Explicit Context Mapping for Stereo Matching

Anonymous CVPR submission

Paper ID 0000

Abstract

With the success of deep neural networks, stereo matching has made significant progress recently. Transforming matching computation into a low-resolution space is a commonly used strategy to increase the matching accuracy with low computational cost, but it also brings the inevitable detail loss such as blur edges or even the disappearance of small objects. Therefore, there remains a crucial issue: how to map the low-resolution matching results back to the high-resolution space with an acceptable loss. In this paper, we propose a novel mapping method that explicitly models the local context between different resolutions for stereo matching. Since each low-resolution value inherently corresponds to a mapped region in the high-resolution space, we deem the local context as a descriptor of how much a low-resolution value could influence the mapped high-resolution values. As the influence can be represented by the similarities between correspondences, we present a novel learning-based similarity measure to model a more supportive context, which adds the location matrix as the additional geometrical information to the deep feature. For the mapping on the disparity map, besides the mapped region, we extend the influenced region to the four connected regions to keep more details. For the cost volume, the influenced regions are extended to six connected regions. Our explicit context mapping method reaches state-of-the-art performance on the SceneFlow and comparable results on the KITTI dataset.

1. Introduction

In the past decades, the computer vision community has paid many efforts to increase the accuracy of stereo matching [19, 25, 22, 29]. Currently, with the powerful deep neural networks, stereo matching has reached a new stage [28, 8, 2, 18], but the promotion of accuracy is built on a high computational burden. Transforming the matching into a low-resolution space can effectively increase the accuracy while requiring a low computational resource [8, 2], but it also brings the inevitable detail loss like blur edges or even the disappearance of small objects. Therefore, it is a crucial issue to appropriately map the low-resolution matching results back to the high-resolution space. In this paper, we propose the explicit context mapping method which can keep more details while still requiring a low computational resource.

The generic local context describes how much a pixel can influence other pixels among a specific region. Explicitly modeling the local context as the affinity matrix shows excellent performance on the depth completion and refinement task with the constant resolution [12, 3]. By contrast, the local context for the mapping of stereo matching describes how much a low-resolution value would influence the high-resolution values within its mapped region. Currently, there are two approaches to model this kind of local context: the traditional interpolation and the learning-based up-sampling. The interpolation approaches such as the nearest or bilinear model the local context by a simple linear relation [13, 27, 2]. It barely increases the computational resource, but when the scales become large, the linear relation cannot reliably reveal the true local context, which greatly impairs the accuracy. The learning-based up-sampling leverages the networks with sub-sampling and up-sampling units to implicitly model the local context [8, 2, 10]. It can maintain some details by the joint optimization, but the implicit modeling requires a large computational resource to extract the supportive context, which lowers the efficiency of low-resolution matching.

In this paper, we propose to explicitly model the local context as the similarities between the low-resolution values and their corresponding high-resolution values. We design a novel learning-based similarity measure to model a more supportive context, which fuses the deep feature with the location map to utilize the low-level geometrical information. With the obtained similarities, we can simply form the mapping matrices to compute the high-resolution values. However, the high-resolution value may get the wrong value if the similarities are too small, which means the low-resolution value cannot offer a reliable influence. To tackle this issue, we extend the influenced region to the four connected region when mapping on the disparity map. As

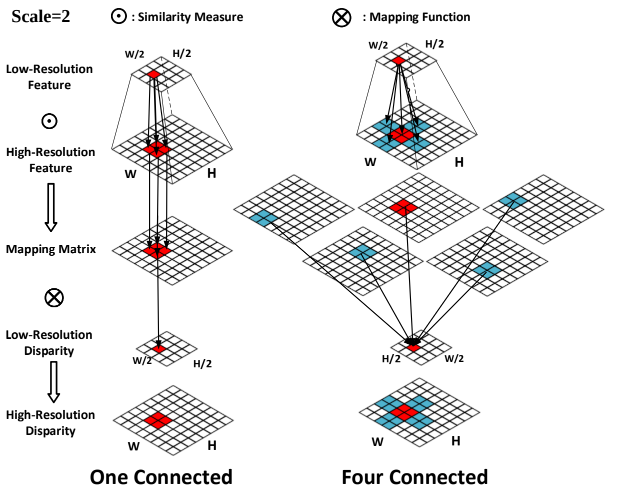

Figure 1. The explicit context mapping for disparity map. For the one connected mapping, each low-resolution value only influences high-resolution values in the mapped region (red), where the same color indicates correspondences. For the four connected mapping, each low-resolution value would influence not only the mapped region (red) but also the four connected regions (blue).

shown in Figure 1, each high-resolution value will be influenced not only by the mapped low-resolution value but also the additional four connected values. As for the cost volume, two additional regions along the depth dimension are also involved. To fuse influences from different low-resolution values, we design a fully differential aggregation function by multiplying similarities with their softmax value.

We also design a novel stereo matching network with the proposed mapping method. We adopt a siamese structure with pyramid units to provide the multi-scale feature. All sub-sampling units are applied before the cost computation to form low-resolution feature volume. After that, the cost computation can work efficiently in a low-resolution space and finally produce a low-resolution cost volume. At the same time, we can compute the mapping matrices as mentioned above with a very light similarity module composed of three 2D convolutional layers. With the obtain mapping matrices, we can choose to conduct the mapping on the cost volume or the disparity map, where the explicit context mapping method can be briefly used as the interpolation to replace all of the redundant up-sampling layers.

The explicit context mapping method reaches state-of-the-art performance on SceneFlow and comparable results on KITTI, which promotes our method can increase the accuracy with a low computational resource. The contribution of our work can be summarized in three folds.

- We propose a novel mapping method that explicitly models the local context between different resolutions for stereo matching. The local context is represented by the similarities computed from a novel learning-based similarity measure.
- We design a novel deep stereo matching network with the proposed mapping method and present the four connected strategy for the mapping on the disparity map and six connected strategy for the mapping on the cost volume.
- The explicit context mapping method reaches state-of-the-art performance on the SceneFLow. The explicit mapping is proven to be effective to hold back the details loss with a low computational resource.

2. Related Work

Stereo matching has made significant progress in the past few years owing to the powerful deep representation and joint optimization. Zbontar and LeCun [27] proposed the first deep stereo matching network, where they adopted convolutional layers to extract the deep representation and then computed the matching costs through the fully connected layers. Sooner, Luo et al. [13] proposed a more efficient pipeline by replacing the similarity measure with the inner product unit. At this stage, most deep stereo matching only had a trainable feature extractor while the post-processing like cross-based aggregation and conditional random field (CRF) were indispensable to ensure the accuracy [24, 29, 6]. Then Kendall et al. [8] proposed an end-to-end trainable deep stereo matching pipeline, GC-Net, which simultaneously modeled the context and geometry by the $3D$ convolutional auto-encoder. This architecture is capable of obtaining the multi-scale feature and jointly realizing the cost aggregation by the end-to-end training. Later, the pyramid spatial pooling unit is used to strengthen the representation by incorporating the nearly global context into the feature [2]. Although the multi-scale feature is beneficial for the accuracy, it also suffers from the loss the details. Yu et al. [26] brought the low-level RGB feature to enhance the details during cost aggregation while Liang et al. [10] used the feature constancy as supervision to promote the refinement. Song et al. [20] proposed to use the edges as additional information to represent the local context and Yang et al. [23] brought into the prior knowledge from the segmentation results to regularize the loss function. These approaches can resolve the loss of details, but they require a significant computational resource. Currently, Khamis et al. [9] proposed the StereoNet which added a hierarchical refinement on the GC-Net. The light and efficient refinement module could ensure the accuracy even transforming the matching into very low resolution. Our explicit context mapping method serves as a general mapping module which can both serve for the cost volume and the disparity map. It fuses the high-level multi-scale feature with low-level geometrical information to model the local context. In addition, the computation of the local context barely increases the computational resource.

The two mostly related works to the explicit modeling of local context are Liu et al. [12] and Cheng et al.[3], who proposed to model the local context as the affinity matrix. The affinity matrix describes how close two points are in a space, which shows excellent performance for the depth refinement and completion on the constant resolution. By contrast, the local context for the mapping of stereo matching reveals the relations between different resolutions. Compared with the common super-resolution task which only gets the information of low-resolution, we get both the high-resolution and low-resolution information. So we can directly determine the mapped regions by the scale size and then compute the similarities between low-resolution values and high-resolution values as mapping matrices.

3. Explicit Context Mapping

We present the explicit context mapping method for stereo matching. The requirements of the mapping method are discussed in Section 3.1, where we theoretically formulate the mapping process as the computation of local context. Then we introduce three ways to use the local context in Section 3.2. Finally, in Section 3.3, we construct a deep neural network for stereo matching with the proposed mapping method and integrate the network with the three strategies.

3.1. Problem Formulation

3.1.1 Mapping on Feature Volume

For stereo matching, the mapping from the low-resolution to the high-resolution is realized on the $4D$ feature volume, $3D$ cost volume and $2D$ disparity map. The $4D$ feature volume is the concatenated feature map of matching candidates

$$V(x,y,d) = F_L(x,y) \oplus F_R(x-d,y), \tag{1}$$

where F_L and F_R is the feature map of the left image L and right image R, $\oplus$ is the concatenation operation [1]. $V(x,y,d)$ is the feature representing the matching cost between the pixel $L(x,y)$and $R(x-d,y)$. The mapping on the $4D$ feature volume is always implicitly realized by the de-convolutional layers with the skip connection, which can be deemed as the fusion of multi-scale features as

$$V^h(p) = \frac{\sum_{i=1}^{k^3} \omega_i V^l(p_i)}{\sum_{i=1}^{k^3} \omega_i} + V^{h'}(p). \tag{2}$$

Here, V^h is the high-resolution feature volume, V^l is the low-resolution feature volume, $V^{h'}$ is the former high-resolution feature from skip connection, p represents the postion (x,y,d), p_i is related pixels determined by the kernel size k of de-convolutional layers, and ω is the network parameter representing the similarity between $V^l(p_i)$ and $V_h(p)$. The $4D$ mapping can maintain the details by the implicit realization of the cost aggregation, but the computation and usage of ω are a total black box. Besides, any operation on the $4D$ volume will cause an unaffordable computational resource to store the $4D$ volume. Therefore, in this paper, we focus on the mapping on the cost volume and disparity map.

3.1.2 Mapping on Cost Volume

The computation of cost volume is denoted as

$$C(x,y,d) = f(F_L(x,y), F_R(x-d,y)), \tag{3}$$

where C is the matching cost and $f(\cdot)$ is the cost computation function. Compared with the feature volume, each item $C(x,y,d)$ directly represents the matching cost between $L(x,y)$ and $R(x-d,y)$. So the mapping for the cost volume is expressed as

$$C^h(x,y,d) = \omega \times C^l(x_l,y_l,d_l). \tag{4}$$

We can directly determine the corresponding low-resolution value $C^l(x_l,y_l,d_l)$ by the scale size s as $x_l = \lfloor x \div 2^s \rfloor$, where $\lfloor \cdot \rfloor$ is the exact division. The problem for this mapping is how to compute the ω which denotes how much the low-resolution value will influence the high-resolution values. We regard the low-resolution cost $C^l(x_l,y_l,d_l)$ as the regional cost on behalf of s^3 high-resolution cost values. So mapping is treated as computing simialrities between the regional feature of the low-resolution cost and the pixel-wise feature of high-resolution costs

$$\begin{aligned} \omega &= m(V^h(x,y,d), V^l(x_l,y_l,d_l)) \\ &= m(F_L^h(x,y) \oplus F_R^h(x-d,y), F_L^l(x_l,y_l) \oplus F_R^l(x_l-d_l,y_l)) \end{aligned}, \tag{5}$$

where the $m(\cdot)$ is the similarity measure. However, constructing the high-resolution feature volume V^h will take too much resource, so we simplify the matching process by using the feature map F to represent matching feature V. This operation is realized by a two-steps mapping on the $x-y$ dimension and d dimension separately. When mapping on the $x-y$ dimension, we use the $F_R^l(x_l-d_l,y_l)$ to represent the $F_R^h(x-d,y)$, which means that the pixel-to-pixel matching between $L(x,y)$ and $R(x-d,y)$ is represented by the pixel-to-region matching, then the Equation 5 is written as

$$\begin{aligned} \omega &= m(F_L^h(x,y) \oplus F_R^l(x_l-d_l,y_l), F_L^l(x_l,y_l) \oplus F_R^l(x_l-d_l,y_l)) \\ &= m(F_L^h(x,y), F_L^l(x_l,y_l)) \end{aligned}. \tag{6}$$

[1] The cost volume and disparity are computed for the left image.

As for the mapping on the d dimension, we use the $F_L^l(x_l, y_l)$ to represent $F_L^h(x, y)$ as

$$\begin{aligned}\omega &= m(F_L^l(x_l, y_l) \oplus F_R^h(x-d, y), F_L^l(x_l, y_l) \oplus F_R^l(x_l-d_l, y_l)) \\ &= m(F_R^h(x-d, y), F_R^l(x_l-d_l, y_l))\end{aligned}. \tag{7}$$

3.1.3 Mapping on Disparity Map

The disparity map D indicates the pixel-wise matching between $L(x, y)$ and $R(x-d, y)$. The mapping on the disparity map is given by

$$D^h(x, y) = \omega \times D^l(x_l, y_l) \times 2^s. \tag{8}$$

The ω represents the similarities between the low-resolution pixel $L^l(x_l, y_l)$ and the high-resolution pixel $L^h(x, y)$. The computation of the ω can be directly obtained by

$$\omega = m(F_L^h(x, y), F_L^l(x_l, y_l)) \tag{9}$$

3.2. Computation of the Local Context

In this section, we introduce the computation of the local context. We propose a novel similarity measure to fuse the deep feature with the low-level geometrical information. We present three ways to model the local context from the computed similarities. For the basic one connected mapping, each high-resolution value will only be influenced by one low-resolution values. We extend the influenced regions to the four connected region for the disparity map and six connected region for the cost volume. A fully differential aggregation function is designed to fuse the influences from different low-resolution values, which can keep more details and offer a better smoothness.

3.2.1 Learning-based Similarity Measure

We design a novel similarity measure to fuse the high-level deep feature and low-level geometrical map. The high-resolution and low-resolution deep feature is directly obtained from the network. We first expand the low-resolution feature as the same size of high-resolution feature by $F^e(x, y) = F^l(x_l, y_l)$, where the F^e is the expanded feature map. Then we concatenate the expanded feature map with the high-resolution feature map to form the fused deep representation $F^f = F^h \oplus F^e$. Despite the deep representation, we also add location map as the geometrical information. Compared with the work of Novotny et al.[17], we design a more elaborate location map by using the distance matrix G. The distance matrix denotes how far the pixels are to their mapping center, where with the scale size s, the distance matrix for each region is computed as

$$\begin{pmatrix} -\frac{2^s}{2} & \cdots & -1 & 1 & \cdots & \frac{2^s}{2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -\frac{2^s}{2} & \cdots & -1 & 1 & \cdots & \frac{2^s}{2} \end{pmatrix}. \tag{10}$$

Then we concatenate G with G^T to form the two channel distance map as the geometrical representation F^g. Finally, we adopt three convolutional layers as $m(\cdot)$ to compute the similarity as

$$M = m(F^f \oplus F^g). \tag{11}$$

3.2.2 One Connected Mapping

For the basic one connected mapping, each high-resolution value $C^h(x, y)$ or $D^h(x, y)$ will only be influenced by its corresponding low-resolution value $C^l(x_l, y_l)$ or $D^l(x_l, y_l)$, where $x_l = \lfloor x \div 2^s \rfloor$ with the scale size s. For the mapping on the disparity map, we can directly compute the similarity matrix on the left image as Equation 11 and then expand the low-resolution disparity map into the expanded high-resolution map D^e. The high-resolution disparity is computed as

$$D^h(x, y) = D^e(x, y) \times M(x, y) \times 2^s. \tag{12}$$

The mapping on the cost volume is realized by two steps. We first expand the low-resolution cost along the x, y dimensions to get C^e. Then we use the representation computed from the left image to form the similarity matrix as Equation 6 and compute the high-resolution volume as

$$C^e(x, y, d_l) = C^e(x, y, d_l) \times M(x, y). \tag{13}$$

After that, we expand the C^e along the d dimension and compute the similarity matrix by the representation from the right image as Equation 7. The final high-resolution cost volume is computed by

$$C^h(x, y, d) = C^e(x, y, d_l) \times M(x-d, y). \tag{14}$$

The one connected mapping can compute the high-resolution values by the influence from low-resolution values. However, if the similarity is minimal which means the low-resolution values weakly influence the high-resolution values, how can we compute the high-resolution values? Here, we tackle this problem by extending the influenced region of each low-resolution value, in other words, each high-resolution value will be influenced by more than one low-resolution value. The extension of regions can not only solve the above problem but also realize the local aggregation to offer better performance.

3.2.3 Four Connected Mapping

For the mapping on the disparity map, we extend the influenced region to the four connected region. Besides the Equation 9 to compute the ω_1, we compute four additional similarities as

$$\begin{aligned}\omega_2 &= m(F_L^h(x, y), F_L^l(x_l-1, y_l)) \\ \omega_3 &= m(F_L^h(x, y), F_L^l(x_l+1, y_l)) \\ \omega_4 &= m(F_L^h(x, y), F_L^l(x_l, y_l-1)) \\ \omega_5 &= m(F_L^h(x, y), F_L^l(x_l, y_l+1))\end{aligned} \tag{15}$$

The final high-resolution disparity is computed by fusing the five similarities. We design a fully differential fusion function using the softmax weights $\sigma(\cdot)$. The mapping to the high-resolution space is expressed as

$$\begin{aligned} D^h(x,y) = D^l(x_l,y_l) \times \omega_1 \times \sigma(\omega_1) \times 2^s \\ + D^l(x_l-1,y_l) \times \omega_2 \times \sigma(\omega_2) \times 2^s \\ + D^l(x_l+1,y_l) \times \omega_3 \times \sigma(\omega_3) \times 2^s. \\ + D^l(x_l,y_l-1) \times \omega_4 \times \sigma(\omega_4) \times 2^s \\ + D^l(x_l,y_l+1) \times \omega_5 \times \sigma(\omega_5) \times 2^s \end{aligned} \tag{16}$$

3.2.4 Six Connected Mapping

For the mapping on the cost volume, besides the four connected regions, we extend two additional regions along the depth dimension. The computation of similarities is expressed as

$$\begin{aligned} \omega_6 = m(F_R^h(x,y), F_R^l(x_l-d_l-1,y_l)) \\ \omega_7 = m(F_R^h(x,y), F_R^l(x_l-d_l+1,y_l)) \end{aligned}. \tag{17}$$

We first map along the x, y dimensions by four connected mapping as Equations 13 and 16 to compute C^e. Then the high-resolution cost volume is obtained by

$$\begin{aligned} C^h(x,y,d) = C^e(x,y,d_l-1) \times \omega_6 \times \sigma(\omega_6) \\ + C^e(x,y,d_l+1) \times \omega_7 \times \sigma(\omega_7) \end{aligned}. \tag{18}$$

Both the Equations 16 and 18 can be efficiently conducted by the broadcast matrix multiplication as Equation 12 shows.

3.3. Deep Stereo Matching with Explicit Mapping

We design a novel stereo matching network which realizes the proposed mapping method on the cost volume and the disparity map. The network uses the pyramid stereo matching network (PSM) [2] as the baseline. Instead of sub-sampling the images at the first layer of the network, we add four convolutional layers to extract the full resolution feature map. As shown in Table 1, the first sub-sampling is employed at layer 5 with stride 2. For the scale $s = 2$, the second sub-sampling is employed at layer 9 as PSM. As for the scale $s = 3$, the second and third sub-sampling are employed at layer 6 and 9, while for the scale $s = 4$, the sub-sampling operations are employed at $6, 9$ and 24 layers. We adopt a standard pyramid pooling unit with the averaging pooling size of $64\times64, 32\times32, 16\times16, 8\times8$. Then layer 34 outputs the fused multi-scale feature with $1/2^s$ resolution. After that, we can combine the multi-scale deep feature with out location map and compute the mapping matrix as discussed in Section 3.2.1. The multi-scale features from the left and right images are used to construct the feature volume as shown in Section 3.1.1. We adopt the same 3D convolutional layers and three stacked hourglass modules as the PSM to compute the cost volume. Finally, we can choose to use the proposed six connected mapping method to map the $1/2^s$ resolution cost volume into full-resolution cost volume and use the standard soft-argmin function [8, 26, 2] to regress the disparity map. Or, we can directly use the soft-argmin function to compute the disparity map with $1/2^s$ resolution and then employ four connected mapping to get the full-resolution disparity map.

Table 1. Network Setting

unit	**index**	**stride**	**input**	**output**
Pyramid Feature Extraction				
Conv	1-4	1	3	32
Conv	5-6	2	32	32
Residual	6-8	1	32	32
Residual	9-24	2	64	64
Residual	24-26	1	128	128
Residual	26-28	1	128	128
Spatial Pooling Size: $64\times64, 32\times32, 16\times16, 8\times8$				
SPP	29-32	1	128	128
concatenate output of 24,28 29,30,31,32				
Conv	33-34	1	320	32
Computation of Local Context				
take output of 4 and 34				
Conv	35-38	1	66	1
Feature Volume				
concatenate left and shifted right feature				
3D CNN				
3dConv	39-42	1	32	32
hourglass	43-48	1	32	32
cost1: add 48 and 42				
hourglass	49-54	1	32	32
cost2: add 54 and 42				
hourglass	55-60	1	32	32
cost3: add 60 and 42				
Cost Volume				
3dConv on cost1	61	1	32	1
3dConv on cost2	62	1	32	1
3dConv on cost3	63	1	32	1
3D mapping	cost1: six connected mapping on output of 61			
3D mapping	cost2: six connected mapping on output of 62			
3D mapping	cost3: six connected mapping on output of 63			
Disparity map				
D1: soft-argmin on cost1				
D2: soft-argmin on cost2				
D3: soft-argmin on cost3				
2D mapping	D1: four connected mapping on D1			
2D mapping	D2: four connected mapping on D2			
2D mapping	D3: four connected mapping on D3			

3.3.1 Implementation Details

Except for the $61, 62, 63$ layers, all of the convolutional layers are followed with the GroupNorm [21] and ReLu [14]. The reason we choose the GroupNorm rather than the BatchNorm [7] is to keep the fairness between different scale sizes. When the scale size is 2, the batchsize is hard to

Table 2. The Ablation on FlyingThings3D. EPE: End-point-error

ID	Hourglass	Scale			On Disparity Map		On Cost Volume			EPE	EPE Non-Occluded	Memory(M)	Time(s)
		2	3	4	One Connection	Four Connection	One Connected	Four Connected	Six Connected				
BaseLine1	✓	✓								0.8202	0.6970	5805	0.68
Exp1	✓	✓			✓					1.2510	1.1700	2931	0.61
Exp2	✓	✓				✓				1.1138	1.0257	3303	0.65
Exp3	✓	✓					✓			0.7564	0.6162	6533	0.72
Exp4	✓	✓						✓		0.7012	0.6056	6935	0.81
Exp5	✓	✓							✓	0.6587	0.5575	8264	0.83
Exp6		✓							✓	0.7367	0.5850	6781	0.58
BaseLine2	✓		✓							0.9848	0.8809	3479	0.21
Exp7	✓		✓			✓				1.4749	1.4175	2229	0.19
Exp8	✓		✓						✓	0.9599	0.8695	6381	0.44
Exp9			✓						✓	0.9046	0.8060	5477	0.25
BaseLine3	✓			✓						1.9162	1.8085	3702	0.16
Exp10	✓			✓		✓				2.2570	2.1496	2161	0.12
Exp11	✓			✓					✓	1.6858	1.5815	6085	0.36
Exp12				✓					✓	1.6405	1.5482	5343	0.20

set more than 8 on a single GPU, which is shown to have a big impair of accuracy [21]. So we use group normalization for all of the networks with different scales to remove the influence of different normalization units. Besides, the average pooling size for the spatial pyramind pooling module always ensures the largest kernel can obtain the global view by pooling the feature map as 1×1 size. So we modify the pooling size as $32 \times 32, 16 \times 16, 8 \times 8, 4 \times 4$ for the scale 3 while $16 \times 16, 8 \times 8, 4 \times 4, 2 \times 2$ for the scale 4. We use the standard Smooth L_1 as Fast-RCNN and PSM [2, 4] to train the whole network.

4. Experiement

In this section, we show the evaluation results of our method on SceneFLow [15] and KITTI 2015 [16]. All networks are implemented with PyTorch and optimized by a standard Adam optimizer with $\beta_1 = 0.9, \beta_2 = 0.999$. The inputs to the network are preprocessed using the color normalization with standard parameters on ImageNet as $mean = (0.485, 0.456.0.406), std = (0.229, 0.224, 0.225)$. During training, we randomly crop the image with size of $(256, 512)$ and set the maximum disparity as 192. The batch size is set as 4 for all of the training on four Nvidia 1080Ti GPUs (each of 1).

4.1. Ablation Studies on SceneFlow

SceneFlow is a large synthetic dataset containing 34896 training images and 4248 testing images with size of 540×960. This dataset has three rendered sub-dataset: FlyingThings3D, Monkaa and Driving. FlyingThings3D is rendered from ShapeNet [1] dataset and has 21828 training data and 4248 testing data. Monkaa is rendered from the animated film Monkaa and has 8666 training data. The Driving is constructed by the naturalistic, dynamic street scene from the viewpoint of a driving car and has 4402 training samples. Here, we use the FlyingThings3D as the training and testing dataset for the ablation of the explicit mapping method with different network settings and different scales.

All of the baseline models are first trained for 7 epochs. Then both comparison models and baseline models are initialized from the 8 epochs pre-trained models and trained for other 7 epochs. The baseline models are built based on the PSM with different scales. We add five additional layers to extract the full-resolution feature and replace the BatchNorm with the standard GroupNorm $(group = 32)$. The quantity results are shown in Table 2.

4.1.1 Acuracy and Efficiency Analysis

For the scale 2, BaseLine1 is the standard PSM. The Exp1 and Exp2 directly regress the low-resolution disparity map and use the context mapping to get the high-resolution disparity map. The explicit context mapping reduces more than 2500M memory, which comes from the 3D convolutional layers on the full-resolution cost volume. Using only 57% memory, the EPE error is increased by 0.43 and 0.29 in contrast with the EPE on the non-occluded pixels reducing by 0.03. The reason for this result is that the context mapping relies on the correct low-resolution values. But for the occluded pixels, the matching results are not reliable at all, so the mapped values will also get wrong values, which increases the EPE. While for the reliable non-occluded pixels, the mapping offers a further promotion. We try to train the four connected mapping on the disparity for other 11 epochs and find out the EPE error comes down to 0.7614 and 0.6200, which means the context mapping for the disparity map ensures the BaseLine accuracy, but it is much harder to learn. The Exp2 has a 0.14 promotion on EPE from the Exp1, which shows that the four connected mapping offers a better smoothness on the occluded pixels.

The Exp3, 4, 5 remarkably reduce the EPE error both on all pixels and non-occluded pixels. The additional 700M memory between Exp3 and BaseLine1 comes from storing the three expanded full-resolution cost volume each of which is 200M and 100M for the three convolutional layers of the similarity measure. The additional 400M memory of the Exp4 comes from the computation of the other four mapping matrix. We see the computation of local context costs about 100M for each mapping matrix. This only increases the storage without the training parameters since the parameters are shared for all similarity computation. The increase of memory 1329M of Exp5 comes from the mapping along the depth dimension, since we expand the mapping matrix as the same size of cost volume which cannot be automatically freed during training. Comparing the EPE error of Exp3, 4, 5, we see the extension of connected regions obviously increases the accuracy. The EPE of Exp4, 5 on the non-occluded pixels shows that the extension along the depth dimension can effectively map the right matching results, where the 0.5 promotion of EPE mainly comes from the non-occluded pixels. Comparing the EPE and Memory of Exp2 and Exp4, we see the context mapping on the cost volume offers better performance with 0.55 promotion on accuracy while the mapping on the disparity map offers a better efficiency with $3230M$ decrement of memory.

For the scale 3, we build BaseLine2 with sub-sampled 8 times PSM. Comparing BaseLine2 and BaseLine1, we see the sub-sampling reduces 2326M about 40% memory with the weakness of EPE 0.1646. The sub-sampling also offers a 58% promotion on speed, where we only need 0.2 s to run the whole network. The Exp7 shows the four connected mapping on disparity promotes 35% on the memory, but the 48% higher EPE impairs the promotion on the efficiency. We also train the Exp7 model with additional 11 epochs and find out the final accuracy reaches 1.0016 and 0.8790. Comparing Exp8 and BaseLine2, we find the 0.08 promotion also mainly comes from the non-occluded pixels.

For the scale 4, we see the networks get a distinct weakness on EPE with an obvious increase of speed. We can run the whole network with almost 0.12s. The 17% weakness of the four connected mapping with the BaseLine3 is not as large as the experiments on the low scales. We also prolong the training to 11 epochs and finds the EPE comes to 1.832 and 1.8286. From the prolonged training of Exp2, 7, 10, we conclude that it is indeed harder to train the four connected mapping of disparity map than the mapping on the cost volume. But with the additional training, the four connected mapping on disparity at least maintain the accuracy with more than 40% lower computational resource and 0.03s promotion on speed. Comparing the promotion on EPE from Exp11, BaseLine3, Exp9, BaseLine2 and Exp5, BaseLine1, we see the six connected mapping on the cost volume effectively increase the accuracy. The promotion of explicit mapping gets lower when the scale becomes, which is sensible because the mapping needs the right low-resolution values. And when the scale becomes larger, the low-resolution matching results are not very reliable, so the effectiveness of mapping is impaired.

We also evaluate how much the hourglass contributes to stereo matching networks. From Exp5 and Exp6, we see the hourglass structure offers a promotion of 0.08 EPE of all pixels and 0.03 EPE of non-occluded pixels, where the promotion mainly comes from the occluded pixels. But for the scale 3 and 4, the hourglass even lowers the accuracy from the Exp8, 9 and Exp 11, 12. The reason is that the hourglass structure can correct the wrong matching results when wrong matching has a small account. However, when the scale becomes larger, the account of wrong matching results becomes larger. The hourglass module cannot modify most of them but accumulating the error by the addition operation. This also proves that the promotion of our mapping method relies on a reliable low-resolution matching. It indeed promotes the baseline with any kinds of low-resolution matching results but for a better baselien, the promotion is higher.

It is also noting that, although the six connected mapping takes an obvious large memory, we only need a constant memory 700M to compute the mapping matrices. The other memory is not caused by the mapping method itself. Theoretically, after computing the mapping matrices, the mapping can directly be conducted by the broadcast matrix multiplication which has the complexity of $O(1)$ on GPU. The increasing of time and memory mainly comes from the expanding realization. We also build an efficient approach for memory by replacing the expandation with the circulative mapping, where the memory is reduced about 800M than the Baseline models but with an increase of time 0.84s. This comes from the unsupportive built-in multi-process module. Our method could be definitely optimized to have the same memory and speed as the built-in bilinear and trilinear interpolation in PyTorch with the basic Cuda realization.

4.1.2 Qualitaty Analysis

We visualize the results on the FLyingThings3D on the Figure 2 with the four connected mapping on the disparity map and six connected mapping on the cost volume. We observe that the six mapping method preserves most of the details while the four connected mapping will lose some details at the edges. Along with the increasing of the scale, the loss of details is inevitable but the explicit context mapping effectively holds back the loss.

4.2. BenchMark Results

4.2.1 SceneFlow

We use the pre-trained model on the FlyingThings3D and finetune on the whole dataset for 8 epochs. The results are shown in Table 3. Our method reaches the best performance on end-to-point error. For the sub-sampled 4 times methods [2, 8], we get a promotion more than 0.4. Compared with the StereoNet 8x [9], we get a 0.17 promotion. But

Table 3. Comparison on SceneFLow dataset

	PSM[2]	CRL[18]	DispNetC [5]	GC-Net [8]	StereoNet 8x[9]	StereoNet 16x [9]	Edge Stereo[20]	CSPN[3]	Our x4	Our x8	Our x16
EPE	**1.09**	**1.32**	**1.68**	**2.51**	**1.101**	**1.525**	**4.12**	**0.78**	**0.6755**	**0.9239**	**1.7043**

Table 4. Comparisons on KITTI2015

Model	All pixels			Non-Occluded Pixels			Time(s)
	D1-bg	D1-fg	D1-all	D1-bg	D1-fg	D1-all	
GC-Net[8]	2.21	6.16	2.87	2.02	5.58	2.61	0.9
MC-CNN[27]	2.89	8.88	3.89	2.48	7.64	3.33	67
Displetv v2[5]	3.00	5.56	3.43	2.73	4.95	3.09	265
PSMNet[2]	1.86	4.62	2.32	1.71	4.31	2.14	0.41
EdgeStereo[20]	2.27	4.18	2.59	2.12	3.85	2.40	0.27
StereoNet[9]	4.30	7.45	4.83	4.05	6.44	4.44	0.02
CSPN[3]	1.51	2.88	1.74	1.40	2.67	1.61	0.5
iResNet-i2e2[11]	2.10	3.64	2.36	1.94	2.55	2.15	0.1
Our Four Connected	2.59	5.85	3.13	2.10	4.99	2.58	0.61
Our Six Connected	2.91	7.05	3.60	2.52	5.88	3.08	0.83

for StereoNet 16x, our method is lower with 0.215. The implicit mapping method seems offer a better smoothness when the scale becomes large. But from the illustration, the explicit shows better performance on the details even the EPE is higher.

4.2.2 KITTI2015

KITTI 2015 is a real-world dataset with street views from a driving car. It contains 200 training stereo image pairs with sparse ground-truth disparities obtained using LiDAR and another 200 testing image pairs without ground-truth disparities. We split the training data as 160 images for training and 40 images for evaluation. We use the pre-trained model from SceneFlow and fine-tune the model on KITTI2015 for 300 epochs. The comparison with other state-of-the-art methods is shown in Table 4. We actually get a lower evaluation error 1.76% than the PSM on the training data. However, for the testing data, the evaluation is all below the PSM. Different from the results on SceneFlow, the four connected mapping even get better performance than the six connected mapping. This result may be caused by the sparse form of the training data while our mapping method needs a dense ground truth for training. The mapping carries out as the determination of the pixel-to-pixel relations. When the ground truth has the sparse shape, this kind relation is not clear enough to learn. For the four connected mapping, the ambiguities of the ground truth make the learning easier than the six connected mapping. But due to our explicit model of the pixel-to-pixel relations, the sparse ground truth is not supportive to reach state-of-the-art performance.

5. Conclusions

In this paper, we have presented the explicit context mapping for stereo matching. The local context can be modeled by the similarities between low-resolution values and their influenced high-resolution values. The proposed learning-based similarity measure can effectively fuse the low-level geometrical information to the high-level deep feature using the additional distance matrix. The four connected mapping on the disparity shows a great promotion on efficiency while the six connected mapping on cost volume can remarkably increase the accuracy. The stereo matching network with our mapping method reached state-of-the-art performance on the SceneFlow and comparable results on the KITTI 2015 with a low computational resource.

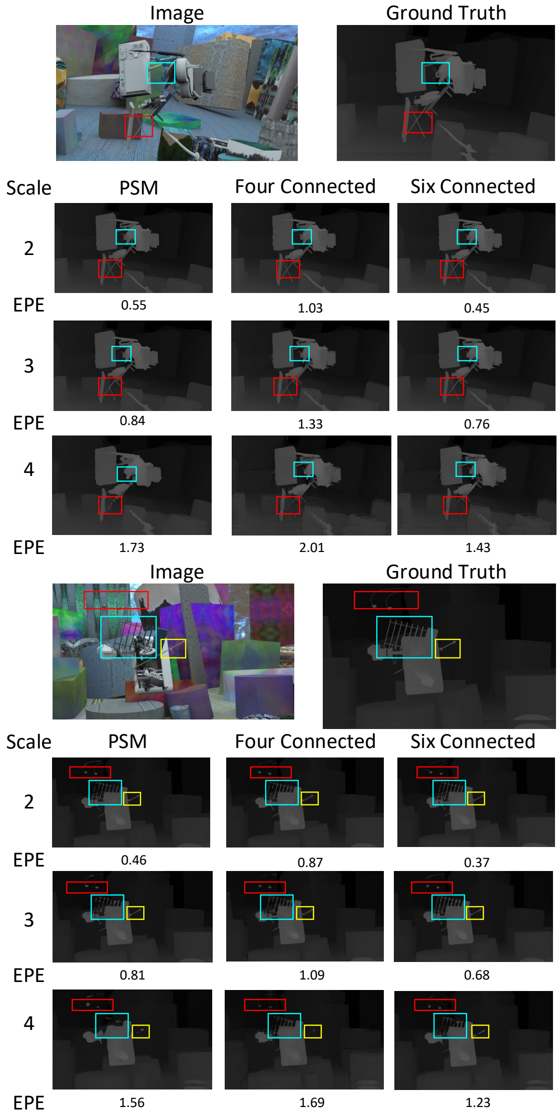

Figure 2. Qualitative analysis on details of the four connected mapping on disparity map and six connected mapping on the cost volume. In each colored box, we observe that the explicit context mapping keeps better details for all scales. This figure better looks with a zoom-in view.

References

[1] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 6

[2] J.-R. Chang and Y.-S. Chen. Pyramid stereo matching network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5410–5418, 2018. 1, 2, 5, 6, 7, 8

[3] X. Cheng, P. Wang, and R. Yang. Learning depth with convolutional spatial propagation network. *arXiv preprint arXiv:1810.02695*, 2018. 1, 3, 8

[4] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 6

[5] F. Guney and A. Geiger. Displets: Resolving stereo ambiguities using object knowledge. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4165–4175, 2015. 8

[6] H. Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on pattern analysis and machine intelligence*, 30(2):328–341, 2008. 2

[7] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 5

[8] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry. End-to-end learning of geometry and context for deep stereo regression. 2017. 1, 2, 5, 7, 8

[9] S. Khamis, S. Fanello, C. Rhemann, A. Kowdle, J. Valentin, and S. Izadi. Stereonet: Guided hierarchical refinement for real-time edge-aware depth prediction. *arXiv preprint arXiv:1807.08865*, 2018. 2, 7, 8

[10] Z. Liang, Y. Feng, Y. G. H. L. W. Chen, and L. Q. L. Z. J. Zhang. Learning for disparity estimation through feature constancy. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2811–2820, 2018. 1, 2

[11] Z. Liang, Y. Feng, Y. Guo, H. Liu, L. Qiao, W. Chen, L. Zhou, and J. Zhang. Learning deep correspondence through prior and posterior feature constancy. *arXiv preprint arXiv:1712.01039*, 2017. 8

[12] S. Liu, S. De Mello, J. Gu, G. Zhong, M.-H. Yang, and J. Kautz. Learning affinity via spatial propagation networks. In *Advances in Neural Information Processing Systems*, pages 1520–1530, 2017. 1, 3

[13] W. Luo, A. G. Schwing, and R. Urtasun. Efficient deep learning for stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5695–5703, 2016. 1, 2

[14] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3, 2013. 5

[15] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4040–4048, 2016. 6

[16] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3061–3070, 2015. 6

[17] D. Novotny, S. Albanie, D. Larlus, A. Vedaldi, A. Nagrani, S. Albanie, A. Zisserman, T.-H. Oh, R. Jaroensri, C. Kim, et al. Semi-convolutional operators for instance segmentation. *arXiv preprint arXiv:1807.10712*, 2018. 4

[18] J. Pang, W. Sun, J. Ren, C. Yang, and Q. Yan. Cascade residual learning: A two-stage convolutional neural network for stereo matching. In *International Conf. on Computer Vision-Workshop on Geometry Meets Deep Learning (ICCVW 2017)*, volume 3, 2017. 1, 8

[19] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1-3):7–42, 2002. 1

[20] X. Song, X. Zhao, H. Hu, and L. Fang. Edgestereo: A context integrated residual pyramid network for stereo matching. *arXiv preprint arXiv:1803.05196*, 2018. 2, 8

[21] Y. Wu and K. He. Group normalization. *arXiv preprint arXiv:1803.08494*, 2018. 5, 6

[22] K. Yamaguchi, D. McAllester, and R. Urtasun. Efficient joint segmentation, occlusion labeling, stereo and flow estimation. In *European Conference on Computer Vision*, pages 756–771. Springer, 2014. 1

[23] G. Yang, H. Zhao, J. Shi, Z. Deng, and J. Jia. Segstereo: Exploiting semantic information for disparity estimation. *arXiv preprint arXiv:1807.11699*, 2018. 2

[24] Q. Yang. A non-local cost aggregation method for stereo matching. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1402–1409. IEEE, 2012. 2

[25] Y. Yang, A. Yuille, and J. Lu. Local, global, and multilevel stereo matching. In *Computer Vision and Pattern Recognition, 1993. Proceedings CVPR'93., 1993 IEEE Computer Society Conference on*, pages 274–279. IEEE, 1993. 1

[26] L. Yu, Y. Wang, Y. Wu, and Y. Jia. Deep stereo matching with explicit cost aggregation sub-architecture. *arXiv preprint arXiv:1801.04065*, 2018. 2, 5

[27] J. Zbontar and Y. LeCun. Computing the stereo matching cost with a convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1592–1599, 2015. 1, 2, 8

[28] J. Zbontar and Y. LeCun. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*, 17(1-32):2, 2016. 1

[29] K. Zhang, J. Lu, and G. Lafruit. Cross-based local stereo matching using orthogonal integral images. *IEEE transactions on circuits and systems for video technology*, 19(7):1073–1079, 2009. 1, 2