Deep Stereo Matching with Explicit Cost Aggregation Sub-Architecture

Lidong Yu[1], Yucheng Wang[2], Yuwei Wu[1]*, and Yunde Jia[1]
[1]Beijing Laboratory of Intelligent Information Technology, School of Computer Science,
Beijing Institute of Technology, Beijing, 100081
[2]Kandao Australia Research Center
Suite 3.05 1 Richardson Place North Ryde NSW 2113, Australia
{yulidong, wuyuwei, jiayunde}@bit.edu.cn, wyc@kandaovr.com

Abstract

Deep neural networks have shown excellent performance for stereo matching. Many efforts focus on the feature extraction and similarity measurement of the matching cost computation step while less attention is paid on cost aggregation which is crucial for stereo matching. In this paper, we present a learning-based cost aggregation method for stereo matching by a novel sub-architecture in the end-to-end trainable pipeline. We reformulate the cost aggregation as a learning process of the generation and selection of cost aggregation proposals which indicate the possible cost aggregation results. The cost aggregation sub-architecture is realized by a two-stream network: one for the generation of cost aggregation proposals, the other for the selection of the proposals. The criterion for the selection is determined by the low-level structure information obtained from a light convolutional network. The two-stream network offers a global view guidance for the cost aggregation to rectify the mismatching value stemming from the limited view of the matching cost computation. The comprehensive experiments on challenge datasets such as KITTI and Scene Flow show that our method outperforms the state-of-the-art methods.

Introduction

Stereo matching is one of the fundamental problems in computer vision community. The goal of stereo matching is to compute a disparity map from images collected by stereo cameras. The disparity map is widely used in 3D scene reconstruction, robotics, and autonomous driving. Driven by the emergence of large-scale data sets and fast development of computation power, deep neural networks have proven effective for stereo matching. Many state-of-the-art methods raise the performance by learning robust local features or similarity measurements for cost computation (Zbontar and LeCun 2015; Luo, Schwing, and Urtasun 2016; Shaked and Wolf 2016). However, these methods still have difficulties in textureless areas and occluded regions because of the limited view field during cost computation.

To handle mismatching values of the cost computation results, which is called cost volume, the cost aggregation step is indispensable in traditional stereo matching methods. Cost aggregation is applied to the cost volume to rectify the incorrect values by aggregating the computed matching cost. It is typically performed by summing or averaging the matching cost over a support region within a constant disparity (Yang 2012; Min, Lu, and Do 2011; Tombari et al. 2008). However, the traditional cost aggregation methods are limited by the shallow, hand-designed scheme to perform the aggregation. They cannot effectively take global view guidance into account while keeping the local fitness. In this paper, we propose a learning-based cost aggregation to keep the balance between global view and local fitness using a novel two-stream neural network.

[*]corresponding author

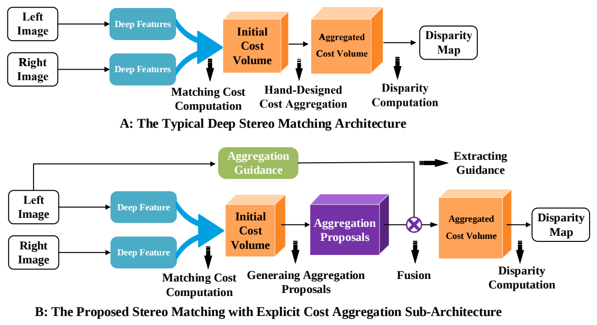

Figure 1: Comparisons between the typical deep stereo matching pipeline and pipeline with our learning-based cost aggregation. The architecture A is the typical deep stereo matching pipeline with traditional cost aggregation method. The architecture B is our learning-based cost aggregation. The details of architectures will be shown in Figure 2, where the parts are matching according to the colors.

The proposed cost aggregation can incorporate with other deep stereo matching pipeline in an end-to-end manner because it is conducted as a sub-architecture for the whole network. With the learning-based cost aggregation, the end-to-end trainable stereo matching pipeline can not only learn the feature and similarity measurementment for cost computation but also perform the cost aggregation. The comparisons of the proposed architecture with typical deep stereo pipelines are shown in Figure 1. We can see that the learning-based cost aggregation is carried out by a two-stream network in an explicit manner.

The cost aggregation process is reformulated as a learn-

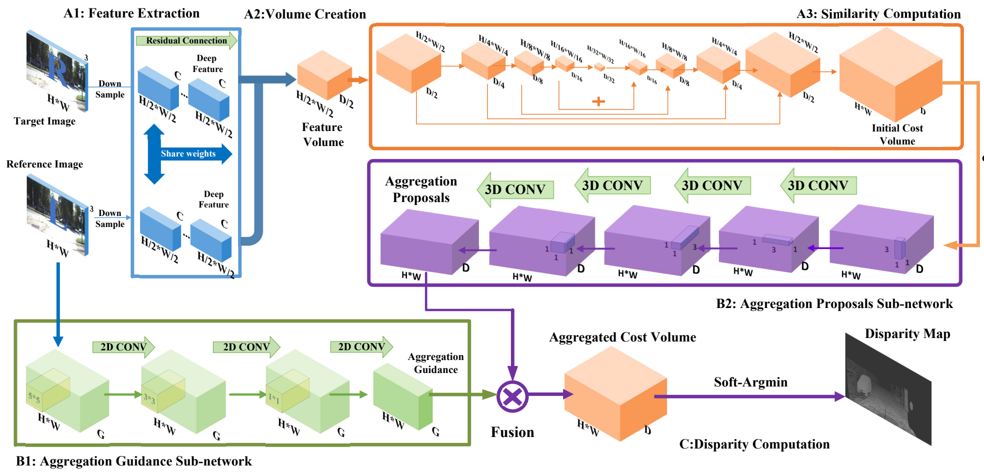

Figure 2: Our stereo matching pipeline with learning-based cost aggregation. The different colors represent the different size of data: blue for $W \times H \times C$, orange for $D \times H \times W \times C$, green for $H \times W \times G$ and purple for $D \times H \times W \times G$. The cost computation step is divided into three components. A1 is a feature extraction sub-network using the residual Siamese network. A2 is a feature volume construction part, and the detailed illustration is shown in Figure 3. A3 computes the similarity between the feature volume using a 3D convolutional network and produces the initial cost volume. The learning-based cost aggregation is carried out by a two-stream network as a sub-architecture for the whole pipeline. The guidance stream is illustrated in B1. The proposals stream is shown in B2. The cost aggregation result is obtained by a winner-take-all strategy to select the best proposal. Finally, a soft-argmin function is employed to compute the disparity map.

ing mechanism to generate potential cost aggregation results called proposals and select the best one. Accordingly, the learning-based cost aggregation is carried out by a two-stream network: one stream is used for generating the proposals and the other stream is employed for evaluating proposals. The first stream holds the local fitness by generating potential aggregation results according to the cost volume computed from the cost computation. The generation is performed by a convolutional operation along the three dimensions of the cost volume, which aggregates information both on the spatial and depth space. The second stream brings in global view guidance for the cost aggregation by evaluating each proposal. For each proposal, it is evaluated by the guidance with the same size of the image, which is considered as the global view guidance. The guidance is obtained by a light convolutional network to bring in low-level structure information which is treated as the evaluation criterion for proposals. Since the structure information only contains 2D information, which is independent in depth, the guidance is unchanged along the depth dimension. Therefore, the evaluation for each proposal shares the same guidance for different disparities. After evaluating each proposal, a winner-take-all strategy is employed to choose the best-aggregated value to form the aggregated cost volume

The proposed architecture reaches a promising accuracy on the Scene Flow (Mayer et al. 2016) and the KITTI benchmark (Menze and Geiger 2015; Geiger, Lenz, and Urtasun 2012). Our contributions are three-fold.

- This work is, to the best of our knowledge, the first to explicitly model the cost aggregation in a learning-based scheme for stereo matching. We reformulate the cost aggregation as the learning process of generation and selection of cost aggregation proposals.
- We propose a novel two-stream network to carry out the generation and selection of cost aggregation proposals. The proposed two-stream network maintains the rich semantic information while brings in low-level structure information, which demonstrates the ability to fuse the high-level feature with the low-level feature.
- The proposed learning-based cost aggregation is carried out as a sub-architecture of the deep stereo matching pipeline in an end-to-end trainable manner. It is flexible for the pipelines without cost aggregation to raise accuracy.

Related Work

Deep neural networks for Cost computation

Using deep neural networks for stereo matching was firstly introduced by Zbontar et al. (Zbontar and LeCun 2015) with a Siamese network for cost computation. Luo et al. (Luo, Schwing, and Urtasun 2016) reduced the computation time by replacing the full-connection layer with an inner product. For the stereo matching task, the Siamese network is responsible for extracting deep representations for each pixel. The original simple convolutional layers are limited to generate the rich semantic representation, so the improved highway network such as the residual network was employed to improve representations under the Siamese architecture (Shaked and Wolf 2016; Xu, Ranftl, and Koltun 2017). Then a similarity measurementment is applied to compute the matching cost between corresponding pixels. Inspired by the progress of the dense pixel-wise task such as optical flow and semantics segmentation, the 3D auto-encoder shows excellent performance by a large view field. The closely work with our method is GC-Net which is an end-to-end pipeline using a 3D auto-encoder as the similarity measurement (Kendall et al. 2017). Similarly, we utilize the residual Siamese network for feature extraction and leverage the 3D auto-encoder to compute the similarity. For deep stereo matching pipelines, the use of volume processing has been proven effective to combine the feature extraction and similarity measurement (Xu, Ranftl, and Koltun 2017). We modify the traditional concatenating construction with an additional shift operation to construct a more effective feature volume.

Despite the usage of deep neural networks for cost computation improve the stereo matching performance, it still has limitations on textureless areas, weak structure, and occluded regions. Hand-designed cost aggregation methods are normally used on the initial cost volume, whose improvement is barely adequate (Zbontar and LeCun 2015; Luo, Schwing, and Urtasun 2016). In this paper, we present a learnable cost aggregation method which can collaborate with deep cost computation methods in an end-to-end trainable manner. The two-stream network is shown effective to fuse different classes of features in video action recognition (Simonyan and Zisserman 2014). Inspired by this, we design a novel two-stream network to carry out the cost aggregation. The two-stream network is presented to maintain the rich semantics of the cost computation while bringing into low-level structure information to guide the cost aggregation. The low-level structure information can be used as the global view guidance by a light neural network architecture (Mahendran and Vedaldi 2015; Zeiler and Fergus 2014). The fusion of two-stream network is always realized by a concatenating function (Feichtenhofer, Pinz, and Zisserman 2016), in contrast, we introduce a winner-take-all strategy to fuse the two streams.

Cost Aggregation

According to the taxonomy of stereo matching (Scharstein and Szeliski 2002), a typical stereo matching pipeline can be divided into four steps: matching cost computation, cost aggregation, disparity computation, and disparity refinement. Many cost aggregation methods have been proposed to obtain high-quality disparity maps. Normally, most of them were performed locally by aggregating the matching cost value among a support region within the same disparity (Min, Lu, and Do 2011). The traditional cost aggregation is implemented by the construction of support regions obtained by a similarity function that can measurement the similarity between two potentially related pixels in the same reference image (Yang 2012). Yoon and Kweon et al. proposed an adaptive support region approach whose similarity function can be interpreted as a variant of joint bilateral filtering (Yoon and Kweon 2006). Cross-based approaches use a shape-adaptive window which consists of multiple horizontal lines spanning adjacent vertical rows based on the function of the color similarity and an implicit connectivity constraint (Zhang, Lu, and Lafruit 2009). A more thorough overview of cost aggregation methods can be found in (Min, Lu, and Do 2011). Most traditional methods, however, are limited by the shallow, hand-designed similarity function which cannot adequately build the support region for the cost aggregation. The usage of deep neural networks for cost aggregation can collaborate with deep cost computation methods in a trainable manner.

With the superiority of the two-stream architecture (Simonyan and Zisserman 2014; Feichtenhofer, Pinz, and Zisserman 2016), we propose an explicit learning-based cost aggregation. In this paper, we reformulate the cost aggregation process as the generation and selection of cost aggregation proposals. The proposals are obtained by generating potential cost aggregation results from the initial cost volume. The selection of proposals uses the structure information as global view guidance in a winner-take-all (WTA) strategy.

Table 1: Architecture for Feature Extraction

Index	layer	output
1	$5 \times 5 \times 32$ stride 2	$1/2H \times 1/2W \times F$
2-17	$3 \times 3 \times 32$ stride 2 residual connection*8	$1/2H \times 1/2W \times F$
18	$3 \times 3 \times 32$ stride 2	$1/2H \times 1/2W \times F$

Network Architecture

As a pixel-wise matching task, stereo matching is required to compute similarities between each pixel in the left image with D corresponding pixels in right image, where D is the maximum disparity. The computed matching cost can form the cost volume $C_0(h, w, d)$. The stereo matching pipeline with the proposed cost aggregation is carried out by an end-to-end trainable network. Compared with using networks as a black box, we take experience from classical stereo matching algorithm (Scharstein and Szeliski 2002) to conduct the cost aggregation explicitly by a two-stream network. In this paper, unless otherwise specified, we refer to the left image as the reference image and the right image as the target image, where the disparity is computed from the reference image.

The overview of our method is illustrated in Figure 2. The matching cost computation can be divided into three parts: feature extraction, volume construction and similarity computation, as shown in Figure 2.A1, Figure 2.A2 and Figure 2.A3, respectively. The detailed volume construction method is elucidated in Figure 3. A two-stream network carries out the proposed learning-based cost aggregation: the proposal network and the guidance network which are illustrated in Figure 2.B2 and Figure 2.B1, respectively. The disparity computation is shown in Figure 2.C, the detailed implementation of C will be discussed later in this section.

Matching Cost Computation

Matching cost computation is designed to compute the similarity between corresponding pixels at the reference image and the target image. The disparity map can then be obtained

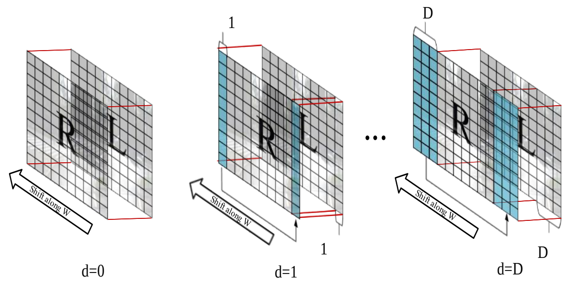

Figure 3: The operation for the feature volume construction. Each grid square represents the feature for the pixel. We can simply employ a shift and concatenation operation to form a feature volume.

from the cost volume. To determine the pixel-wise matching cost, we firstly generate deep representations for each pixel using a residual Siamese network. Then outputs from the Siamese network is fed into the feature volume construction which can transform features into the volume. Finally, the similarity measurement using 3D auto-encoder is applied on the feature volume to compute the matching cost volume.

A1: Feature Extraction To compute the similarity between two pixels, we require a powerful representation for each pixel. Compared with the traditional raw pixel intensities, deep feature representation is more effective and robust to mitigate textureless regions and thin structure. As shown in Figure 2.A1, we describe a Siamese network to extract the feature of each pixel. The Siamese network consists of two shared-weight sub-networks which concurrently deal with two input images. Each sub-network is composed of several residual blocks each of which consists of two 3×3 convolutional layers. To reduce the computational demand, we apply a 5×5 convolutional layer with 2×2 stride as a subsampling operation before the residual connection. For each residual block, it is activated before the residual operation. Each convolutional layer is followed by a batch normalized layer and a rectified linear unit except the last layer. From the detailed layer setting shown in Table 1, we can see that the result of the Siamese network produces two $H/2 \times W/2 \times F$ feature maps, where H and W denotes original input images size and F indicates the filter channel. The two feature maps contain the deep feature for each pixel in the reference image and the target image, respectively.

A2: Feature Volume Construction Obtained the representation of each pixel, the next step is to compute the similarities between pixels. Since the volume input can be effective for the 3D convolutional computation, we transform the extracted features into a feature volume which contains the underlying group of pixels. Each element of the feature volume represents the feature for computation of the similarity between two pixels. Because input images have been rectified, we can simply employ a shift operation to form the feature volume. We set the output of the left sub-network as the base feature and the output from the right as the shift feature. The base feature is awaiting to be concatenating at the bottom, and the shift feature slides on the base feature. As depicted in Figure 3, the shift feature slides on base feature and concatenates with the base feature along feature channel. The mathematical definition is given by

$$F(d,h,w) = B(h,w) \oplus S(d,h,(w+d) \bmod w), \qquad (1)$$

where B represents the base feature, S denotes the shift feature and $\oplus$ indicates the concatenating operation. After packing the concatenating results, we get a 4D feature volume of $D \times H/2 \times W/2 \times 2F$ size, where D denotes the maximum disparity.

A3: Similarity Computation The matching cost is designed to compute the similarities of corresponding pixels. The key of cost computation is the similarity measurement between two pixels. As we have obtained the feature volume, we expect to learn a similarity measurement as

$$C = T(F), \qquad (2)$$

which is designed to transform the feature volume into a cost volume. Each element of the cost volume represents the similarity computed from the corresponding element of the feature volume.

3D convolutional networks are effective to take into account the context and geometry information and operate computation from the height, width and disparity three dimensions (Kendall et al. 2017). However, the 3D convolutional operation commonly suffers from the burden on both computational time and intermediate results storage. With

Table 2: Architecture for cost computation. Each layer except layer 37 is followed by batch normalization and ReLU. Layer 33-37 are 3D-deconvolutional layers.

Index	layer	output
input	Volume Construction	$1/2D \times 1/2H \times 1/2W \times 2F$
19	$3 \times 3 \times 3 \times 32$ stride 1	$1/2D \times 1/2H \times 1/2W \times F$
20	$3 \times 3 \times 3 \times 32$ stride 1	$1/2D \times 1/2H \times 1/2W \times F$
21	$3 \times 3 \times 3 \times 64$ stride 2	$1/4D \times 1/4H \times 1/4W \times 2F$
22	$3 \times 3 \times 3 \times 64$ stride 1	$1/4D \times 1/4H \times 1/4W \times 2F$
23	$3 \times 3 \times 3 \times 64$ stride 1	$1/4D \times 1/4H \times 1/4W \times 2F$
24	$3 \times 3 \times 3 \times 64$ stride 2	$1/8D \times 1/8H \times 1/8W \times 2F$
25	$3 \times 3 \times 3 \times 64$ stride 1	$1/8D \times 1/8H \times 1/8W \times 2F$
26	$3 \times 3 \times 3 \times 64$ stride 1	$1/8D \times 1/8H \times 1/8W \times 2F$
27	$3 \times 3 \times 3 \times 64$ stride 2	$1/16D \times 1/16H \times 1/16W \times 2F$
28	$3 \times 3 \times 3 \times 64$ stride 1	$1/16D \times 1/16H \times 1/16W \times 2F$
29	$3 \times 3 \times 3 \times 64$ stride 1	$1/16D \times 1/16H \times 1/16W \times 2F$
30	$3 \times 3 \times 3 \times 128$ stride 2	$1/32D \times 1/32H \times 1/32W \times 4F$
31	$3 \times 3 \times 3 \times 128$ stride 1	$1/32D \times 1/32H \times 1/32W \times 4F$
32	$3 \times 3 \times 3 \times 128$ stride 1	$1/32D \times 1/32H \times 1/32W \times 4F$
33	$3 \times 3 \times 3 \times 64$ upsampling stride 2	$1/16D \times 1/16H \times 1/16W \times 2F$ add output of layer 29
34	$3 \times 3 \times 3 \times 64$ upsampling stride 2	$1/8D \times 1/8H \times 1/8W \times 2F$ add output of layer 26
35	$3 \times 3 \times 3 \times 64$ upsampling stride 2	$1/4D \times 1/4H \times 1/4W \times 2F$ add output of layer 23
36	$3 \times 3 \times 3 \times 32$ upsampling stride 2	$1/2D \times 1/2H \times 1/2W \times F$ add output of layer 20
37	$3 \times 3 \times 3 \times 1$ stride 1	$D \times H \times W \times 1$

the auto-encoder structure, the computational burden can be reduced by subsampling and upsampling operations.

The illustration of the auto-encoder with 3D convolutional layers is presented in Figure 2.A3 and layer setting is shown in Table 2. We apply four sub-sampling units as the encoder and four up-sampling units as the decoder. For the encoder, each sub-sampling unit consists of three 3D-convolution layers and the first convolution layer is applied with $2 \times 2 \times 2$ stride. For the decoder, the up-sampling unit is realized by one 3D convolution layer with $2 \times 2 \times 2$ stride, besides, the convolution output adds the same resolution feature map from the last layer of the corresponding sub-sampling unit in the encoder.

Since we apply a sub-sampling in feature extraction, to reach the same resolution as the original image, we add an extra up-sampling unit with a single convolution layer. The final output of cost computation is a cost volume with size of D, H, W and each element $C(d, h, w)$ in the volume indicates the matching cost between pixel $R(h, w)$ in the reference image and pixel $T(h, w - d)$ in the target image.

Cost Aggregation

The cost aggregation method is employed to rectify the mismatching cost value computed from the local feature according to the global view guidance. Besides, the cost aggregation can ensure a high-quality disparity map with smoothness and continuity. Through the matching cost computation, we get the initial cost volume $C_0(D, H, W)$. In general, the cost aggregation generates support regions on the cost volume by a statistic or dynamic similarity function. Obtained the regions, the aggregating can be formulated as the convolutional operation on the cost volume, which is expressed as

$$C(d, h, w) = W(d, h, w) \otimes C_0(d, h, w), \tag{3}$$

where W represents filters and $\otimes$ indicates the convolutional operation.

Compared with the traditional cost aggregation using hand-designed similarity measurement, we propose a learning-based cost aggregation using a two-stream network. The proposed cost aggregation can be directly employed on the initial cost volume and cooperate with the deep cost computation network in an end-to-end trainable manner. Instead of using the deep neural network as a black box, we present an explicit way to leverage the neural network. The cost aggregation is formulated as the selection of cost aggregation proposals, where proposals are potential cost aggregation results. As a result, the two-stream network is designed: one stream for generating the cost aggregation proposals, the other for selecting the best proposals. The proposal stream uses a 3D convolutional network to produce possible cost aggregation results. The results maintain the large receptive field and the rich semantic information transferred from cost computation. The guidance stream directly extracts information from the reference image with a 2D convolutional network. A light convolutional network is employed to extract the low-level structure information as the global view guidance for the selection.

Algorithm 1: Deep Cost Aggregation

```
Input: Initial Cost Volume $C_0(d, h, w)$
       Reference Image $I(h, w, 3)$
Output: Aggregated Cost Volume $C_a(d, h, w)$
1  \∗ Generation of proposals ∗\
2  Step 1: Aggregation along depth dimension:
     $C_d(d, h, w, g) = C_0(d, h, w, 1) \otimes F_d$ ;
3  Step 2: Aggregation along height dimension
     $C_h(d, h, w, g) = C_d(d, h, w, g) \otimes F_h$ ;
4  Step 3: Aggregation along width dimension
     $C_w(d, h, w, g) = C_h(d, h, w, g) \otimes F_w$ ;
5  Step 4: Normalization of aggregation proposals
     $C_p(d, h, w, g) = C_w(d, h, w, g) \otimes F_0$ ;
6  \∗ Extraction of Guidance for Cost Aggregation ∗\
7  Step 5: $G_0(h, w, g) = I(h, w, 3) \otimes F_0$;
8  Step 6: $G_1(h, w, g) = G_0(h, w, g) \otimes F_1$;
9  Step 7: $G_2(h, w, g) = G_1(h, w, g) \otimes F_2$;
10 \∗ Fusion and Selection ∗\
11 Step 8: Fusing the two output from the two-stream netwok:
12 $C_f = C_p(d, h, w, g) \odot G_2(h, w, g)$
13 Step 9: Choosing the best evaluated proposal:
     $C_a(d, h, w) = max\,\{C_f(d, h, w, g)\}$
```

Many works on understanding deep neural networks (Mahendran and Vedaldi 2015; Zeiler and Fergus 2014) have found that features of the first several convolutional layers are rich in low-level structure information. In contrast, the features from the last several layers have strong high-level semantic information. Both the structure and semantic information is crucial for the cost aggregation. The proposal stream maintains the semantic information, while the guidance stream brings into structure information. The rich semantic information is implicit in the generated proposals, and the structure information is used as global view guidance to evaluate each proposal. The cost aggregation is explicitly carried out by the fusion of these two streams. The details of our two-stream network will be discussed in the following two sub-sections.

B1: Proposal Sub-network The proposal stream is designed to generate the possible cost aggregation results by aggregating matching cost values along the height, width, and depth three dimensions. The aggregating operation is implemented by a 3D convolutional network with rectangle filters. The 3D convolutional network maintains the large view field from the previous cost computation step. The structure of the proposal sub-network is illustrated in Figure 2.B2. Three 3D convolutional layers are adopted to the initial cost volume. We first use $3 \times 1 \times 1$ convolutional filters to aggregate the cost values along the depth dimension, then employ $1 \times 3 \times 1$ and $1 \times 1 \times 3$ filters along the height and width dimensions. The rectangle convolutional filters are used to simulate the cost value aggregation process along different dimensions. Compared with the general square filters, the rectangle filter can run in a more explicit manner to aggregate information along different dimensions while actively reduce the computational burden for the 3D convolutional operation. Finally, a convolutional layer with $1 \times 1 \times 1$ filter is employed to summarize the potential cost aggregation results into G potential aggregation proposals with the size of $D \times H \times W \times G$, where G represents the number of cost aggregation proposals.

The operation along one dimension can be expressed as

$$C(d,h,w) = F_i(d,h,w) \otimes C_0(d,h,w), \tag{4}$$

where F represents the rectangle filters, i donates the convolutional direction, and $\otimes$ indicates the convolutional operation.

B2: Guidance Stream Since proposals are computed from features of the last layer which has strong semantic information but lacks low-level structure information. The guidance stream is designed to introduce the structure information as the global view guidance to the selection of proposals. It can extract structure information from the reference image to evaluate the generated proposals. As shown in Figure 2.B1, we employ 2D convolutional network on the reference image to extract the low-level structure information. The convolutional filter is set from 5×5 to 3×3 which can equip the structure information with a large field of view. Moreover, a final 1×1 filter is employed to summarize the guidance to the size of $H \times W \times G$ corresponding to the generated proposals. Furthermore, the guidance is converted into probability value using the softmax method along the dimension of G, which ensures that the sum of the evaluation of the proposals is 1. Since we hypothesize the guidance for cost aggregation at different disparities is unchanged, the computed probability value can be treated as the evaluation for different aggregation proposals. The guidance $G_2(H,W,i)$ is used as the evaluation for the proposal $C_a(D,H,W,i)$.

In the end, the selection of proposals is achieved by a fusion scheme. The fusion uses the guidance to evaluate the proposals and choose the best evaluation of the fusion results to form the aggregated cost volume. The global view guidance evaluates its corresponding aggregation proposal by a matrix multiplication in a broadcasting manner. The evaluation for each proposal is based on the structure information of the whole reference image so the guidance for the selection is global view. The aggregated cost volume can be obtained by selecting the maximum value along the dimension of G. The fusion scheme is indicated as

$$C_a(d,h,w) = max\left\{C_p(d,h,w,g) * C_g(h,w,g)\right\}, \tag{5}$$

where C_p are proposals, C_g represents the guidance, $*$ donates the matrix multiplication and max indicates the maximum function. The process of conducting the cost aggregation algorithm is shown in Algorithm 1.

C: Disparity computation

The aggregated cost volume will be transformed into disparity through a soft-argmin function similar to (Kendall et al. 2017) which can retain a sub-pixel disparity accuracy. The matching cost value is converted into probability value by a softmax function along the dimension of depth. The final disparity is obtained by the weighted sum of the probability, where the weights are the corresponding depth value d. The mathematical equation is given by

Table 3: Comparisons on Scene Flow

Model	error > 1px	error > 3 px	MAE(px)	T(ms)
GC-Net	11.3	7.2	2.21	0.95
Without guidance	12.3	7.2	2.15	0.93
Without proposal	10.81	6.8	1.83	0.85
Without aggregation	13.8	7.5	2.71	0.95
Our model	**8.93**	**5.62**	**1.75**	**1.12**

$$D(h,w) = \sum_{d=0}^{D_{max}} d \times \sigma(-C_a(d,h,w)), \tag{6}$$

where σ donates the softmax function, C_a is the aggregated cost volume and d is the disparity.

Compared with the traditional WTA strategy, the soft-argmin function can enable the computed disparity influenced by the cost value of all disparity. Therefore, a better smoothness and sub-pixel level accuracy can be obtained. Besides, the soft-argmin is fully differentiable, which ensures that the training can be carried out using back-propagation.

We train the model using the ℓ_1 error between the ground truth and the predicted disparity. The supervised loss is defined as

$$Loss = \sum_h \sum_w \left\| D_a(h,w) - D_g(h,w) \right\|_1, \tag{7}$$

where $\|\cdot\|_1$ donates the ℓ_1 norm, D_g is the ground truth disparity map and D_a represents the predicted disparity map.

Experimental Results

We evaluate our method on three datasets, including Scene Flow (Mayer et al. 2016), KITTI2015 (Menze and Geiger 2015) and KITTI2012 (Geiger, Lenz, and Urtasun 2012). We especially compare our method with the state-of-the-art GC-Net (Kendall et al. 2017) to demonstrate the effectiveness of the learning-based cost aggregation. Our architecture is implemented by the Tensoflow (Abadi et al. 2016) with a standard RMSProp (Tieleman and Hinton 2012) and a constant learning rate of 0.0001. We train the network on the Scene Flow dataset from a random initialization with shuffled orders. The training takes 23h after 300K iterations on a single NVIDIA 1080Ti GPU. For the KITTI dataset, we fine-tune the model pre-trained on Scene Flow dataset with 70k iterations. Limited by the computation resource, we sub-sample all data by four times using the bilinear interpolation.

Benchmark results

Scene Flow is a synthetic data set for stereo matching which contains 35454 training and 4370 testing image pairs. Synthetic dataset ensures dense ground truth without inaccurate labels and is large enough to train a complex network without over-fitting. In Table 3, we evaluate our method and GC-Net on the Scene Flow dataset. We observe that our method outperforms GC-Net among all pixel errors and the RMS error. In addition, to demonstrate the effectiveness of each

Table 4: Comparisons on KITTI2012

Model	>2px		> 5 px		Mean Error		T(s)
	Non-Occ	All	Non-Occ	All	Non-Occ	All	
PSMNet	**2.62**	**3.24**	**0.94**	**1.20**	**0.5**	**0.6**	1.3
GC-Net	2.71	3.46	1.77	2.30	0.6	0.7	0.9
SegStereo	3.24	3.82	1.10	1.35	0.6	0.6	0.6
Displets v2	3.43	4.46	1.72	**2.17**	0.7	0.8	265
L-ResMatch	3.64	5.06	**1.50**	2.26	0.7	1.0	48
MC-CNN	3.90	5.45	1.64	2.39	0.7	0.9	67
CATN	8.11	9.44	3.31	4.07	1.1	1.2	10
S+GF	14.72	16.76	5.53	7.79	2.1	3.4	140
Our model	2.68	3.42	1.63	2.23	0.6	0.7	1.13

Table 5: Comparisons on KITTI2015

Model	All pixels			Non-Occluded Pixels			Time(s)
	D1-bg	D1-fg	D1-all	D1-bg	D1-fg	D1-all	
PSMNet	**1.97**	4.41	**2.38**	**1.81**	4.00	**2.17**	1.3
SegStereo	2.16	**4.02**	2.47	2.01	**3.62**	2.28	**0.6**
GC-Net	2.21	6.16	2.87	2.02	5.58	2.61	0.9
MC-CNN	2.89	8.88	3.89	2.48	7.64	3.33	67
Displetv v2	3.00	5.56	3.43	2.73	4.95	3.09	265
DRR	2.58	6.04	3.16	2.34	4.87	2.76	0.4
L-ResMatch	2.72	6.95	3.42	2.35	5.74	2.91	48
3DMST	3.36	13.03	4.97	3.03	12.11	4.53	93
Our model	2.17	5.46	2.79	2.06	5.32	2.32	1.13

stream of our network, we evaluate the network with different settings. From Table 3, we can see the guidance stream is crucial to improving the performance, which demonstrates the structure information can be used as global view guidance to improve the accuracy.

The KITTI benchmark consists of challenging and complex road scene collected from a moving vehicle. The ground truth of disparity image is obtained from LIDAR data. The KITTI 2012 dataset contains 192 training and 195 testing images, and the KITTI 2015 dataset contains 200 training and 200 testing images. In the Table 4, the comparisons on KITTI2012 with deep stereo methods such as GC-net (Kendall et al. 2017), Displets v2 (Guney and Geiger 2015), L-ResMatch (Shaked and Wolf 2016) and MC-CNN (Zbontar and LeCun 2015) are shown, besides, the comparisons with other cost aggregation methods including CAT (Ha et al. 2014) and S+GF (Zhang et al. 2014) are also illustrated in Table 5, the leaderboard on KITTI2015 compares our method with GC-Net (Kendall et al. 2017), MC-CNN (Zbontar and LeCun 2016), Displetv v2 (Guney and Geiger 2015), DRR (Gidaris and Komodakis 2016), L-ResMatch (Shaked and Wolf 2016) and 3DMST (Li et al. 2017). Our method outperforms previous works which use a hand-designed aggregation method or ignoring the aggregation step. It can be inferred that the usage of learning-based cost aggregation method can improve the performance of the deep stereo matching.

Sub-architecture Analysis

To demonstrate the effectiveness of the learning-based cost aggregation, we visualize the guidance obtained from the guidance stream in this section. According to the visualization of the Figure 4, we can infer that the guidance stream can obtain the structure information from reference image which can select the aggregation proposal with a global view. The visualized feature map of guidance subnetwork is realized by averaging the output of the guidance stream along the dimension G. We can obviously see the guidance contains low-level structure information, which demonstrates that the two-stream network can introduce structure information as the global view guidance for the selection of proposals.

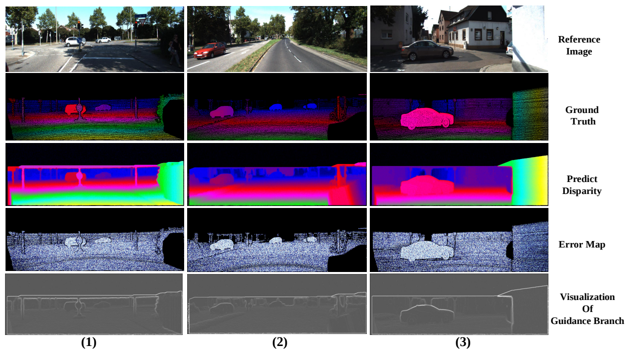

Figure 4: The visualization of our experimental results. From top to bottom, images are the reference image, ground-truth disparity map, predicted disparity map, error map and the visualization of the output from our guidance stream, respectively. The visualization of the guidance stream shows that it exactly extracts structure information.

Conclusion

In this paper, we have proposed a learning-based cost aggregation for stereo matching. The learning-based cost aggregation can be embedded into the deep stereo matching solution in an end-to-end manner. With this end-to-end trainable manner, our cost aggregation achieved a higher accuracy by effectively collaborating with the deep cost computation methods. According to the analysis of the two-stream network, we demonstrated that the low-level structure information can be used as global view guidance for selection of the proposals of the rich semantic information. Furthermore, the proposed two-stream network had the potential ability for feature fusion works such as motion recognition and scene understanding. The experiment results have demonstrated the good ability of our explicit architecture for stereo matching.

Acknowledgement

This work was supported in part by the Natural Science Foundation of China (NSFC) under Grants No. 61773062 and No. 61702037.

References

[Abadi et al. 2016] Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G. S.; Davis, A.; Dean, J.; Devin, M.; et al. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

[Feichtenhofer, Pinz, and Zisserman 2016] Feichtenhofer, C.; Pinz, A.; and Zisserman, A. 2016. Convolutional two-stream network fusion for video action recognition. In

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1933–1941.

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

[Gidaris and Komodakis 2016] Gidaris, S., and Komodakis, N. 2016. Detect, replace, refine: Deep structured prediction for pixel wise labeling. *arXiv preprint arXiv:1612.04770*.

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

[Ha et al. 2014] Ha, J.; Jeon, J.; Bae, G.; Jo, S.; and Jeong, H. 2014. Cost aggregation table: cost aggregation method using summed area table scheme for dense stereo correspondence. In *International Symposium on Visual Computing*, 815–826. Springer.

[Kendall et al. 2017] 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.

[Li et al. 2017] Li, L.; Yu, X.; Zhang, S.; Zhao, X.; and Zhang, L. 2017. 3d cost aggregation with multiple minimum spanning trees for stereo matching. *Applied Optics* 56(12):3411–3420.

[Luo, Schwing, and Urtasun 2016] 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.

[Mahendran and Vedaldi 2015] Mahendran, A., and Vedaldi, A. 2015. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5188–5196.

[Mayer et al. 2016] 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.

[Menze and Geiger 2015] 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.

[Min, Lu, and Do 2011] Min, D.; Lu, J.; and Do, M. N. 2011. A revisit to cost aggregation in stereo matching: How far can we reduce its computational redundancy? In *Computer Vision (ICCV), 2011 IEEE International Conference on*, 1567–1574. IEEE.

[Scharstein and Szeliski 2002] 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.

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

[Simonyan and Zisserman 2014] Simonyan, K., and Zisserman, A. 2014. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, 568–576.

[Tieleman and Hinton 2012] Tieleman, T., and Hinton, G. 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning* 4(2):26–31.

[Tombari et al. 2008] Tombari, F.; Mattoccia, S.; Di Stefano, L.; and Addimanda, E. 2008. Classification and evaluation of cost aggregation methods for stereo correspondence. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 1–8. IEEE.

[Xu, Ranftl, and Koltun 2017] Xu, J.; Ranftl, R.; and Koltun, V. 2017. Accurate Optical Flow via Direct Cost Volume Processing. In *CVPR*.

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

[Yoon and Kweon 2006] Yoon, K.-J., and Kweon, I. S. 2006. Adaptive support-weight approach for correspondence search. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28(4):650–656.

[Zbontar and LeCun 2015] 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.

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

[Zeiler and Fergus 2014] Zeiler, M. D., and Fergus, R. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*, 818–833. Springer.

[Zhang et al. 2014] Zhang, K.; Fang, Y.; Min, D.; Sun, L.; Yang, S.; Yan, S.; and Tian, Q. 2014. Cross-scale cost aggregation for stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1590–1597.

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