

# Local Similarity Pattern and Cost Self-Reassembling for Deep Stereo Matching Networks

Biyang Liu<sup>1,2</sup>, Huimin Yu<sup>1,2,3,4</sup>, Yangqi Long<sup>1,2</sup>

<sup>1</sup> College of Information Science and Electronic Engineering, Zhejiang University

<sup>2</sup> ZJU-League Research & Development Center

<sup>3</sup> State Key Lab of CAD&CG, Zhejiang University

<sup>4</sup> Zhejiang Provincial Key Laboratory of Information Processing, Communication and Networking  
{biyangliu, yhm2005, longyangqi}@zju.edu.cn

## Abstract

Although convolutional neural network based stereo matching architectures have made impressive achievements, there are still some limitations: 1) Convolutional Feature (CF) tends to capture appearance information, which is inadequate for accurate matching. 2) Due to the static filters, current convolution based disparity refinement modules often produce over-smooth results. In this paper, we present two schemes to address these issues, where some traditional wisdoms are integrated. Firstly, we introduce a pairwise feature for deep stereo matching networks, named LSP (Local Similarity Pattern). Through explicitly revealing the neighbor relationships, LSP contains rich structural information, which can be leveraged to aid CF for more discriminative feature description. Secondly, we design a dynamic self-reassembling refinement strategy and apply it to the cost distribution and the disparity map respectively. The former could be equipped with the uni-modal distribution constraint to alleviate the over-smoothing problem, and the latter is more practical. The effectiveness of the proposed methods is demonstrated via incorporating them into two well-known basic architectures, GwcNet and GANet-deep. Experimental results on the SceneFlow and KITTI benchmarks show that our modules significantly improve the performance of the model. Code is available at <https://github.com/SpadeLiu/Lac-GwcNet>.

## Introduction

Depth information is essential for plenty of applications such as autonomous driving and augmented reality. As a convenient and low-cost means to obtain depth, stereo matching has been studied for a long time. The objective of this task is to estimate the disparity from a rectified image pair, which can be converted to depth with the focal length and baseline of the camera system. Typically, stereo matching algorithms consist of four steps (Scharstein and Szeliski 2002): matching cost computation, cost aggregation, disparity optimization, and disparity refinement.

Although convolutional neural network (CNN) based stereo matching methods have been dominant on several benchmarks, the vanilla convolution is not perfect, there are still some flaws in CNN based architectures. Previous works have proved that integrating traditional knowledge will benefit the designing of the model (Zhang et al. 2019; Xu and

Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

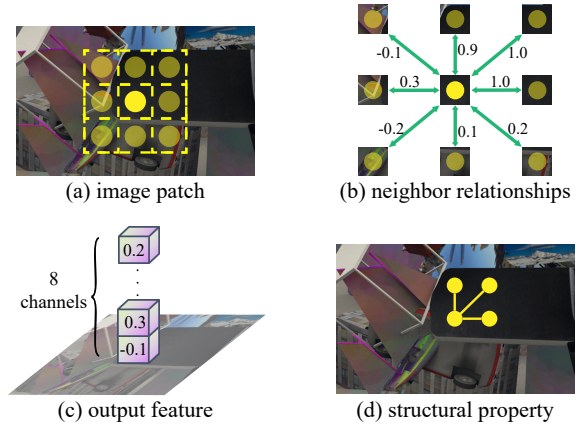


Figure 1: The operation paradigm of LSP. Each yellow circle represents a point in the image or the feature map. Local structural property is reflected in the close relationships.

Zhang 2020). However, they concentrate more on the cost aggregation stage. In this paper, we mainly investigate some limitations of the used features and the refinement modules. Aiming at breaking the limitations, we present two schemes, introduced as follows.

1) Convolutional Feature (CF) is utilized to describe the points to construct the matching cost in current stereo matching networks. Despite the rich appearance information, CF contains less structural information since the learned convolutional kernels can not explicitly reveal the pixel relationships. Due to the view transformation, the appearance of an object might get changed while the structural characteristic is more stable (Zabih and Woodfill 1994). Therefore, it is essential for accurate matching to combine the appearance information and the structural information, as demonstrated in AD-Census (Mei et al. 2011).

To this end, we design a pairwise feature for deep stereo matching networks, named LSP (Local Similarity Pattern). Based on the unary features extracted from the convolutional layers, the relationships (e.g. cosine similarities) of a point and its neighbors are calculated as LSP, as illustrated in Figure 1. With such an operating mechanism, LSP clearly discloses some structural clues (e.g. trees are vertical), making it a beneficial complement for CF. At the same time, LSP

is calculated in a bottom-up fashion (Hu et al. 2019), thus it is unnecessary to iterate multiple times to obtain a representative description. Further to make LSP more complete, we extend it with a multi-scale and multi-level scheme. The former is aimed at handling objects of various sizes, and the latter is designed considering the unary features from different convolutional layers contain different information.

2) At present, most refinement methods are based on a CNN architecture that takes multi-modal inputs to predict a disparity residual map (Khamis et al. 2018; Liang et al. 2018). We observe the results from these methods tend to be over-smooth (Chen, Chen, and Cheng 2019). The reason might be that the learned static and shared convolutional filters are difficult to restore various error patterns. Under the guidance of the  $L1$  loss, convolutions try to reduce the average error. It is important to address the over-smoothing problem, especially for some downstream tasks, such as 3D object detection (Garg et al. 2020).

Inspired by the elegant disparity filling methods (Huq, Koschan, and Abidi 2013), we propose a dynamic refinement strategy CSR (Cost Self-Reassembling). The basic hypothesis is that the true cost distribution of an outlier should be close to those of some neighbors. Rather than manually searching for these close and reliable neighbors, we build a network and force it to learn to predict the locations of the neighbors by itself. In this way, the disparity propagation process will adapt to the image content and is not limited to the regular grid, making CSR more powerful than other CNN based refinement modules. By further imposing the unimodal distribution constraint on the refined cost, the over-smoothing problem can be effectively mitigated (Chen, Chen, and Cheng 2019). In addition to the cost distribution, the dynamic reassembling strategy could be applied to the disparity map and DSR (Disparity Self-Reassembling) is more suitable for practical algorithms.

Our main contributions are summarized as:

- We introduce a pairwise feature LSP for deep stereo matching networks. The structural information contained in LSP makes it a beneficial supplement for CF.
- We propose two dynamic disparity refinement strategies CSR and DSR. In particular, CSR can be equipped with the unimodal distribution constraint to alleviate the over-smoothing problem.
- Through inserting the designed modules, we surpass the baseline GwcNet by a substantial margin on the SceneFlow and KITTI benchmarks. We also incorporate them into a state-of-the-art (SOTA) model GANet-deep and achieve a significant performance improvement.

## Related Works

### Deep Stereo Matching Networks

In early works, CNNs were used to compute the matching costs (Zbontar and LeCun 2015; Chen et al. 2015; Luo, Schwing, and Urtasun 2016), followed by traditional cost aggregation and disparity refinement methods. Nowadays, end-to-end frameworks are more popular. According to the dimension of the convolutions used in cost aggregation, they

can be divided into two categories: 2D-CNN based models (Mayer et al. 2016; Song et al. 2020; Xu and Zhang 2020) and 3D-CNN based models (Kendall et al. 2017; Chang and Chen 2018; Guo et al. 2019; Zhang et al. 2019). Generally speaking, stereo matching networks belonging to the former category are more efficient, yet those belonging to the latter category perform better.

### Structural Features

CT (Census Transform) (Zabih and Woodfill 1994) and LBP (Local Binary Pattern) (Ojala, Pietikainen, and Maenpää 2002) are two well-known structural features in the field of computer vision. Despite the great achievements made in traditional algorithms, the non-differentiable Heaviside step function prevents them from being applied to the learned features of CNNs. In LSP, we alternatively utilize cosine similarity to measure the relationships. Apart from making LSP trainable, it solves the problem that the input of CT can only be the 1-channel grayscale image. We also note the affinity feature in (Gan et al. 2018). Except for LSP is more complete with a multi-scale and multi-level scheme, our motivation is different. Affinity feature is used to understand the depth relationships (Gan et al. 2018), while LSP is to describe a point with structural information.

### Disparity Refinement

In current refinement modules, a CNN architecture is built to predict the disparity residual given the image features, projection error, etc (Khamis et al. 2018; Liang et al. 2018). Due to the stationarity of the convolutional filters, existing methods tend to produce over-smooth results (Chen, Chen, and Cheng 2019). The problem is alleviated by our CSR, which is inspired by the classic disparity filling methods (Huq, Koschan, and Abidi 2013; Tosi, Poggi, and Mattocchia 2019). At the same time, it absorbs the advantage of deep learning. Instead of manually searching, we predict the locations of the neighbors with a network (Dai et al. 2017) based on the image content. A similar strategy is adopted in the refinement of the depth maps (Ramamonjisoa, Du, and Lepetit 2020). From the formula, it can be regarded as an instance of our DSR that assembles only one neighbor. Moreover, we predict the offsets without the initial disparity since our module can learn to find accurate neighbors by itself.

## Method

### Overall Architecture

In the following sections, we will describe the proposed modules and show how to insert them into a general deep stereo matching network. With GwcNet (Guo et al. 2019) as the basic model, the overall architecture is shown in Figure 2, where the feature extraction module after embedding single-level or multi-level LSP and the refinement module are detailedly displayed. Note the cost volume in our model is constructed by feature concatenation rather than group-wise correlation (Guo et al. 2019) since feature concatenation is a more popular manner.

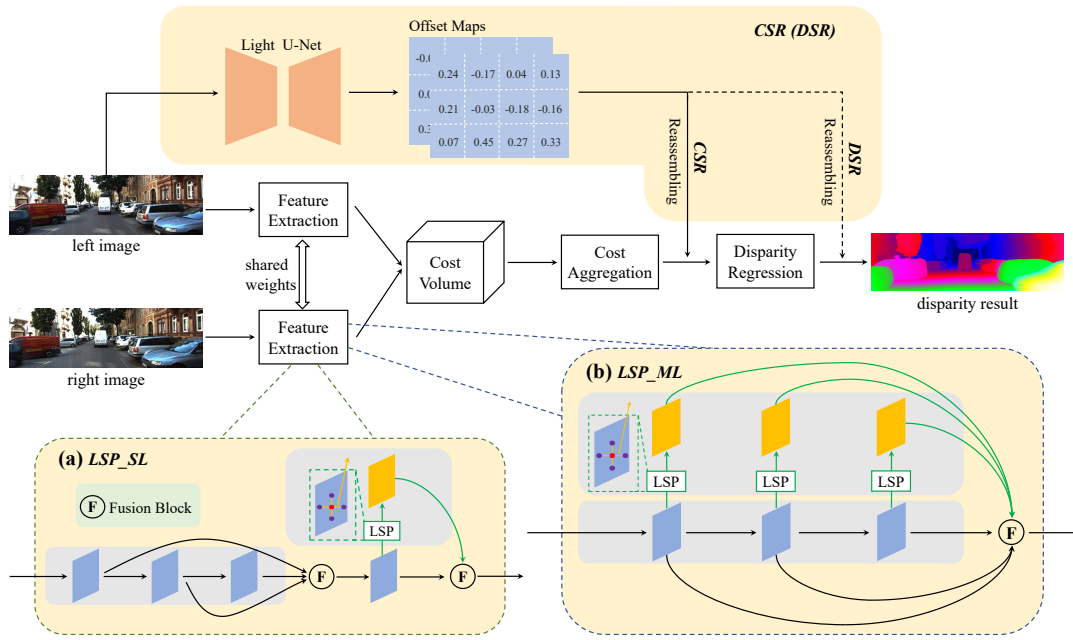


Figure 2: Overall architecture of our model based on GwcNet. Single-level or multi-level LSP is calculated with the convolutional features, then they are fused together for subsequent processings. The cost volume is formed via concatenating the left and the right features and is further aggregated by several 3D convolutions. Final disparity result is regressed by *softmax* and weighted average. The self-reassembling refinement strategy can be applied to the cost distribution or the disparity map.

## Local Similarity Pattern

While CF contains rich appearance information, the structural information is often ignored. The reason lies in the fact that the learned weights in convolution can not explicitly reveal the neighbor relationships. To compensate for the lack, we introduce a pairwise feature, referred to as LSP. In LSP, each channel represents the relationship between the central point and one of its neighbors. It is formulated as:

$$f_L^k(x, y) = \Phi(\mathbf{f}_C(x, y), \mathbf{f}_C(x + \delta x_k, y + \delta y_k)) \quad (1)$$

where  $f_L^k(x, y)$  denotes the  $k$ -th channel of the LSP of point  $(x, y)$ ,  $\mathbf{f}_C$  is the convolutional feature with multiple channels, and  $(\delta x_k, \delta y_k)$  represents the offset of the  $k$ -th neighbor point.  $\Phi(\cdot, \cdot)$  is a function to measure the similarity of two vectors, which will output a scalar. We select cosine similarity for its numerical stability.

Similar to convolution, we adopt a square window to define the neighborhood, as illustrated in Figure 1. Note the output channels of  $\mathbf{f}_L$  are only decided by the number of the neighbor points, i.e. the size of the window. Another important difference from CF is that LSP is dynamic and manually calculated (Hu et al. 2019). By this means, it is unnecessary to stack multiple layers to recognize various local patterns, making LSP convenient to embed.

Intuitively, smaller neighborhoods are beneficial for the matching at disparity discontinuities, while larger neighborhoods can help to understand the structures in textureless regions. To fully recognize the objects of various sizes, we design a **multi-scale** scheme to construct the pairwise relationships in different scopes (i.e. spatial resolutions). Considering the pooling strategy (He et al. 2015) might lead to a

loss of detail, we adopt the dilation strategy (Yu and Koltun 2015) to enlarge the receptive field.

In CNNs, low-level features contain rich texture information while high-level features normally reveal semantic clues. With the unary features extracted from different layers, the calculated pairwise relationships are diverse. Therefore, we design a **multi-level** scheme for LSP. As shown in Figure 2 (a) and (b), under the single-level (SL) setting, we only calculate LSP with the fused convolutional feature. Under the multi-level (ML) setting, the multi-level convolutional features are separately used to calculate LSP.

In practice,  $1 \times 1$  convolution is utilized to adjust the channels for each LSP (not shown in the figure). After that, multiple LSPs are fused with the original CFs by the fusion block shown in Figure 2, which consists of a concatenation operation and one  $1 \times 1$  convolutional layer. In this way, the fused feature will learn to balance the appearance information and the structural information by itself.

## Cost Self-Reassembling

Stereo matching models tend to make mistakes at occluded areas since the matching candidate points are invisible in the other view. Fortunately, in most cases only parts of the objects are occluded. We can leverage the accurate disparity values obtained in the unoccluded regions to infer those of the occluded regions, according to the disparity relationships among them. This is the basic hypothesis of traditional disparity filling methods (Huq, Koschan, and Abidi 2013).

Current disparity refinement methods rectify the outliers by predicting the residual with convolutions (Khamis et al.

2018; Xu and Zhang 2020). Due to the stationarity of the filter, it is difficult for convolutions to restore various local error patterns. As a result, existing convolution based refinement methods tend to produce over-smooth results to reduce the overall error. To adaptively refine the disparity, we propose CSR, which holds a similar spirit as the disparity filling methods but possesses the advantage of being trainable.

Specifically, we first excavate the contextual information in the image to understand the disparity relationships between the points. It is realized by a lightweight U-Net (Ronneberger, Fischer, and Brox 2015) which takes the left image as input and outputs dense features. After that, for each point, several offsets are predicted with the features. Each offset contains two channels, indicating the pixel coordinate of a neighbor point. With the coordinates, we sample the neighbor cost distributions and combine them to update the central one. This reassembling process is formulated as:

$$\mathbf{C}_r(x, y) = \frac{1}{N} \sum_{i=1}^N \mathbf{C}_0(x + \Delta x_i, y + \Delta y_i) \quad (2)$$

where  $\mathbf{C}_r$  and  $\mathbf{C}_0$  are the refined and initial cost distribution, both with  $d_{max}$  channels. They can be converted to the disparity by *softmax* and weighted average. The predicted offset  $(\Delta x_i, \Delta y_i)$  might be two decimals and then the sampled values will be obtained by bilinear interpolation.  $N$  is a hyperparameter controlling the number of the reassembled neighbors. Intuitively, more points lead to better results but consume more memory.

Considering the contributions of the neighbors might be different, we further introduce the modulation scheme (Zhu et al. 2019). For each neighbor point, an additional modulation factor will be predicted to weight the sampled cost. Then Equation 2 can be rewritten as:

$$\mathbf{C}_r(x, y) = \frac{1}{\sum_i m_i} \sum_{i=1}^N m_i \cdot \mathbf{C}_0(x + \Delta x_i, y + \Delta y_i) \quad (3)$$

where  $m_i$  is the modulation factor of the  $i$ -th neighbor point, ranging from 0 to 1, output by a *sigmoid* layer.

In addition to the cost distribution, we can apply the dynamic reassembling scheme to the disparity map, as illustrated in Figure 2. Operating on the disparity, DSR is more similar to the disparity filling methods, formulated as:

$$D_r(x, y) = \frac{1}{\sum_i m_i} \sum_{i=1}^N m_i \cdot D_0(x + \Delta x_i, y + \Delta y_i) \quad (4)$$

where  $D_r$  and  $D_0$  are single-channel disparity maps.

Compared with DSR, the advantage of CSR is that it allows us to constrain the refined disparity distribution to be unimodal with the cross entropy loss (Tulyakov, Ivanov, and Fleuret 2018), which has been proved to be effective for the over-smoothing problem (Chen, Chen, and Cheng 2019). However, CSR requires more memory since for each assembled neighbor point, we have to store a vector rather than a single value. Although the burden brought by CSR is affordable in our model, we recommend DSR for the designing of practical algorithms.

## Loss Function

We use cross entropy loss and smooth  $L1$  loss (Chang and Chen 2018) to respectively supervise the disparity probability distributions and the final disparity results:

$$L_{ce}(\hat{P}(d), P(d)) = \frac{1}{I} \sum_{i=1}^I \sum_{d=0}^{d_{max}} -\hat{P}_i(d) \cdot \log P_i(d) \quad (5)$$

$$L_{sm}(\hat{D}, D) = \frac{1}{I} \sum_{i=1}^I \text{smooth}_{L1}(\hat{D}_i, D_i) \quad (6)$$

where  $\hat{P}(d)$  is the predicted distribution from *softmax* and  $P(d)$  is the ground truth distribution, a normalized Laplacian distribution centered at the disparity ground truth  $D$ .  $\hat{D}$  is the predicted disparity calculated by weighted average.  $I$  denotes the number of the pixels in the image.

To sum up, the objective function for training is:

$$L = \sum_{m=1}^M \lambda_m (L_{ce} + \mu L_{sm}) \quad (7)$$

where  $M$  is the number of the outputs of our whole model. For GwcNet, we supervise the results from the three hour-glass structures and the one from the refinement module, thus  $M$  is 4.  $\lambda$  and  $\mu$  are the balance weights.

## Experiments

### Datasets & Evaluation Metrics

The utilized datasets and the corresponding evaluation metrics are listed as follows:

**SceneFlow.** SceneFlow (Mayer et al. 2016) is a synthetic dataset containing 35454 training pairs and 4370 testing pairs, both with dense ground truth. We use **EPE** (End Point Error, calculated by  $\frac{1}{N} \sum_i |\hat{d}_i - d_i^*|$ ) and **1-pixel error** (percentage of the points with absolute error larger than 1 pixel) to evaluate the performance on SceneFlow.

**KITTI.** KITTI datasets consist of KITTI 2012 (Geiger, Lenz, and Urtasun 2012) and KITTI 2015 (Menze and Geiger 2015). The ground truth maps of the training images are sparse and the testing images are without ground truth. On KITTI 2012, there are 194 training pairs and 195 testing pairs. On KITTI 2015, there are 200 training pairs and 200 testing pairs, we split the training pairs into a training set (160) and a validation set (40). The used metrics on KITTI are **EPE** and **D1** (percentage of the outliers whose absolute error is larger than 3 pixels and 5% of the ground truth).

**Middlebury & ETH3D.** Middlebury 2014 (Scharstein et al. 2014) and ETH3D (Schöps et al. 2017) are two real-world datasets with less than 50 training pairs. Therefore, these two datasets are only used for generalization ability evaluation. **Bad2.0** (percentage of the points with absolute error larger than 2 pixel) and **Bad1.0** are the metrics on Middlebury and ETH3D respectively.

### Implementation Details

The framework is implemented with PyTorch. Adam ( $\beta_1 = 0.9, \beta_2 = 0.999$ ) is used as the optimizer. On SceneFlow, we

|     | Model       |            | SceneFlow   |             | KITTI 2015  |             |
|-----|-------------|------------|-------------|-------------|-------------|-------------|
|     | Feature     | Refinement | EPE (px)    | >1px (%)    | EPE (px)    | D1-all (%)  |
| (a) | CF          | <b>X</b>   | 1.04        | 7.68        | 0.68        | 1.65        |
| (b) | CF + LSP_SS | <b>X</b>   | 0.97        | 7.41        | 0.67        | 1.62        |
| (c) | CF + LSP_SL | <b>X</b>   | 0.99        | 7.55        | 0.67        | 1.68        |
| (d) | CF + LSP_F  | <b>X</b>   | 0.95        | 7.24        | 0.65        | 1.56        |
| (e) | CF          | ConvNet    | 0.84        | 8.05        | 0.63        | 1.72        |
| (f) | CF          | DSR        | 0.80        | 7.03        | 0.61        | 1.49        |
| (g) | CF          | CSR        | 0.79        | 6.70        | 0.61        | 1.35        |
| (h) | CF + LSP_F  | CSR        | <b>0.75</b> | <b>6.52</b> | <b>0.60</b> | <b>1.30</b> |

Table 1: Quantitative results of the ablation experiments on the SceneFlow test set and KITTI 2015 validation set.

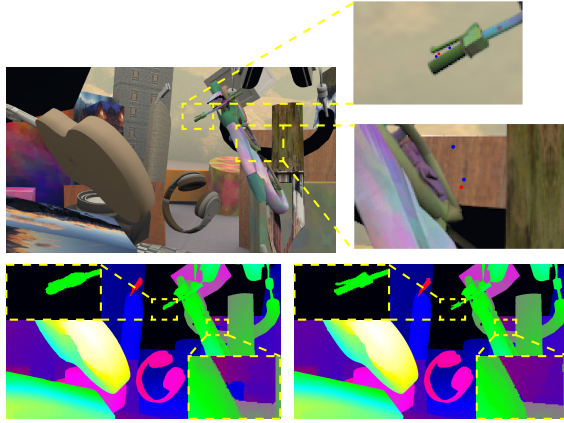


Figure 3: Illustration of the predicted locations in CSR. The red point denotes the central point and the blue ones denote the neighbor points. The disparity results before and after refinement are shown in the bottom.

| $N$        | 0    | 1    | 2    | 4    | 8    |
|------------|------|------|------|------|------|
| EPE (px)   | 0.95 | 0.80 | 0.75 | 0.74 | 0.72 |
| Memory (G) | 3.3  | 4.4  | 5.5  | 7.6  | 9.7  |

Table 2: Effects of the assembled point number  $N$  in CSR, results are evaluated on SceneFlow.

train the model for 10 epochs with a learning rate of 0.001. On KITTI, we finetune the model trained on SceneFlow for 300 epochs, with the learning rate of 0.001 for the first 200 epochs and 0.0001 for the remainings. For the model submitted to the KITTI benchmark, we use the total training pairs to finetune it for 600 epochs with the learning rate initialized as 0.001 and decayed by 10 times every 200 epochs. At the training phase, images are randomly cropped to  $512 \times 256$ , and the batch size is 4 on two TITAN Xp GPUs.

The maximum disparity is set as 192. In LSP, the neighbor window size is  $3 \times 3$ , and the multi-scale dilation rates are 1, 2, 4, 8. In the loss function,  $\mu$  is 0.1, the  $\lambda$  for the refined result is 1, and others are set as same as in the basic model.

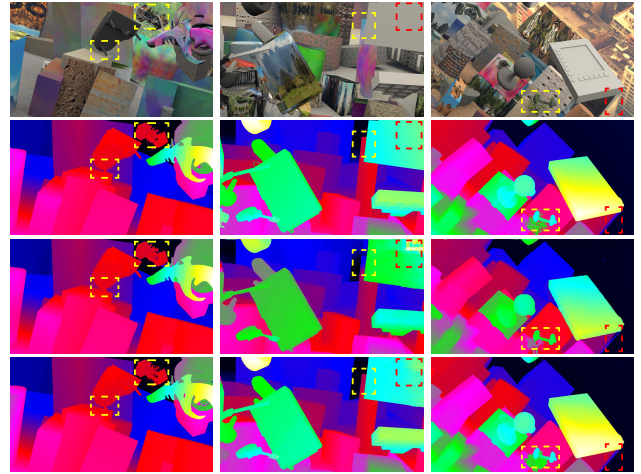


Figure 4: Qualitative results of the ablation experiments on SceneFlow. **From top to bottom:** input left images, ground truth disparity maps, disparity results from the basic model, and those from our whole model. Please zoom in for details.

## Ablation Studies

The ablation experiments are conducted on SceneFlow and KITTI 2015. GwcNet (Guo et al. 2019) with the cost volume formed by feature concatenation is adopted as the basic architecture. The evaluation results are listed in Table 1. Line (a) represents the baseline, where only CF is used to build the cost and no refinement method is utilized.

In lines (b) to (d), we study the effects of LSP and its multi-scale and multi-level settings. In the table, *SS* denotes single-scale, *SL* represents single-level, and *F* is the full setting. As indicated in the results, fusing LSP with the original CF all achieves a performance increase. We argue that the operation paradigm of convolution makes CF contain less structural information that lies in the pixel relationships. Therefore, LSP is a beneficial complement, especially with our multi-scale and multi-level schemes, which further enriches the structural information. In the following experiments, if there is no special instruction, all LSPs are equipped with the multi-scale and multi-level settings.



| Model    | PSMNet (2018) | SSPCVNet (2019) | GwcNet (2019) | AcfNet (2020b) | StereoDRNet (2019) |
|----------|---------------|-----------------|---------------|----------------|--------------------|
| EPE (px) | 1.09          | 0.98            | 0.98          | 0.87           | 0.86               |

---

| Model    | GANet (2019) | CSPN (2019) | LEAStereo (2020) | LaC + GwcNet (ours) | LaC + GANet (ours) |
|----------|--------------|-------------|------------------|---------------------|--------------------|
| EPE (px) | 0.80         | 0.78        | 0.78             | 0.75                | <b>0.72</b>        |

Table 3: Evaluation results of current stereo matching algorithms on the SceneFlow test set.

|     | Model          | Params. | FLOPs  | Memory |
|-----|----------------|---------|--------|--------|
| (a) | Baseline (B.)  | 6.88 M  | 1037 G | 3.28 G |
| (b) | B. + LSP       | 7.14 M  | 1054 G | 3.29 G |
| (c) | B. + LSP + DSR | 9.43 M  | 1149 G | 3.45 G |
| (d) | B. + LSP + CSR | 9.43 M  | 1150 G | 5.51 G |

Table 4: Complexity analysis of our methods. The resolution of the input images is  $960 \times 540$ .

In lines (e) to (g), we compare our DSR and CSR with an existing refinement method (Xu and Zhang 2020). Although EPE can be improved by predicting the disparity residual with a ConvNet, the percentage of the outliers increases as well. This is caused by the over-smoothing problem (Chen, Chen, and Cheng 2019). During convolution, an outlier could be corrected but an inlier might also be destroyed since the fixed kernels can not restore various error patterns. On the contrary, CSR and DSR propagate the accurate values in an adaptive manner and are not limited to the regular grid, making them more effective. An example is displayed in Figure 3, as we can see, the predicted neighbors are all belonging to the same semantical object as the central point. Further with the unimodal distribution constraint, CSR improves the *Outlier Percentage* well. In Table 2, we evaluate the effects of the assembled point number  $N$ . Finally considering the tradeoff between the accuracy and memory usage, we set  $N$  as 2.

In Table 1, line (h) shows the result of the whole model. Compared with the baseline, we achieve a significant performance increase by inserting the designed modules. Qualitative results on the SceneFlow test set are displayed in Figure 4. As illustrated in the dashed boxes, the disparity in details, textureless regions, and occluded areas are all improved.

### Complexity Analysis

Since stereo matching algorithms might be deployed to mobile devices, it is important to analyze the complexities. In this section, we record the increase of the parameter number, time consumption, and memory usage after inserting the designed modules. Considering the running time is also related to the used devices, we adopt FLOPs to indicate the time consumption. In this experiment, GwcNet is the basic model and the size of the input images is  $960 \times 540$ , i.e. the resolution of the SceneFlow images.

As shown in Table 4, inserting LSP brings little burden for the model. With regard to CSR, while the increases of the FLOPs and parameter number are not significant, it con-

|     | Model          | Middlebury<br>Bad2.0-noc (%) | ETH3D<br>Bad1.0-noc (%) |
|-----|----------------|------------------------------|-------------------------|
| (a) | Baseline(B.)   | 22.2                         | 9.7                     |
| (b) | B. + LSP       | 20.1                         | 9.6                     |
| (c) | B. + LSP + CSR | <b>18.7</b>                  | <b>9.2</b>              |

Table 5: Evaluation results of the generalization ability experiment. Models are trained on SceneFlow.

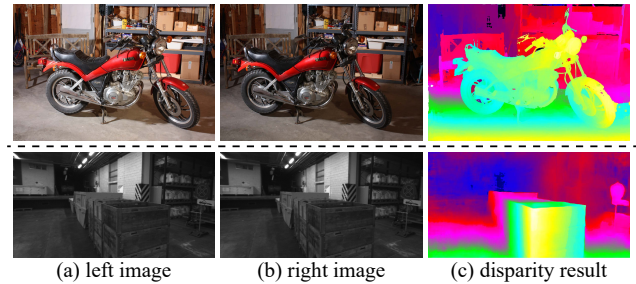


Figure 5: Visualization results of the generalization ability experiment. Images are from Middlebury and ETH3D.

sumes much memory. As discussed before, in CSR, we have to store a vector for each assembled point. On the contrary, the memory consumption of DSR is negligible, making it more suitable for practical algorithms.

### Generalization Ability

In this section, we study the influence of our modules on the generalization ability, with GwcNet as the baseline. Models are trained on SceneFlow and evaluated on the training sets of Middlebury and ETH3D. As presented in Table 5, the designed modules are less affected by the domain gap. On the one hand, CSR avoids leveraging the features of the matching network which are more susceptible than the input image (Zhang et al. 2020a). On the other hand, LSP is beneficial for the illumination inconsistent image pairs from Middlebury, as suggests in (Zabih and Woodfill 1994). Some visualization results of our whole model are displayed in Figure 5.

### Benchmarks

In this section, we compare our model with other SOTA algorithms on the SceneFlow dataset and KITTI benchmarks. To further validate the effectiveness, we also implement our modules onto GANet-deep (Zhang et al. 2019), which is a stronger baseline.

| Model               | 2015 All (%) |             |             | 2015 Noc (%) |             |             | 2012 All (%) |             | Time (s) |
|---------------------|--------------|-------------|-------------|--------------|-------------|-------------|--------------|-------------|----------|
|                     | D1-bg        | D1-fg       | D1-all      | D1-bg        | D1-fg       | D1-all      | Out-Noc      | Out-All     |          |
| PSMNet (2018)       | 1.86         | 4.62        | 2.32        | 1.71         | 4.31        | 2.14        | 1.49         | 1.89        | 0.41     |
| GwcNet-g (2019)     | 1.74         | 3.93        | 2.11        | 1.61         | 3.49        | 1.92        | 1.32         | 1.70        | 0.32     |
| AAANet+ (2020)      | 1.65         | 3.96        | 2.03        | 1.49         | 3.66        | 1.85        | 1.55         | 2.04        | 0.06     |
| AcfNet (2020b)      | 1.51         | 3.80        | 1.89        | 1.36         | 3.49        | 1.72        | 1.17         | 1.54        | 0.48     |
| CFNet (2021)        | 1.54         | 3.56        | 1.88        | 1.43         | 3.25        | 1.73        | 1.23         | 1.58        | 0.18     |
| GANet (2019)        | 1.48         | 3.46        | 1.81        | 1.60         | 3.11        | 1.63        | 1.19         | 1.60        | 1.80     |
| CSPN (2019)         | 1.51         | 2.88        | 1.74        | 1.40         | 2.67        | 1.61        | -            | -           | 1.00     |
| LEAStereo (2020)    | <b>1.40</b>  | 2.91        | <b>1.65</b> | 1.29         | 2.65        | 1.51        | 1.13         | 1.45        | 0.30     |
| LaC + GwcNet (ours) | 1.43         | 3.44        | 1.77        | 1.30         | 3.29        | 1.63        | 1.13         | 1.49        | 0.65     |
| LaC + GANet (ours)  | 1.44         | <b>2.83</b> | 1.67        | <b>1.26</b>  | <b>2.64</b> | <b>1.49</b> | <b>1.05</b>  | <b>1.42</b> | 1.72     |

Table 6: KITTI online leaderboards, sorted by the *D1-all* of all pixels on KITTI 2015.

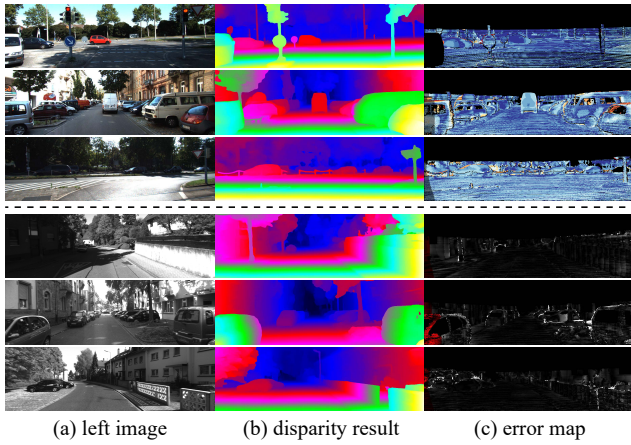


Figure 6: Visualization results of our method on the KITTI test sets. Images in the top three rows are from KITTI 2015 and those in the last three rows are from KITTI 2012.

**SceneFlow.** For a fair comparison, we only report the methods that share similar training strategies with us. The results are listed in Table 3, most of them are excerpted from the original papers. GwcNet and GANet are replicated by us with the official codes since there is no result consistent with our training strategy (GwcNet) or the model architecture setting (GANet) in their papers.

**KITTI.** The evaluation results on the KITTI benchmarks are provided by the online server. In Table 6, we present some superior methods that only involve the data of the stereo matching task. Several visualization results of our model based on GwcNet are displayed in Figure 6.

**Discussion.** Through inserting the designed modules (LaC, LSP and CSR), we surpass GwcNet by a significant margin on both benchmarks (**0.23px** on SceneFlow and **0.34%** on KITTI 2015). With GANet-deep as the baseline, the increase (**0.08px** on SceneFlow and **0.14%** on KITTI 2015) is not as substantial as GwcNet. In addition to the original performance of GANet is already superior, we an-

alyze the guided aggregation layers in GANet might play a similar role as CSR, i.e. propagating the matching cost based on the image content. Nevertheless, CSR holds the advantage of being able to establish long-range relationships in only one iteration. It is worth mentioning that CSPN (Cheng, Wang, and Yang 2019) and LEAStereo (Cheng et al. 2020) are not adopted in our work since the former model for the stereo matching task is not open-source and the submitted results of the latter can not be reproduced by us with the searched architecture.

In this paper, we mainly demonstrate the effectiveness of the designed modules in this paper. For real-time applications, we can adopt a faster basic network and incorporate the proposed modules, which will improve the performance with sacrificing a little speed, as shown in Table 4.

## Conclusions

In this paper, we have investigated some limitations of current CNN based stereo matching architectures. In the feature extraction module, aiming at tackling the issue that CF contains less structural information, we design a pairwise feature LSP, which can be combined with CF to enhance the feature representation for more accurate matching. In the disparity refinement module, to mitigate the problem that static convolutions tend to produce over-smooth results, we present an adaptive refinement scheme CSR. CSR is powerful and alleviates the over-smoothing problem well with the unimodal distribution constraint. The practical version of CSR, DSR is also introduced. Experimental results on the SceneFlow and KITTI benchmarks show that the proposed modules can improve the model performance by a significant margin. We confirm that injecting some traditional knowledge into deep networks would be beneficial and meaningful, as explored in previous researches (Zhang et al. 2019; Duggal et al. 2019).

Future works mainly focus on two aspects: 1) As a novel feature descriptor, LSP has more application scenes worth discovering, such as those with drastic illumination change (Zabih and Woodfill 1994). 2) DSR will be extended to other pixel-level tasks, e.g. semantic segmentation.

## References

- Chabra, R.; Straub, J.; Sweeney, C.; Newcombe, R.; and Fuchs, H. 2019. Stereodnnet: Dilated residual stereonet. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11786–11795.
- Chang, J.-R.; and Chen, Y.-S. 2018. Pyramid stereo matching network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5410–5418.
- Chen, C.; Chen, X.; and Cheng, H. 2019. On the over-smoothing problem of cnn based disparity estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8997–9005.
- Chen, Z.; Sun, X.; Wang, L.; Yu, Y.; and Huang, C. 2015. A deep visual correspondence embedding model for stereo matching costs. In *Proceedings of the IEEE International Conference on Computer Vision*, 972–980.
- Cheng, X.; Wang, P.; and Yang, R. 2019. Learning depth with convolutional spatial propagation network. *IEEE transactions on pattern analysis and machine intelligence*, 42(10): 2361–2379.
- Cheng, X.; Zhong, Y.; Harandi, M.; Dai, Y.; Chang, X.; Li, H.; Drummond, T.; and Ge, Z. 2020. Hierarchical Neural Architecture Search for Deep Stereo Matching. *Advances in Neural Information Processing Systems*, 33.
- Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; and Wei, Y. 2017. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, 764–773.
- Duggal, S.; Wang, S.; Ma, W.-C.; Hu, R.; and Urtasun, R. 2019. Deeppruner: Learning efficient stereo matching via differentiable patchmatch. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4384–4393.
- Gan, Y.; Xu, X.; Sun, W.; and Lin, L. 2018. Monocular depth estimation with affinity, vertical pooling, and label enhancement. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 224–239.
- Garg, D.; Wang, Y.; Hariharan, B.; Campbell, M.; Weinberger, K. Q.; and Chao, W.-L. 2020. Wasserstein distances for stereo disparity estimation. *arXiv preprint arXiv:2007.03085*.
- Geiger, A.; Lenz, P.; and Urtasun, R. 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 3354–3361. IEEE.
- Guo, X.; Yang, K.; Yang, W.; Wang, X.; and Li, H. 2019. Group-Wise Correlation Stereo Network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9): 1904–1916.
- Hu, H.; Zhang, Z.; Xie, Z.; and Lin, S. 2019. Local relation networks for image recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, 3464–3473.
- Huq, S.; Koschan, A.; and Abidi, M. 2013. Occlusion filling in stereo: Theory and experiments. *Computer Vision and Image Understanding*, 117(6): 688–704.
- Kendall, A.; Martirosyan, H.; Dasgupta, S.; Henry, P.; Kennedy, R.; Bachrach, A.; and Bry, A. 2017. End-to-end learning of geometry and context for deep stereo regression. In *Proceedings of the IEEE International Conference on Computer Vision*, 66–75.
- Khamis, S.; Fanello, S.; Rhemann, C.; Kowdle, A.; Valentin, J.; and Izadi, S. 2018. Stereonet: Guided hierarchical refinement for real-time edge-aware depth prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 573–590.
- Liang, Z.; Feng, Y.; Guo, Y.; Liu, H.; Chen, W.; Qiao, L.; Zhou, L.; and Zhang, J. 2018. Learning for disparity estimation through feature constancy. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2811–2820.
- Luo, W.; Schwing, A. G.; and Urtasun, R. 2016. Efficient deep learning for stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5695–5703.
- Mayer, N.; Ilg, E.; Hausser, P.; Fischer, P.; Cremers, D.; Dosovitskiy, A.; and Brox, T. 2016. 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*, 4040–4048.
- Mei, X.; Sun, X.; Zhou, M.; Jiao, S.; Wang, H.; and Zhang, X. 2011. On building an accurate stereo matching system on graphics hardware. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 467–474. IEEE.
- Menze, M.; and Geiger, A. 2015. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3061–3070.
- Ojala, T.; Pietikainen, M.; and Maenpaa, T. 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7): 971–987.
- Ramamonjisoa, M.; Du, Y.; and Lepetit, V. 2020. Predicting sharp and accurate occlusion boundaries in monocular depth estimation using displacement fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14648–14657.
- Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 234–241. Springer.
- Scharstein, D.; Hirschmüller, H.; Kitajima, Y.; Krathwohl, G.; Nešić, N.; Wang, X.; and Westling, P. 2014. High-resolution stereo datasets with subpixel-accurate ground truth. In *German conference on pattern recognition*, 31–42. Springer.
- Scharstein, D.; and Szeliski, R. 2002. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *International Journal of Computer Vision*, 47(1-3): 7–42.



- Schöps, T.; Schönberger, J. L.; Galliani, S.; Sattler, T.; Schindler, K.; Pollefeys, M.; and Geiger, A. 2017. A Multi-View Stereo Benchmark with High-Resolution Images and Multi-Camera Videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Shen, Z.; Dai, Y.; and Rao, Z. 2021. CFNet: Cascade and Fused Cost Volume for Robust Stereo Matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13906–13915.
- Song, X.; Zhao, X.; Fang, L.; Hu, H.; and Yu, Y. 2020. Edgestereo: An effective multi-task learning network for stereo matching and edge detection. *International Journal of Computer Vision*, 1–21.
- Tankovich, V.; Hane, C.; Zhang, Y.; Kowdle, A.; Fanello, S.; and Bouaziz, S. 2021. Hitnet: Hierarchical iterative tile refinement network for real-time stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14362–14372.
- Tosi, F.; Poggi, M.; and Mattoccia, S. 2019. Leveraging confident points for accurate depth refinement on embedded systems. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 0–0.
- Tulyakov, S.; Ivanov, A.; and Fleuret, F. 2018. Practical deep stereo (pds): Toward applications-friendly deep stereo matching. In *Advances in Neural Information Processing Systems*, 5871–5881.
- Wu, Z.; Wu, X.; Zhang, X.; Wang, S.; and Ju, L. 2019. Semantic Stereo Matching with Pyramid Cost Volumes. In *Proceedings of the IEEE International Conference on Computer Vision*, 7484–7493.
- Xu, H.; and Zhang, J. 2020. AANet: Adaptive Aggregation Network for Efficient Stereo Matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1959–1968.
- Yu, F.; and Koltun, V. 2015. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*.
- Zabih, R.; and Woodfill, J. 1994. Non-parametric local transforms for computing visual correspondence. In *European conference on computer vision*, 151–158. Springer.
- Zbontar, J.; and LeCun, Y. 2015. Computing the stereo matching cost with a convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1592–1599.
- Zhang, F.; Prisacariu, V.; Yang, R.; and Torr, P. H. 2019. GANet: Guided Aggregation Net for End-to-end Stereo Matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 185–194.
- Zhang, F.; Qi, X.; Yang, R.; Prisacariu, V.; Wah, B.; and Torr, P. 2020a. Domain-invariant stereo matching networks. In *European Conference on Computer Vision*, 420–439. Springer.
- Zhang, Y.; Chen, Y.; Bai, X.; Yu, S.; Yu, K.; Li, Z.; and Yang, K. 2020b. Adaptive unimodal cost volume filtering for deep stereo matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 12926–12934.
- Zhu, X.; Hu, H.; Lin, S.; and Dai, J. 2019. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9308–9316.