Explicit Context Mapping for Stereo Matching

Anonymous ICCV submission

Paper ID 616

Abstract

Deep stereo matching usually transforms the computation into the low-resolution space to reduce the computational burden. However, mapping the disparity map from the low-resolution (LR) space to the high-resolution (HR) space causes loss of details. Many efforts have been paid on developing additional refinements to recover details. In this paper, we propose an explicit context mapping method that directly preserves details during the mapping process. We formulate the mapping as a Bayesian inference, where we deem the disparity map in the LR space as the prior and that in the HR space as the posterior. We present the explicit context which is defined as the similarity between the HR and LR disparity maps to realize the inference. Specifically, we build an explicit context mapping module which seamlessly integrates with stereo matching networks without modifying original network architectures. Experiments on the Scene Flow and KITTI 2012 datasets show that our method outperforms the state-of-the-art methods.

1. Introduction

The computer vision community has paid many efforts to increase the accuracy of stereo matching [24, 34, 31, 39]. Currently, the powerful deep neural networks bring the performance to a new stage [38, 4, 23, 12]. Transforming the computation into the low-resolution space [12, 4, 23, 17] can reduce the high computational burden of deep neural networks and provide a more reliable disparity map in ill-posed areas such as the repeated pattern, occlusion, poor-texture, and reflective surfaces. However, mapping the disparity map from the low-resolution (LR) space to the high-resolution (HR) space is prone to cause loss of details and decrease the accuracy. Many methods recover the details using additional cost aggregation and/or refinement modules [28, 32, 5, 9, 4, 16, 1, 36]. But the additional modules are not flexible to integrate with different networks. In this paper, we propose an explicit context mapping method that preserves details directly during the mapping. We build the explicit context mapping module to seamlessly integrate with stereo matching networks without modifying original network architectures.

We formulate the mapping as a Bayesian inference problem, where we deem the disparity map in the LR space as the prior and that in the HR space as the posterior. We propose the explicit context which is defined as the similarity between the HR and LR disparity maps to realize the inference. As shown in Figure 1(a), when the scale is 2, the value at the position A of the LR disparity map D_L corresponds to four values at the positions a, b, c and d of the HR disparity map D_H, so the explicit context is the similarity between D_L and D_H. We infer the posterior from the prior by mapping LR disparity values to their corresponding HR disparity values according to the similarity. We design a learning-based similarity measure to compute the explicit context by both the deep representations and relative positions of corresponding values.

For the details existing in the HR disparity map but disappearing in the LR disparity map, the explicit context cannot reliabely infer the HR disparity value from the corresponding LR value because the similarity is too small. To deal with this problem, we introduce the extended explicit context which provides more corresponding LR values for the inference. As shown in Figure 1(b), we extend the corresponding LR disparity values of the $D_H(a)$ to eight-connected values of $D_L(A)$. Then the extended explicit context is computed by the learning-based similarity measure using the deep representations and relative positions of $D_H(a)$ and its corresponding LR disparity values. The posterior can be inferred from the most reliable prior by selecting the most similar LR disparity value to compute the HR disparity.

The explicit context mapping module is able to seamlessly integrate with stereo matching networks. For the learning-based similarity, we form an adaptive location map to represent the relative positions and use a very light unit with only three 1×1 convolutional layers to compute the similarity. For the extended explicit context, the selection operation may break the end-to-end training, so we present a fully differential aggregation function to approximate the selection operation. Besides, the explicit context mapping

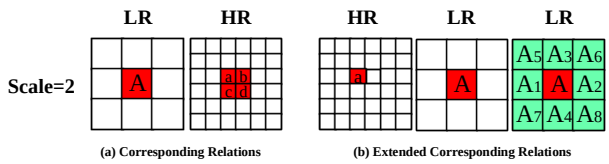

Figure 1. The corresponding relations when scale is 2. (a) The LR disparity value $D_L(A)$ at position A corresponds to four HR disparity values at positions a, b, c and d (red color). (b) We take D_H at position a as an example. The HR disparity value $D_H(a)$ at position a corresponds to the LR value $D_L(A)$ at position A (red color). The eight-connected values of $D_L(A)$ are at A_1....A_8 (green color). D_H at positions b, c, d have the same corresponding relations at position a.

module can be used for the mapping on the cost volume where the explicit context is computed by the similarity between corresponding costs. To reasonably validate our method, we test our explicit context mapping module on the popular pyramid stereo matching network (PSMNet) [4]. Experiments on the Scene Flow and KITTI 2012 datasets show that our method outperforms the PSMNet.

We summarize our contribution in two folds.

- We formulate mapping the disparity map from the LR space to the HR space as a Bayesian inference and exploit the explicit context to realize the inference.
- We design an explicit context mapping module to seamlessly integrate with deep stereo matching networks with no need for the modification of original network architectures.

2. Related Work

The early works using deep neural networks for stereo matching mainly contributed to designing better feature extractors and similarity measures for the cost computation [37, 19, 38]. For these methods, the hand-crafted post-processings like cross-based aggregation, conditional random field (CRF), semi-global optimization (SGM), filtering-based refinement were indispensable to get a sufficient accuracy [33, 39, 10, 20, 24, 27]. Kendall et al. [12] proposed an end-to-end trainable deep stereo matching network, GC-Net, which uses a $3D$ convolutional auto-encoder to model the context and geometry simultaneously. PSMNet [4] offers a more effective strategy to exploit the context by the pyramid pooling module. Currently, Liu et al. [18] and Cheng et al. [5] proposed to model the local context as the affinity matrix for stereo matching . These methods show that exploiting the context is the key to improve the accuracy. In this paper, we explicitly model the context as the similarity between the LR and HR disparity map and use it to realize the mapping process.

The performance can also be improved using additional cost aggregation and refinement modules [26, 9, 23, 36, 4]. Particularly, the EdgeStereo [28] and SegStereo [32] built the refinement modules with additional guidance like the edge and semantic segmentation, where the additional information is effective to recover the details by preserving the HR image structure. However, integrating additional modules with networks is not easy, which requires to elaborately modify original network architectures. In this paper, we design the explicit context mapping module which can seamlessly integrate with networks as the up-sampling unit for the cost volume and disparity map. Our explicit context mapping module requires barely additional computational resource or modification on the original network.

3. Explicit Context Mapping

3.1. Problem Formulation

Most existing stereo matching networks transform the computation into the low-resolution (LR) space to efficiently produce a reliable LR disparity map D_L. To make the result has the same resolution to the input images, we need to compute the high-resolution (HR) disparity map D_H from the LR disparity map D_L. This process can be realized by mapping the LR disparity value $D_L(x, y)$ to the HR disparity value $D_H(x^{'}, y^{'})$. The corresponding positions is determined by $x = \left\lfloor x^{'}/scale \right\rfloor$ and $y = \left\lfloor y^{'}/scale \right\rfloor$, where $scale$ is the scale size between HR and LR disparity maps and $\lfloor\ \rfloor$ is the floor operation. In this paper, we formulate this process as a Bayesian inference where we deem the LR disparity map as the prior and HR disparity map as the posterior. The Bayesian inference is defined as

$$P(D_H/D_L) = P(D_H, D_L)/P(D_L). \tag{1}$$

The $P(D_L)$ indicates the corresponding relations that whether LR disparity value $D_L(x, y)$ is used to infer the HR disparity $D_H(x^{'}, y^{'})$. The $P(D_H, D_L)$ is the similarity between the LR disparity value $D_L(x, y)$ and the HR disparity value $D_H(x^{'}, y^{'})$. The $P(D_H/D_L)$ donates the probability that the HR disparity value $D_H(x^{'}, y^{'})$ can be computed by the LR disparity value $D_L(x, y)$. When $D_L(x, y)$ is the corresponding LR disparity of $D_H(x^{'}, y^{'})$, the $P(D_L)$ is 1, otherwise the $P(D_L)$ is 0. When the mapping is an identity mapping, the $P(D_H, D_L)$ is 1. With the Bayesian formulation, the mapping can be given by

$$\begin{aligned} D_H(x^{'}, y^{'}) &= P(D_H/D_L) \times D_L(x, y) \\ &= P(D_H, D_L)/P(D_L) \times D_L(x, y). \end{aligned} \tag{2}$$

According to the Bayesian formulation, to preserve the details during the mapping, we need to satisfy two conditions. (1) $P(D_H, D_L)$ should reliably reveal the similarity between $D_H(x^{'}, y^{'})$ and $D_L(x, y)$. (2) We can find

a reliable corresponding value from $P(D_L)$ to infer the $D_H(x^{'}, y^{'})$.

For the first condition, we propose the explicit context to represent the $P(D_H, D_L)$. Since the critical component is to preserve relevant details between different resolutions, $P(D_H, D_L)$ needs to correctly reveal the relation between disparity values at the image structure [1] of the LR disparity map D_L and those of the HR disparity map D_H. However, we cannot get the HR disparity map to determine the image structure. Many evidences demonstrate that the deep feature from networks can model the image structure at a certain scale [6, 2, 15]. Motivated by this observation, we propose the explicit context which leverages the similarity between the HR feature F_H and the LR feature F_L, to represent the similarity between HR and LR disparity values and design a learning-based similarity measure to compute the similarity. As we can directly get the features F_H and F_L from networks, the Equation (2) is rewritten as

$$D_H(x^{'}, y^{'}) = s(F_H(x^{'}, y^{'}), F_L(x, y))/P(D_L) \times D_L(x, y), \tag{3}$$

where $s(\cdot, \cdot)$ is the learning-based similarity measure for the HR and LR features.

We introduce the extended explicit context by expanding the corresponding relations of $P(D_L)$ to make sure every HR disparity can find a reliable correspondense in the LR disparity map. As shown in Figure 1(a), HR disparity value $D_H(x^{'}, y^{'})$ at position a has only one corresponding LR disparity value $D_L(x, y)$ at position A. When the $P(F_H, F_L)$ is low which means the $D_L(x, y)$ is not a reliable correspondence for $D_H(x^{'}, y^{'})$, the $P(D_L)$ is not supportive to determine the exact value of $D_H(x^{'}, y^{'})$. The extended explicit context is modeled by extending the corresponding values of HR disparity value $D_H(x^{'}, y^{'})$ to other LR disparity values connected to the $D_L(x, y)$. Assuming we get n corresponding LR disparity values to the $D_L(x, y)$, we can infer $n + 1$ HR disparity values from the corresponding LR disparity values. The corresponding LR disparity determined by $P(D_L)$ is at the position set T, where $(x, y) \in T$ if $P(D_L) = 1$. The final HR disparity value $D_H(x^{'}, y^{'})$ is computed by selecting the most similar $D_L(x, y)$ as prior, which is expressed as

$$\begin{aligned} D_H(x^{'}, y^{'}) &= s(F_H(x^{'}, y^{'}), F_L(x, y)) \times D_L(x, y) \\ where\ (x, y) &= \underset{(x,y)\in T}{\operatorname{argmax}}\, s(F_L(x, y), F_H(x^{'}, y^{'})). \end{aligned} \tag{4}$$

3.2. Explicit Context Mapping for Disparity Map

To realize the explicit context mapping for the disparity map in the Equation (4), we design a relative location map for the similarity measure which describes the local geometrical information between the corresponding values. The relative location maps concatenate with the feature maps as two additional channels. We adopt convolutional layers with 1×1 kernel size to compute the similarity, which ensures that the similarity between $D_L(x, y)$ and $D_H(x^{'}, y^{'})$ is only calculated by the $F_L(x, y)$, $F_H(x^{'}, y^{'})$ and their relative positions.

The extended explicit context for disparity map is modeled as the Figure 1(b), where we expand the corresponding values of HR disparity value $D_H(x^{'}, y^{'})$ at position a to the eight-connected values of the LR disparity value $D_L(x, y)$ at position A. Donating the position a by $(x^{'}, y^{'})$, the corresponding position set T for $D_H(x^{'}, y^{'})$ is $\{(x+i, y+j)\}$ where $i, j \in \{-1, 0, 1\}$. With the extended explicit context, we can infer nine HR disparity values from the corresponding LR disparity values $D_L(T)$. To ensure the end-to-end training with extended explicit context, we design a fully differential aggregation function to approximate the Winner-Take-All (WTA) selection of the most similar correspondence. It fuses the extended context by the probability from the softmax function. Then the Equation (4) is rewritten as

$$\begin{aligned} D_H(x^{'}, y^{'}) = &\sum_{(x,y)\in T} \sigma(s(F_H(x^{'}, y^{'}), F_L(x, y))) \\ &\times P(D_L) \times D_L(x^{'}, y^{'}), \end{aligned} \tag{5}$$

where $\sigma(\cdot)$ is the softmax function which computes the probability by the similarity $\sigma(s(F_L(x, y), F_H(x^{'}, y^{'})))$. As the LR disparity D_L is assumed to be reliable during the Bayesian inference, mapping with the softmax value can approximate the WTA selection. If the corresponding value exists in the extended corresponding relations, the LR disparity value directly represents the HR disparity value by the identity mapping. Otherwise, the value is approximated by fusing the corresponding values to ensure the smoothness.

3.3. Explicit Context Mapping for Cost Volume

For the cost volume, the LR matching cost $C_L(d, x, y)$ is computed by

$$C_L(d, x, y) = cnn(F_L(x, y), \widetilde{F}_L(x - d, y)). \tag{6}$$

where d is the depth level, F_L is the LR feature map from the left image and $\widetilde{F}_L$ is the LR feature map from the right image, and $cnn(\cdot, \cdot)$ is the cost computation function realized by convolutional networks. The similarity measure for the HR and LR costs leverages both the feature map from the left and right image, which is given by

$$\begin{aligned} P(C_L, C_H) = &s(F_H(x^{'}, y^{'}), F_L(x, y)) \\ &\times s(\widetilde{F}_H(x^{'} - d^{'}, y^{'}), \widetilde{F}_L(x - d, y)). \end{aligned} \tag{7}$$

[1] In this paper, the image strcuture refers to edges, corners and junctions [14, 29, 25]

The extended corresponding relations on the cost volume are determined by the eight-connected relations from both the left and right images. Limited by the structure of the cost volume, which is computed by shifting the image along X dimension, there are 27 corresponding costs on the LR cost volume C_L. For the HR cost at $(d^{'}, x^{'}, y^{'})$, the corresponding position set of T is computed by

$$T \in \begin{Bmatrix} (d, x+i, y+j) & i,j \in \{-1,0,1\}; \\ (d+1, x+i, y+j) & i \in \{0,1\}, j \in \{-1,0,1\}; \\ (d-1, x+i, y+j) & i \in \{-1,0\}, j \in \{-1,0,1\}; \\ (d+2, x+i, y+j) & i \in \{1|, j \in \{-1,0,1\}; \\ (d-2, x+i, y+j) & i \in \{-1\}, j \in \{-1,0,1\}; \end{Bmatrix}.$$

The Equation (4) is rewritten as

$$C_H(d^{'}, x^{'}, y^{'}) = \sum_{(o,m,n)\in T} C_L(o,m,n) \times \sigma(s(F_H(x^{'}, y^{'}), F_L(m,n)) \times s(\widetilde{F_H}(x^{'} - d^{'}, y^{'}), \widetilde{F_L}(m-o, n))). \tag{8}$$

This operation approximates the real matching cost result computed from the HR feature, but it has large computational complexity. Besides, the $soft\text{-}argmin$ function selects the lowest matching cost along the D dimension to get the final HR disparity map. So the mapping that won't influence the selection is redundant. For example, assuming the $soft\text{-}argmin$ result for $C_L(:,x,y)$ is d, then the $soft\text{-}argmin$ result $d^{'}$ for $C_H(:, x^{'}, y^{'})$ should be $d^{'} \in [d * scale, (d+1) * scale)$. Therefore, the mapping on other depth level is not helpful to compute the final disparity.

To efficiently realize the mapping on the cost volume, we separate the mapping into two steps, first mapping along the D dimension and then mapping along X, Y dimensions. The extended explicit context is given by nine similarity matrices from the left image and three similarity matrices from the right image. S indicates the similarity matrices computed by $F(x^{'}, y^{'})$ and $F(T)$ from the left image, $T \in \{(x+i, y+j)\}$ where $i,j \in \{-1,0,1\}$. $\widetilde{S}$ is the similarity matrices computed from the right image between $\widetilde{F}_L(x^{'}, y^{'})$ and $\{\widetilde{F}_L(x-d, y),\ \widetilde{F}_L(x-d-1, y),\ \widetilde{F}_L(x-d+1, y)\}$. The realization of the first step is shown in Algorithm 1. The C_L is first expanded through the identity mapping to form $C_L^{'}$ which has same size of C_H. Then we use the $\widetilde{S}$ to aggregate the $C_L^{'}$ along D dimension by the three connected corresponding relations. After that, the $C_L^{'}$ is squeezed along the X, Y dimenions to finish the mapping along D dimension. We select the most representative cost $C_L^{'}(:, x^{'}, y^{'})$ to represent the $C_L(x,y)$ according to the similarity between $F_H(T)$ and $F_L(x,y)$. The position set T is $\{(x * scale + i, y * scale + j)\}$, where

Algorithm 1: Mapping of Cost Volume along D Dimension

Input: similarity matrices of left image S, similarity matrices of right image $\widetilde{S}$, LR cost volume C_L
Output: Expanded LR cost volume $C_L^{'}$

```
S.shape: (9, H, W), S̃.shape: (3, H, W)
C_L.shape: (D/4, H/4, W/4)
max_d=192
scale=4
S̃ =softmax(S̃,dim=0)
C_L' = C_L.expand_as(D, H, W)
for (i = 0; i <= 192; i++) do
    C_L'[i, :, i:] += C_L'[i, :, i:] × S̃(0)[:, :-i]
                    + C_L'[i+scale, :, i:] × S̃(1)[:, :-i]
                    + C_L'[i-scale, :, i:] × S̃(2)[:, :-i]
end
S'(0) = S(0).reshape_as(H/4, W/4, 16)
S'(0) =softmax(S'(0), dim=-1)
C_L' = C_L'.reshape_as(D, H/4, W/4, 16)
C_L' =sum(C_L' × S'(0), dim=-1)
C_L'. shape:(D, H/4, W/4)
return C_L'
```

$i,j \in \{0,1,...,scale\}$. The second step is similar to the mapping on the disparity map as Equation (5), where the difference is the mapping value $C_H^{'}$ is with the shape of $(D,\ H/4,\ W/4)$. The corresponding position set T for the $C_H(:, x^{'}, y^{'})$ is $\{(x+i, y+j)\}$ where $i,j \in \{-1,0,1\}$. The mapping along the X, Y dimensions is computed by

$$C_H(:, x^{'}, y^{'}) = \sum_{(x,y)\in T} \sigma(s(F_L(x,y), F_H(x^{'}, y^{'}))) \times C_L^{'}(:, x^{'}, y^{'}). \tag{9}$$

4. Implementation

4.1. Learning-based Similarity Measure

We design a learning-based similarity measure using both the deep representation and relative positions. The HR feature is computed before the first sub-sampling unit, and the LR feature is computed from the last $2D$ convolutional layer. We use identity mapping to expand the LR feature F_L as the same size of HR feature F_H to get the expanded feature F_e. We concatenate F_e and F_H along the channel dimension to form the fused feature F_f. We design an adaptive location map G according to relative positions of corresponding values. The location map denotes how far the pixels are to their mapping center. With the $scale$ size 4, the location map G of $[x*4:(x+1)*4, y*4:(y+1)*4]$ is calculated by

$$x^{'} = \begin{pmatrix} -2 & -1 & 1 & 2 \\ -2 & -1 & 1 & 2 \\ -2 & -1 & 1 & 2 \\ -2 & -1 & 1 & 2 \end{pmatrix}, y^{'} = \begin{pmatrix} 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 \\ -2 & -2 & -2 & -2 \end{pmatrix}.$$

The location map concatenates with the fused feature F_f as two additional channels to form the similarity feature F_s.

We adopt three convolutional layers with 1×1 kernel to compute the similarity from the similarity feature F_s. Each layer is followed by LeakeyRelu with 0.1 threshold except for the last layer with output channel 1.

The extended explicit context is computed by shifting the feature map and location map. We take the similarity maps between $F_H(x^{'}, y^{'})$ and $F_L(x-1, y)$ as the example. The fused feature $F_f(:, 4:)$ is obtained by concatenating $F_H(:, 4:)$ with the expanded feature $F_e(:, :-4)$. Four zero-paddings are appended on the left side to get the fused feature F_f. The location map G is calculated by

$$x^{'} = \begin{pmatrix} -4 & -3 & -2 & -1 \\ -4 & -3 & -2 & -1 \\ -4 & -3 & -2 & -1 \\ -4 & -3 & -2 & -1 \end{pmatrix}, y^{'} = \begin{pmatrix} 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 \\ -2 & -2 & -2 & -2 \end{pmatrix}.$$

4.2. Network Architecture

We use the pyramid stereo matching network (PSMNet) as our baseline[4]. We add some little modifications to the network but do not change the main structure. As we need the HR feature from the network, we add four $2D$ convolutional layers at the start of the network. Besides, limited by the computational resource, we use the GroupNorm [30] to replace the BatchNorm [11] with a group division of 32. Using the GroupNorm ensures fairness when we test the effectiveness of our method in different scales. The batch size for PSMNet is hard to set more than 2 on a single GPU when the $scale$ is 4. For the computational efficiency, we set the batch size to 8 when the $scale$ is 8. For the training loss, we use the SmoothL_1 loss as PSMNet [4, 8]. The details of the architecture settings are shown in Table 1 in the **Supplement Materials**.

4.3. Complexity Analysis

The complexity of using CUDA with GPU for convolutional operation is $O(log_2(k^2))$, where k is the kernel size. As the similarity measure is realized by three 1×1 convolutional layers, the computational complexity is the constant time $O(1)$. The aggregation function is built based on softmax, which has a complexity of $O(log_2(n)$ on the CUDA, where n is the number of channels of the softmax dimension. The aggregation is with constant n where for the cost volume $n = 3, 9$, and for the disparity map $n = 9$. So the computational complexity of aggregation is also constant time $O(1)$. In theory, matrix product, matrix addition, convolution and softmax operation can be performed parallel for all pixels and channels, so both the complexity of similarity measure and aggregation are irrelevant to the input size. For the mapping on the disparity map, the overall complexity is $O(1)$ as there is no iterative operation. For the mapping on the cost volume, the overall complexity is $O(n)$ as we shift the $\widetilde{S}$ for n times, where n is the maximum disparity. The actual storage cost is discussed in detail with the experiments in Section 5.2.

5. Experiements

We evaluate our method on the Scene Flow [21], KITTI 2012 [7] and KITTI 2015 [22]. All networks are implemented with PyTorch and optimized by a standard Adam optimizer with $\beta_1 = 0.9, \beta_2 = 0.999$. The input images 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 a size of (256×512) and set the maximum disparity value as 192. All of the training parameters are the same as the open source code of PSMNet [4]. With the modification of the normalization unit and four additional 2D convolutional layers, we reproduce the PSMNet as the baseline to test the effectiveness of our method.

5.1. Benchmark Results

Comparisons on the Scene Flow dataset: Scene Flow [21] is a large synthetic dataset containing 34896 training images and 4248 testing images with size of 540×960. This dataset has three rendered sub-datasets: FlyingThings3D, Monkaa, and Driving. FlyingThings3D is rendered from the ShapeNet [3] 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. We use all of the three sub-datasets for the training and use the standard test dataset for the evaluation.

We train our network for 8 epochs on the whole training dataset with learning rate 0.001. The results are shown in Table 1. Compared with other state-of-the-art methods like CRL[23], DispNetC [9], GC-Net [12], StereoNet [13], Edge Stereo [28], CSPN [5], we get the best performance with the lowest EPE error. For the mapping on the disparity map, we get 0.7012 of the EPE (end-point error) compared with the baseline PSMNet (disparity) 0.92, which is the promotion of 0.39 and 35.7% to the PSMNet (disparity). For the mapping on the cost volume, we get 0.5917 EPE compared with PSMNet 1.09 EPE, which has a promotion 0.50 and 45% to the PSMNet. Compared with EdgeStereo method [28] with EPE of 0.751, our method on the cost volume achieves 0.1593 gain on EPE, and the mapping on the disparity map obtain 0.045 gain on EPE.

Comparisons on the KITTI 2012 dataset: KITTI 2012 is a real-world dataset with street views from a driving car. It contains 194 training stereo image pairs with sparse ground-truth disparities obtained using LiDAR and another 200 testing image pairs without ground-truth disparities. We take 160 images for training and left 34 images for eval-

Table 1. Comparison on SceneFlow dataset

Method	CRL[23]	DispNetC [9]	GC-Net [12]	StereoNet [13]	Edge Stereo[28]	CSPN[5]	**PSMNet**[4]	**PSMNet (disparity)**[4]	**Ours on disparity**	**Ours on cost volume**
EPE	1.32	1.68	2.51	1.101	0.751	0.78	1.09	0.92	0.7012	0.5917

Table 2. Comparisons on KITTI 2012

Method	>2 px		>3 px		>5 px		Mean Error(px)		Time(s)
	Noc	All	Noc	All	Noc	All	Noc	All	
M2S_CSPN [5]	1.79	2.27	1.19	1.53	0.77	0.98	0.5	0.5	0.5
HD3-Stereo [35]	2.00	2.56	1.40	1.80	0.94	1.19	0.5	0.5	0.09
EdgeStereo [28]	2.32	2.88	1.46	1.83	0.83	1.04	0.4	0.5	0.32
SegStereo [32]	2.66	3.19	1.68	2.03	1.00	1.21	0.50	0.6	0.6
GC-Net [12]	2.71	3.46	1.77	2.30	1.12	1.46	0.6	0.7	0.9
PSMNet (disparity) [4]	2.76	3.84	1.68	2.59	0.91	1.45	0.6	0.6	0.37
PSMNet [4]	2.44	3.01	1.49	1.89	0.90	1.15	0.5	0.6	0.41
Ours on disparity map	2.61	3.72	1.58	2.43	0.90	1.43	0.5	0.6	0.37
Ours on cost volume	**2.44**	3.82	**1.48**	2.63	**0.85**	1.61	0.5	0.6	0.45

uation. We use the pre-trained model from SceneFlow and fine-tune the model on KITTI 2012 for 1000 epochs, where we use learning rate 0.001 the first 700 epochs and modify it to 0.0001 for the left 300 epochs.

The comparisons with other state-of-the-art methods like GC-Net [12], StereoNet [13], Edge Stereo [28], CSPN [5], HD3-stereo [35], PSMNet [4] are shown in Table 2. We also report the PSMNet (disparity) which gets the HR disparity by the bilinear of disparity map. We can see our method for the disparity map can outperform the PSMNet (disparity) by $0.15, 0.22$ on the 2-px error, $0.1, 0.16$ on the 3-px error and $0.01, 0.02$ on the 5-px error. The explicit context mapping on the cost volume outperforms the PSMNet in the non-occlusion (Noc) areas which is 0.01 on the 3-px error and 0.05 on the 5-px error.

Comparisons on the KITTI 2015 dataset: 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 take 160 images for training and left 40 images for evaluation. The training process for KITTI 2015 is the same as KITTI 2012. The results compared with state-of-the-art methods are shown in Table 3. We also report the PSMNet (disparity) to compare with our method on the disparity map. Our method for the disparity map outperforms the PSMNet (disparity) by $0.3, 0.58, 0.43$ on the all pixel estimation and $0.17, 0.32, 0.2$ on the non-occluded areas. Our method for the cost volume reaches comparable performance with the PSMNet and outperforms PSMNet in the non-occluded foreground areas by 7%.

On KITTI 2012 and KITTI2015 dataset, the promotion of the PSMNet is much lower than it on the Scene Flow. The reason comes from two sides. The first is that the sparse ground truth suffers from the loss of details itself, so training with the sparse supervision draws back the effectiveness of our method. The ground truth without details makes it ill-posed for our method to preserve the details. Our method is built based on the assumption that the LR results are reliable, so we can see our method can promote the PSMNet on the non-occluded areas. But it draws back the performance on the occluded areas, the reason is obvious that the mismatching results on the occluded areas will also be propagated through the mapping process. The second is the overfitting problem. We get a much low evaluation error on the KITTI 2015 which is 1.63% of the 3-px error compared with the 1.83% of PSMNet. We observe that the evaluation error is much lower which has the promotion of 18% to the baseline but the test error is worse than the baseline by 6%. The explicit context mapping method is not designed to handle the ill-posed areas, where we focus on keeping the details during the mapping which can further improve the performance if we get a reliable matching result. The experiments on the benchmarks show that our method can promote the the PSMNet on the non-occluded areas.

5.2. Ablation

The ablation analysis is conducted on the Scene Flow dataset. We train all of the comparsion models for 8 epochs without using any pre-trained model. The results are shown in Table 4.

5.2.1 Mapping on the Cost Volume and Disparity Map

We test the effectiveness of the explicit context module for both the cost volume and disparity map. Compared with all of the Baselines with Ours, both the mapping on the cost volume and disparity map can obviously improve the accu-

Table 3. Comparisons on KITTI 2015

Model	All pixels			Non-Occluded Pixels			Time(s)
	D1-bg	D1-fg	D1-all	D1-bg	D1-fg	D1-all	
GC-Net[12]	2.21	6.16	2.87	2.02	5.58	2.61	0.9
HD3-stereo[9]	1.70	3.63	2.02	1.56	3.43	1.87	0.09
EdgeStereo[28]	2.27	4.18	2.59	2.12	3.85	2.40	0.27
StereoNet[13]	4.30	7.45	4.83	4.05	6.44	4.44	0.02
CSPN[5]	1.51	2.88	1.74	1.40	2.67	1.61	0.5
PSMNet (disparity)[4]	2.59	5.85	3.13	2.10	4.99	2.58	0.37
PSMNet[4]	1.86	4.62	2.32	1.71	4.31	2.14	0.41
Ours on the disparity map	**2.29**	**5.27**	**2.79**	**1.93**	**4.67**	**2.38**	0.37
Ours on tthe cost volume	2.29	4.93	2.73	1.90	**4.06**	2.26	0.45

Table 4. The Ablation on Scene Flow. EPE: End-point-error

ID	Scale		Mapping on Disparity map	Mapping on Cost Volume	EPE	Non-Occluded	Memory(M)	Time(s)
	4	8						
Baseline1	✓		✓		0.9251	0.8288	2931	0.37
Ours1	✓		✓		**0.7012**	**0.6056**	3107	0.37
Baseline2	✓			✓	0.8202	0.6970	3303	0.46
Ours2	✓			✓	**0.5978**	**0.5068**	3870	0.51
Baseline3		✓	✓		1.0306	0.9245	2229	**0.19**
Ours3		✓	✓		**0.9046**	**0.8086**	2403	**0.19**
Baseline4		✓		✓	0.9594	0.8562	2609	0.21
Ours4		✓		✓	**0.8449**	**0.7421**	3106	0.25

racy.

The mapping on the disparity map gets the EPE of 0.7012 compared with Baseline1 0.9251, which is 24% gaingain, and increases fewcomputational resource with $170M$ memory. By individually removing the additional layers, we find that 40% of the additional memory comes from the additional four 2D convolutional layers at the start of the network which takes $68M$, and the others come from the computation of similarity. The mapping process won't increase the time cost, as the matrix production operation can be realized by $O(1)$ on GPU. From comparisons between Ours1 and Baseline1, we can see the promotion of the mapping on the disparity map is obvious.

The promotion of the EPE estimated from all pixels is 0.225 and that on the non-occluded areas is 0.223 compared to Baseline1 0.9251 and 0.8288. The EPE improves 23% estimated from all pixel which is lower than 26% estimated from the non-occluded areas. The reason is that the context mapping relies on the correct low-resolution values. But for the occluded pixels, the LR matching results are not reliable at all, so the mapped values will get wrong values, which decreases the EPE. While for the reliable non-occluded pixels, the mapping can enjoy the reliable mapping.

Our method on the cost volume gives the best results of EPE 0.5068, which is 28% gain compared with Baseline2 0.6970. The increased accuracy not only comes from the mapping process but also enjoys the cost aggregation performance. In terms of resource cost, however, our method slows down the Baseline2 by $0.05s$ and takes more computational resource around $570M$ memory on GPU. The slower speed and higher memory come from the shift operation of the mapping along the D dimension. Although our method increases the computational resource, it is still more efficient than the other cost aggregation method using deep neural networks whose cost aggregation module takes more than $2400M$ memory on GPU [36]. From comparisons between Ours2 and Baseline2, the promotion of the EPE estimated from all pixels are larger than that on the non-occluded areas, which is 0.223 and 0.191. This is because the mapping on the cost volume can also realize the cost aggregation which can rectify some of mismatching results. The different effectiveness between the occluded and non-occluded areas is consistent on the KITTI benchmark.

5.2.2 Multi-scale Performance

We also test our method on the larger $scale$ 8. Compared with Ours1, Ours3 reduces about 24% memory and speeds up about $0.2s$ with the drop of EPE by 0.2. This is because the LR disparity map is not reliable when the scale becomes

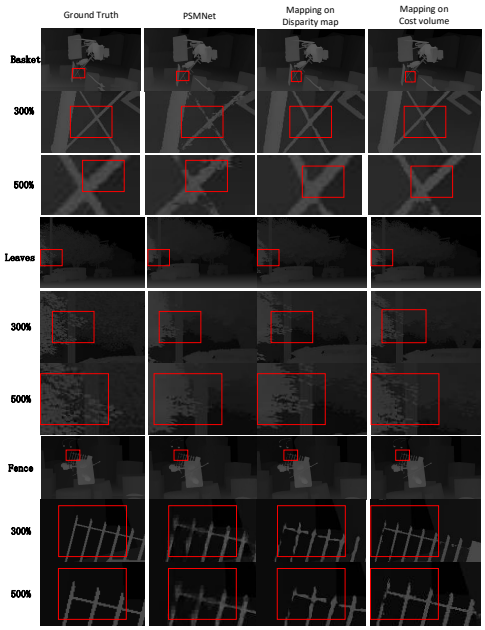

Figure 2. Qualitative analysis on details of our method for disparity map and cost volume (better with a zoom-in view). We visualize the three classes with rich details, Basket, Leaves, Fence. The image is scaled to 300% and 500% to get a clear view of the loss of details.

larger, so the mapping will suffer from the wrong LR disparity values. Compared with Ours2, Our4 get a faster speed of $0.25s$ and lower memory with $770M$ along with the drop of EPE by 0.25. For comparison between Ours3 and Our1, Our4 and Our2, we can see when the scale becomes large, our method get a drop of accuracy from the unreliable LR matching results. However, even the promotions, i.e. 10% on disparity map and 11% on the cost volume are lower than that on the *scale* 4, the explicit context mapping still gives an obvious improvement on the large scale. Besides, with the large scale, the improvement of efficiency is obvious. Some of the applications on the mobile platform can only afford the scale of 8 or even lower, so our explicit context mapping can be used for these applications to improve accuracy.

5.3. Visualization

We visualize the mapping results compared with PSMNet on the Scene Flow. From Figure 2, we can see our method effectively maintains the details like edges and corners. Besides, with the scale of 500%, we can see explicit context mapping can keep the pixel-level image structure, whose edge is as sharp as the ground truth. We can also see the explicit context mapping can prevent the vanishment of small objects by the illustration of the Leaves.

6. Discussion

Here, we discuss the potentiality of explicit context mapping for beyond applications. The mapping process is common for the pixel-wise labeling tasks like segmentation, optical flow and so on. These methods normally suffer from the high computational burden to handle the HR images, so they always transform the computation into the LR space. Therefore, the mapping back to the HR space is indispensable. In this paper, we test the explicit context for the stereo matching, where it shows great effectiveness to preserve the loss of details during the mapping. For other applications, both our Bayesian formulation of the mapping and the explicit context mapping module are appropriate to handle the mapping process. The Bayesian formulation makes it applicable for other methods, where we can simply realize the explicit context mapping by adjusting the similarity measure and corresponding relations. For example, the LR optical flow can be mapped to HR optical flow by the explicit context mapping, where the explicit context is computed by the similarity between corresponding values. The optical flow has a heavier computational burden than the stereo matching, so the computation is always conducted in a lower resolution. As we test in the multi-scale experiments, the explicit context mapping can also keep the promotion in larger scales. When the explicit context mapping provides sufficient accuracy on a large scale, it gives a very efficient performance both in memory and speed. The promotion might be limited for the semantic segmentation because details are not indispensable to the final performance for segmentation. However, the explicit context mapping is beneficial for the small objects segmentation, where the small objects like leaves can keep their shapes in wide scenes, as shown in Figure 2.

7. Conclusions

In this paper, we have presented the explicit context mapping for the stereo matching. The explicit context is effective to support the Bayesian inference to preserve details. Modeling the explicit context as the similarity between HR and LR disparity map can help to align the image structure between HR and LR space, which gives sharp edges and keeps the shape of small objects. The learning-based similarity can leverage both the relative position and deep representation to reliably reveal the relations between HR and LR disparity map. The extended explicit context makes sure the inference can be realized from at least one reliable prior. The explicit context module can be used for the mapping on the disparity map and cost volume. The explicit context mapping can be simply trained end-to-end with other networks and do not change original architectures of stereo networks. The extensive experiments showed our method can improve the accuracy on Scene Flow and KITTI datasets with less computational resources.

References

[1] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015. 1

[2] S.-H. Baek, I. Choi, and M. H. Kim. Multiview image completion with space structure propagation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 488–496, 2016. 3

[3] 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. 5

[4] 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

[5] X. Cheng, P. Wang, and R. Yang. Learning depth with convolutional spatial propagation network. *arXiv preprint arXiv:1810.02695*, 2018. 1, 2, 5, 6, 7

[6] E. L. Denton, S. Chintala, R. Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in neural information processing systems*, pages 1486–1494, 2015. 3

[7] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3354–3361. IEEE, 2012. 5

[8] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 5

[9] 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. 1, 2, 5, 6, 7

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

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

[12] 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, 6, 7

[13] 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. 5, 6, 7

[14] J. J. Koenderink. The structure of images. *Biological cybernetics*, 50(5):363–370, 1984. 3

[15] Z. Lee and T. Q. Nguyen. Multi-resolution disparity processing and fusion for large high-resolution stereo image. *IEEE Transactions on Multimedia*, 17(6):792–803, 2015. 3

[16] 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

[17] 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. 1

[18] 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. 2

[19] 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. 2

[20] Z. Ma, K. He, Y. Wei, J. Sun, and E. Wu. Constant time weighted median filtering for stereo matching and beyond. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 49–56, 2013. 2

[21] 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. 5

[22] 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. 5

[23] 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, 2, 5, 6

[24] 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, 2

[25] I. M. Scott, T. F. Cootes, and C. J. Taylor. Improving appearance model matching using local image structure. In *Biennial International Conference on Information Processing in Medical Imaging*, pages 258–269. Springer, 2003. 3

[26] A. Shaked and L. Wolf. Improved stereo matching with constant highway networks and reflective confidence learning. *arXiv preprint arXiv:1701.00165*, 2016. 2

[27] R. Slossberg, A. Wetzler, and R. Kimmel. Deep stereo matching with dense crf priors. *arXiv preprint arXiv:1612.01725*, 2016. 2

[28] 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. 1, 2, 5, 6, 7

[29] A. Torralba and A. Oliva. Depth estimation from image structure. *IEEE Transactions on pattern analysis and machine intelligence*, 24(9):1226–1238, 2002. 3

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

[31] 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

[32] G. Yang, H. Zhao, J. Shi, Z. Deng, and J. Jia. Segstereo: Exploiting semantic information for disparity estimation. *arXiv preprint arXiv:1807.11699*, 2018. 1, 2, 6

[33] 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

[34] 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

[35] Z. Yin, T. Darrell, and F. Yu. Hierarchical discrete distribution decomposition for match density estimation. *arXiv preprint arXiv:1812.06264*, 2018. 6

[36] L. Yu, Y. Wang, Y. Wu, and Y. Jia. Deep stereo matching with explicit cost aggregation sub-architecture. *arXiv preprint arXiv:1801.04065*, 2018. 1, 2, 7

[37] 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. 2

[38] 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, 2

[39] 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