, and $\omega_{0,\dots4}$ are the corresponding edge weights.

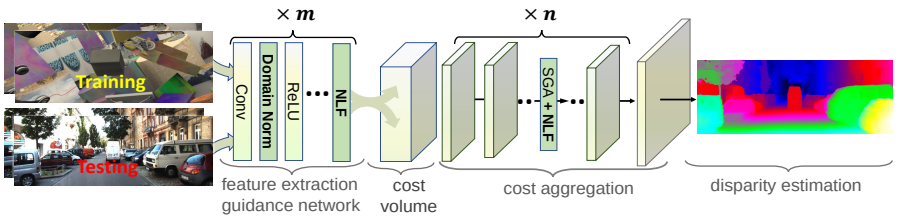


Fig. 6: Overview of the network architecture. Synthetic data are used for training, while using data from other new domains (*e.g.* real KITTI dataset) for testing. The backbone of GANet [63] is used as the baseline. The proposed DN layer is used after each convolutional layer in the feature extraction and guidance network. Several SGF layers are implemented for both feature extraction and cost aggregation.

b) Affinity-based Spatial Propagation in [30] can be achieved as:

$$C^A(\mathbf{p},d) = \left(1 - \sum_{\mathbf{q}\in N_\mathbf{p},\mathbf{q}\neq\mathbf{p}} \omega_{e(\mathbf{q},\mathbf{p})}\right) C(\mathbf{p}) + \sum_{\mathbf{q}\in N_\mathbf{p},\mathbf{q}\neq\mathbf{p}} \omega_{e(\mathbf{q},\mathbf{p})} C^A(\mathbf{q}), \qquad (12)$$

where $\omega_{e(\mathbf{q},\mathbf{p})}$ are the learned affinities. $1 - \sum_{\mathbf{q}\in N_\mathbf{p}} \omega_{e(\mathbf{q},\mathbf{p})}$ is equal to our weight $\omega_{\mathbf{e}(\mathbf{p},\mathbf{p})}$ for $\mathbf{p}$. The graphs for filtering can be constructed as in Fig. 5(b) and 5(c) for the one-way and three-way propagations [30], respectively.

c) Non-local Strategies, Graph Neural Networks and Attentions [16,45,54,56,65] can be used for non-local feature aggregation. But, they are implemented without spatial and structural awareness. Existing attentions and GNNs used in image segmentation task only consider the feature similarity for aggregation which treat pixel locations equally. In geometric problem (*e.g.* stereo matching), spatial proximity is crucial for learning accurate depths since pixels in the same object/class (with similar features) must be spatially close enough to have similar depth values. Therefore, these similarity/affinity based attentions and non-local networks will easily smooth out depth edges and thin structures (as illustrated in the supplementary material). Our SGF utilizes both the feature affinity and the spatial proximity for non-local graph-based filtering. It spatially aggregates the features along the paths which can better preserve the structure of the disparity maps. More importantly, Our graph filter has lower (linear) complexity in both memory requirement and computation since it is realized by the linear spatial propagation and the weight matrix is only $5 \times N$.

3.3 Network Architecture

As illustrated in Fig. 6, we utilize the backbone of GANet as the baseline architecture. The LGA layer in [63] is removed since it's domain-dependent and captures a lot of local patterns that are very sensitive to domain shifts.

We replace the original batch normalization layer by our proposed domain normalization layer for feature extraction. For the feature extraction network, we utilize a total of seven proposed filtering layers. For 3D cost aggregation of the cost volume, two SGF layers are further added for cost volume filtering in each channel/depth. *Details of the architecture are in the supplementary.*

4 Experimental Results

In our experiments, we train our method only with synthetic data and test it on four real datasets to evaluate its domain generalization ability. During training, we use disparity regression [19] for disparity prediction, and the smooth L_1 loss to compute the errors for back-propagation (the same as in [5, 63]). All the models are optimized with Adam ($\beta_1 = 0.9$, $\beta_2 = 0.999$). We train with a batch size of 8 on four GPUs using 288×624 random crops from the input images. The maximum of the disparity is set as 192. We train the model on the synthetic dataset for 10 epochs with a constant learning rate of 0.001. All other training settings are kept the same as those in [63].

4.1 Datasets

KITTI stereo 2012 [10] and 2015 [33] datasets provide about 400 image pairs of outdoor driving scenes for training, where the disparity labels are transformed from Velodyne LiDAR points. The **Cityscapes** dataset [8] provides a large amount of high-resolution ($1k \times 2k$) stereo images collected from city driving scenes. The disparity labels are pre-computed by SGM [14] which is not accurate enough for training deep neural network models. The **Middlebury** stereo dataset [42] is designed for indoor scenes with higher resolution (up to $2k \times 3k$). But it provides no more than 50 image pairs that are not enough to train robust deep neural networks. In addition, **ETH 3D** dataset [43] provides 27 pairs of gray images for training.

These existing real datasets are all limited by their small quantity or poor ground-truth labels, making them insufficient for training. Hence, we use them as test sets for evaluating our models' cross-domain generalization ability.

We mainly use synthetic data to train our domain-invariant models. The existing Scene Flow synthetic dataset [32] contains 35k training image pairs with a resolution of 540×960. This dataset has a limited number of the outdoor driving scenes that provide stereo pairs with a few settings of the camera baselines and image resolutions. We use CARLA [9] to generate a new supplementary synthetic dataset (with 20k stereo pairs) with more diverse settings, including two kinds of image resolutions (720×1080 and 1080×1920), three different focal lengths, and five different camera baselines (in a range of 0.2–1.5m). This supplementary dataset* can significantly improve the diversity of the training set.

The two advantages in using synthetic data are that it can avoid all the difficulties of labeling a large amount of real data, and that it can eliminate the negative influence of wrong depth values in real datasets.

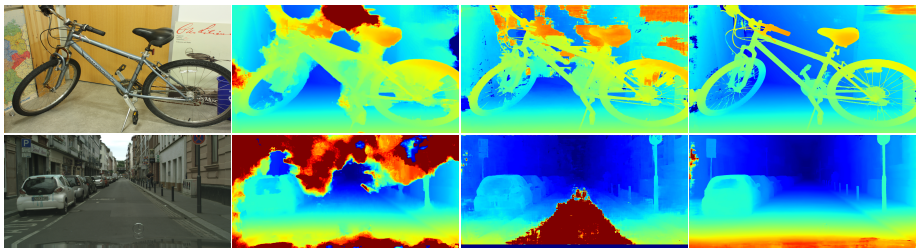
4.2 Ablation Study

We evaluate the performance of our DSMNet with numerous settings, including different architectures, normalization strategies and numbers (0–9) of the proposed SGF layers. As listed in Table 1, the full-setting DSMNet far outperforms the baseline in accuracy by 3% on the KITTI and 8% on the Middlebury datasets. Our proposed domain normalization improves the accuracy by about 1.5%, and the SGF layers contribute another 1.4% on the KITTI dataset.

*Available at https://github.com/feihuzhang/DSMNet.

Table 1: Ablation study. Models are trained on synthetic data (SceneFlow). Threshold error rates (%) are used for evaluations.

Normalization	SGF		Backbone	Middlebury	KITTI
	feature	cost volume		2-pixel	3-pixel
BN			ours	30.3	9.4
DN			ours	27.1	7.9
DN	+3		ours	24.2	7.1
DN	+7		ours	22.9	6.8
DN	+9		ours	22.4	6.8
DN	+7	+2	ours	**21.8**	**6.5**
BN			PSMNet	39.5	16.3
BN			GANet	32.2	11.7
DN	+7	+2	PSMNet	26.1	8.5
DN	+7	+2	GANet	23.7	7.3



(a) Input view (b) HD^3 [60] (c) PSMNet [5] (d) Our DSMNet

Fig. 7: Comparisons with state-of-the-art models. Models are trained on synthetic data and evaluated on high-resolution real datasets (Middlebury and CityScapes). Our DSMNet can produce much more accurate disparity estimation.

Moreover, our proposed layers are generic and could be seamlessly integrated into other deep stereo matching models. Here, we replace our backbone model with GANet [63] and PSMNet [5]. The accuracies are improved by 4–8% on KITTI dataset and 8–13% on Middlebury dataset for coss-domain evaluations compared with the original PSMNet and GANet.

4.3 Component Analysis and Comparisons

To further validate the superiorities of the proposed layers, we compare each of them with other related normalization and attention/affinity strategies.

Normalization Strategies. Table 2 compares our domain normalization with batch normalization [18], instance normalization [53], and the recently proposed adaptive batch-instance normalization [35]. There are also some adaptive BNs [26, 27] for domain adaptation, but they require the full access to the target dataset and cannot be used for more challenging domain generalization task. We keep all other settings the same as our DSMNet and only replace the normalization method for training and evaluation. Our domain normalization is superior to others for domain-invariant stereo matching because it can fully regulate the distribution of the feature vectors and remove both image-level and local contrast differences for cross-domain generalization.

Attentions and Non-local Approaches. Finally, we compare our SGF with attention and non-local networks, including affinity-based propagation [30], non-local

Table 2: Comparisons with Existing Normalization and Filtering/Attention Strategies

Normalization	Middlebury (full)	KITTI	Affinity/Attention	Middlebury (full)	KITTI
Batch Norm	29.1	7.3	Attention [16]	25.2	5.9
Instance Norm	27.1	6.4	Denoising [56]	25.9	6.1
Adaptive Norm [35]	28.2	6.8	Affinity [30]	23.1	5.2
Our Domain Norm	20.1	4.1	Our Graph Filter	20.1	4.1

Table 3: Evaluations on the KITTI, Middlebury, and ETH 3D validation datasets. Threshold error rates (%) are used.

Models	KITTI		Middlebury			ETH3D	Carla
	2012	2015	full	half	quarter		
CostFilter [15]	21.7	18.9	57.2	40.5	17.6	31.1	41.1
PatchMatch [2]	20.1	17.2	50.2	38.6	16.1	24.1	30.1
SGM [14]	7.1	7.6	38.1	25.2	10.7	12.9	20.2
Training set	SceneFlow						
HD^3 [60]	23.6	26.5	50.3	37.9	20.3	54.2	35.7
gwcnet [13]	20.2	22.7	47.1	34.2	18.1	30.1	33.2
PSMNet [5]	15.1	16.3	39.5	25.1	14.2	23.8	25.9
GANet [63]	10.1	11.7	32.2	20.3	11.2	14.1	18.8
Our DSMNet	**6.2**	**6.5**	**21.8**	**13.8**	**8.1**	**6.2**	**9.8**
Training set	SceneFlow + Carla						
HD^3 [60]	19.1	19.5	47.3	35.2	19.5	45.2	–
gwcnet [13]	17.2	18.1	45.2	31.8	17.2	29.4	–
PSMNet [5]	10.3	11.0	35.5	23.7	13.8	20.3	–
GANet [63]	7.2	7.6	31.9	19.7	11.4	13.5	–
Our DSMNet	**3.9**	**4.1**	**20.1**	**13.6**	**8.2**	**6.0**	–

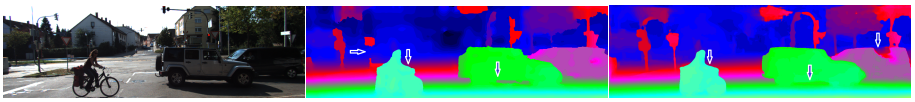
neural network denoising [56], and non-local attention [16] (in Table 2). Our SGF layer is better for capturing the structural and geometric context for robust domain-invariant stereo matching. The non-local neural network denoising [56] and non-local attention [16] do not have spatial constraints that usually lead to smoothness of the depth edges (as shown in the supplementary material). Affinity-based propagations [30] are special cases of our proposed SGF and are not as effective in feature and cost volume aggregations for stereo matching.

4.4 Cross-domain Evaluations

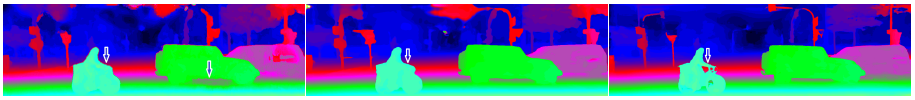
In this section, we compare our DSMNet with state-of-the-art stereo matching models by training with synthetic data and evaluating on real test sets.

Comparisons with State-of-the-Art Models. In Table 3 and Fig. 7, we compare our DSMNet with other state-of-the-art neural network models on the four real datasets. All models are trained on synthetic data (either SceneFlow or a mixture of SceneFlow and Carla). We find that DSMNet far outperforms the state-of-the-art models by 3–30% in error rates for all these datasets. It is also far better than traditional algorithms, like SGM [14], costfilter [15] and patchmatch [2].

Evaluation on the KITTI Benchmark. Table 4 presents the performance of our DSMNet on the KITTI benchmark [33]. Our model far outperforms most of the unsupervised/self-supervised models trained on the KITTI domain. It is even better than supervised stereo matching networks (including, MC-CNN [61], content-CNN [31], and DispNetC [32]) trained or fine-tuned on the KITTI dataset. When compared with other fine-tuned state-of-the-art models (*e.g.*



(a) Input view (b) MC-CNN [61] (c) PSMNet [5]



(d) HD^3 [60] (e) GANet-deep [63] (f) Our DSMNet-synthetic

Fig. 8: Comparisons with the fine-tuned state-of-the-art models. Our model is trained only with synthetic data. All others are fine-tuned on the KITTI target scenes. As pointed by arrows, our DSMNet can produce more accurate object boundaries.

Table 4: Cross-domain evaluation on KITTI 2015 benchmark (all area). DSMNet is trained only with synthetic data.

Models	Training Set	Error Rate (%)
Our DSMNet	**Synthetic**	**3.71**
MC-CNN-acrt [61]	Kitti-gt	3.89
DispNetC [32]	Kitti-gt	4.34
Content-CNN [31]	Kitti-gt	4.54
MADNet-finetune [51]	Kitti-gt	4.66
Weak Supervise [52]	Kitti-gt	4.97
MADNet [51]	Kitti (no gt)	8.23
OASM-Net [21]	Kitti (no gt)	8.98
Unsupervised [68]	Kitti (no gt)	9.91

Table 5: In-domain (after fine-tuning) evaluation (error rates: %) on the KITTI 2015 Benchmark

Models	Non-Occluded	All Area
GANet + **Our SGF**	**1.58**	**1.77**
GANet-deep [63]	1.63	1.81
DSMNet-finetune	1.71	1.90
AcfNet [66]	1.72	1.89
GANet-15 [63]	1.73	1.93
HD^3 [60]	1.87	2.02
gwcnet-g [13]	1.92	2.11
PSMNet [5]	2.14	2.32
GCNet [19]	2.61	2.87

PSMNet [5], HD^3 [60], GANet-deep [63]), our DSMNet (without fine-tuning) produces more accurate object boundaries (Fig. 8).

4.5 Fine-tuning

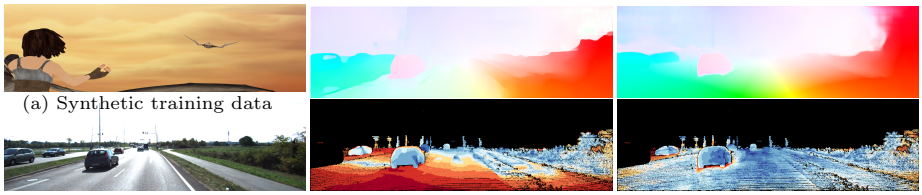
In this section, we show the best performance of our DSMNet when fine-tuned on the target domain. We fine-tune the model pre-trained on synthetic data for a further 700 epochs using the KITTI 2015 training set. The learning rate begins at 0.001 for the first 300 epochs and decreases to 0.0001 for the rest. The results of the test set are submitted to KITTI 2015 benchmark for evaluations.

Table 5 compares the results of the fine-tuned DSMNet and those of other state-of-the-art DNN models. We find that DSMNet outperforms most of the recent models (including PSMNet [5], HD^3 [60], GwcNet [13] and GANet-15 [63]) by a noteworthy margin. This implies that DSMNet can achieve the same accuracy by fine-tuning on one specific dataset, without sacrificing accuracy to improve its generalization ability.

We also separately test the effectiveness of our SGF layer. Using the current best "GANet-deep" [63] (including the Local Guided Aggregation layer) as the baseline, we add five filtering layers for feature extraction. All other settings are kept the same as the original GANet. After training on synthetic data and fine-tuning on the KITTI training dataset, the new model got a new state-of-the-art accuracy (1.58%) and **ranked No. 1** on KITTI 2015 benchmark (non-occluded

Table 6: Evaluations of the Optical Flow Networks for Cross-domain Generalization

Original Models	Error Rates (%)	Improved Models	Error Rates (%)
FlowNet2 [63]	34.1	Domain-invaraint FlowNet2	16.2
PwcNet [48]	16.9	Domain-invaraint PwcNet	11.2



(a) Synthetic training data

(b) Real test view (c) FlowNet2 [17] (d) Domain-invariant FlowNet2

Fig. 9: Performance illustration with optical flow. Models are trained only with synthetic data. (a) Example of the synthetic training data (MPI Sintel [4]), (b) the real test view from KITTI 2015 dataset, (c) the result (top) and the error map (bottom) of the original FlowNet2, (d) the result (top) and the error map (bottom) of the domain-invariant FlowNet2 powered by our DSMNet.

area, by the time of submission). This shows that our SGF can improve not only cross-domain generalization but also the accuracy on the test domains.

5 Extension for Optical Flow

Similar to stereo matching, optical flow is also based on pixel-to-pixel similarity measurement for dense correspondence matching between two different images. Therefore, our domain-invariant matching network can be easily extended to the optical flow task. We use FlowNet2 [17] and PwcNet [48] as baselines and employ our DN and graph filtering to to realize the domain-invariant optical flow networks. The models are trained on synthetic FlyingThings3D [32] and MPI Sintel [4] datasets and evaluated on real flow dataset (KITTI 2015). As shown in Table 6 and Fig. 9, Accuracies are significantly improved by 5.7–17% in the cross-domain evaluations. This further demonstrates the effectiveness of our proposed domain-invariant network.

6 Conclusion

In this paper, we proposed two end-to-end trainable neural network layers for our domain-invariant stereo matching network. Our novel domain normalization can fully regulate the distribution of learned features to address significant domain shifts, and our SGF can capture more robust non-local structural and geometric features for accurate disparity estimation in cross-domain situations. We have verified our model on four real datasets and shown its superior accuracy when compared to other state-of-the-art models in the cross-domain generalization.

Acknowledgement

Research is supported by Baidu, the ERC grant ERC-2012-AdG 321162-HELIOS, EPSRC grant Seebibyte EP/M013774/1 and EPSRC/MURI grant EP/N019474/1. We would also like to acknowledge the Royal Academy of Engineering.

References

1. Balaji, Y., Sankaranarayanan, S., Chellappa, R.: Metareg: Towards domain generalization using meta-regularization. In: Advances in Neural Information Processing Systems. pp. 998–1008 (2018)
2. Bleyer, M., Rhemann, C., Rother, C.: Patchmatch stereo-stereo matching with slanted support windows. In: British Machine Vision Conference (BMVC). pp. 1–11 (2011)
3. Bousmalis, K., Silberman, N., Dohan, D., Erhan, D., Krishnan, D.: Unsupervised pixel-level domain adaptation with generative adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). pp. 3722–3731 (2017)
4. Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A naturalistic open source movie for optical flow evaluation. In: A. Fitzgibbon et al. (Eds.) (ed.) European Conf. on Computer Vision (ECCV). pp. 611–625. Part IV, LNCS 7577, Springer-Verlag (Oct 2012)
5. Chang, J.R., Chen, Y.S.: Pyramid stereo matching network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5410–5418 (2018)
6. Chen, X., Bing Kang, S., Yang, J., Yu, J.: Fast patch-based denoising using approximated patch geodesic paths. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1211–1218 (2013)
7. Choy, C.B., Gwak, J., Savarese, S., Chandraker, M.: Universal correspondence network. In: Advances in Neural Information Processing Systems. pp. 2414–2422 (2016)
8. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). pp. 3213–3223 (2016)
9. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: Carla: An open urban driving simulator. arXiv preprint arXiv:1711.03938 (2017)
10. 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 (CVPR). pp. 3354–3361. IEEE (2012)
11. Ghifary, M., Bastiaan Kleijn, W., Zhang, M., Balduzzi, D.: Domain generalization for object recognition with multi-task autoencoders. In: Proceedings of the IEEE international conference on computer vision (ICCV). pp. 2551–2559 (2015)
12. Guo, X., Li, H., Yi, S., Ren, J., Wang, X.: Learning monocular depth by distilling cross-domain stereo networks. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 484–500 (2018)
13. Guo, X., Yang, K., Yang, W., Wang, X., Li, H.: Group-wise correlation stereo network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3273–3282 (2019)
14. Hirschmuller, H.: Stereo processing by semiglobal matching and mutual information. IEEE Transactions on Pattern Analysis and Machine Intelligence **30**(2), 328–341 (2008)
15. Hosni, A., Rhemann, C., Bleyer, M., Rother, C., Gelautz, M.: Fast cost-volume filtering for visual correspondence and beyond. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**(2), 504–511 (2013)

16. Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., Liu, W.: Ccnet: Criss-cross attention for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 603–612 (2019)
17. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: Flownet 2.0: Evolution of optical flow estimation with deep networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). pp. 2462–2470 (2017)
18. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
19. 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. CoRR, vol. abs/1703.04309 (2017)
20. Khamis, S., Fanello, S.R., Rhemann, C., Kowdle, A., Valentin, J.P.C., Izadi, S.: Stereonet: Guided hierarchical refinement for real-time edge-aware depth prediction. CoRR **abs/1807.08865** (2018)
21. Li, A., Yuan, Z.: Occlusion aware stereo matching via cooperative unsupervised learning. In: Asian Conference on Computer Vision. pp. 197–213. Springer (2018)
22. Li, D., Yang, Y., Song, Y.Z., Hospedales, T.M.: Deeper, broader and artier domain generalization. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 5542–5550 (2017)
23. Li, D., Yang, Y., Song, Y.Z., Hospedales, T.M.: Learning to generalize: Meta-learning for domain generalization. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (2018)
24. Li, H., Jialin Pan, S., Wang, S., Kot, A.C.: Domain generalization with adversarial feature learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5400–5409 (2018)
25. Li, Y., Tian, X., Gong, M., Liu, Y., Liu, T., Zhang, K., Tao, D.: Deep domain generalization via conditional invariant adversarial networks. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 624–639 (2018)
26. Li, Y., Wang, N., Shi, J., Hou, X., Liu, J.: Adaptive batch normalization for practical domain adaptation. Pattern Recognition **80**, 109–117 (2018)
27. Li, Y., Wang, N., Shi, J., Liu, J., Hou, X.: Revisiting batch normalization for practical domain adaptation. arXiv preprint arXiv:1603.04779 (2016)
28. 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 (CVPR). pp. 2811–2820 (2018)
29. Liu, M.Y., Tuzel, O., Taguchi, Y.: Joint geodesic upsampling of depth images. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). pp. 169–176 (2013)
30. Liu, S., De Mello, S., Gu, J., Zhong, G., Yang, M.H., Kautz, J.: Learning affinity via spatial propagation networks. In: Advances in Neural Information Processing Systems. pp. 1520–1530 (2017)
31. 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 (CVPR). pp. 5695–5703 (2016)
32. Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T.: A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4040–4048 (2016)

33. Menze, M., Geiger, A.: Object scene flow for autonomous vehicles. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3061–3070 (2015)
34. Motiian, S., Piccirilli, M., Adjeroh, D.A., Doretto, G.: Unified deep supervised domain adaptation and generalization. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 5715–5725 (2017)
35. Nam, H., Kim, H.E.: Batch-instance normalization for adaptively style-invariant neural networks. In: Advances in Neural Information Processing Systems. pp. 2558–2567 (2018)
36. Nie, G.Y., Cheng, M.M., Liu, Y., Liang, Z., Fan, D.P., Liu, Y., Wang, Y.: Multi-level context ultra-aggregation for stereo matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3283–3291 (2019)
37. Pan, X., Luo, P., Shi, J., Tang, X.: Two at once: Enhancing learning and generalization capacities via ibn-net. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 464–479 (2018)
38. Pang, J., Sun, W., Ren, J.S., Yang, C., Yan, Q.: Cascade residual learning: A two-stage convolutional neural network for stereo matching. IEEE International Conference on Computer Vision Workshops (ICCVW) (2017)
39. Pang, J., Sun, W., Yang, C., Ren, J., Xiao, R., Zeng, J., Lin, L.: Zoom and learn: Generalizing deep stereo matching to novel domains. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2070–2079 (2018)
40. Park, T., Liu, M.Y., Wang, T.C., Zhu, J.Y.: Semantic image synthesis with spatially-adaptive normalization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2337–2346 (2019)
41. Poggi, M., Pallotti, D., Tosi, F., Mattoccia, S.: Guided stereo matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 979–988 (2019)
42. Scharstein, D., Hirschmüller, H., Kitajima, Y., Krathwohl, G., Nešić, N., Wang, X., Westling, P.: High-resolution stereo datasets with subpixel-accurate ground truth. In: German conference on pattern recognition. pp. 31–42. Springer (2014)
43. Schops, T., Schonberger, J.L., Galliani, S., Sattler, T., Schindler, K., Pollefeys, M., Geiger, A.: A multi-view stereo benchmark with high-resolution images and multi-camera videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3260–3269 (2017)
44. Seki, A., Pollefeys, M.: Sgm-nets: Semi-global matching with neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6640–6649 (2017)
45. Shi, L., Zhang, Y., Cheng, J., Lu, H.: Non-local graph convolutional networks for skeleton-based action recognition. arXiv preprint arXiv:1805.07694 (2018)
46. Song, L., Li, Y., Li, Z., Yu, G., Sun, H., Sun, J., Zheng, N.: Learnable tree filter for structure-preserving feature transform. In: Advances in Neural Information Processing Systems. pp. 1709–1719 (2019)
47. Song, X., Zhao, X., Fang, L., Hu, H.: Edgestereo: An effective multi-task learning network for stereo matching and edge detection. arXiv preprint arXiv:1903.01700 (2019)
48. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8934–8943 (2018)

49. Tonioni, A., Poggi, M., Mattoccia, S., Di Stefano, L.: Unsupervised adaptation for deep stereo. In: The IEEE International Conference on Computer Vision (ICCV) (Oct 2017)
50. Tonioni, A., Rahnama, O., Joy, T., Stefano, L.D., Ajanthan, T., Torr, P.H.: Learning to adapt for stereo. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9661–9670 (2019)
51. Tonioni, A., Tosi, F., Poggi, M., Mattoccia, S., Stefano, L.D.: Real-time self-adaptive deep stereo. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 195–204 (2019)
52. Tulyakov, S., Ivanov, A., Fleuret, F.: Weakly supervised learning of deep metrics for stereo reconstruction. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 1339–1348 (2017)
53. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022 (2016)
54. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7794–7803 (2018)
55. 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. arXiv preprint arXiv:1810.11408 (2018)
56. Xie, C., Wu, Y., Maaten, L.v.d., Yuille, A.L., He, K.: Feature denoising for improving adversarial robustness. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 501–509 (2019)
57. Yang, G., Zhao, H., Shi, J., Deng, Z., Jia, J.: Segstereo: Exploiting semantic information for disparity estimation. arXiv preprint arXiv:1807.11699 (2018)
58. Yang, Q.: A non-local cost aggregation method for stereo matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1402–1409. IEEE (2012)
59. Yang, Q.: Stereo matching using tree filtering. IEEE Transactions on Pattern Analysis and Machine Intelligence **37**(4), 834–846 (2014)
60. Yin, Z., Darrell, T., Yu, F.: Hierarchical discrete distribution decomposition for match density estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6044–6053 (2019)
61. Zbontar, J., LeCun, Y.: Computing the stereo matching cost with a convolutional neural network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1592–1599 (2015)
62. Zhang, F., Dai, L., Xiang, S., Zhang, X.: Segment graph based image filtering: fast structure-preserving smoothing. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 361–369 (2015)
63. 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 (CVPR). pp. 185–194 (2019)
64. Zhang, F., Wah, B.W.: Fundamental principles on learning new features for effective dense matching. IEEE Transactions on Image Processing **27**(2), 822–836 (2018)
65. Zhang, S., Yan, S., He, X.: Latentgnn: Learning efficient non-local relations for visual recognition. arXiv preprint arXiv:1905.11634 (2019)
66. 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 Thirty-Second AAAI Conference on Artificial Intelligence (2020)

67. Zhong, Y., Dai, Y., Li, H.: Self-supervised learning for stereo matching with self-improving ability. arXiv preprint arXiv:1709.00930 (2017)
68. Zhou, C., Zhang, H., Shen, X., Jia, J.: Unsupervised learning of stereo matching. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 1567–1575 (2017)