Learning-based Region Support for Stereo Matching

Anonymous ECCV submission

Paper ID 000

Abstract. The application of convolutional neural network achieves great success on the stereo matching. But most methods use neural networks in an implicit manner for stereo matching, which is difficult to sovle a particular challenge such as occlusion or textureless areas. In this paper, we propose the region support to explicitly handle these challenges in stereo matching. The region support is an extension of the traditional variable support, which is enhanced by a novel learning scheme. The learning-based region support is obtained by a novel coarse-depth inference network i.e. the region support network which infers the depth from a single image. In addition, we propose the improved cost computation, cost aggregation, and refinement methods, which are reformulated by region support.The final reformulated stereo matching pipeline reaches remarkable performance both on speed and accuracy. The experiments on SceneFlow and KITTI demonstrate the effectiveness of the region support. The code and network settings will be published online later.

1 Introduction

Stereo matching is one of the most active research areas in computer vision community. Currently, driven by the powerful neural networks, a lot of state-of-the-art stereo matching methods are proposed [5, 6, 45]. But most of these methods realize the stereo matching in an implicit manner, which means the improvement is mainly due to the design of network architecture itself. Although these methods improve the performance of stereo matching, they are hard to determine the solution for certain challenges such as occlusion, textureless or high-textured area and reducing search band in disparity space. The concept of traditional variable support is proven effective to deal with these challenges [1–4]. Taking the basic concept of variable support, in this paper, we propose the region support to explicitly sovle the challenges in stereo matching.

There are two keys of variable support for stereo matching, which are the determination of pixel sets and local relationships. For example, in cost aggregation, the two keys perform as determining pixel sets for each pixel to aggregate with and computing the aggregation weights according to local relationship [2, 7]. As for cost computation and refinement, the segment-based cost computation aims to determine the best division of pixel sets and filtering-based refinement tries to design filters which are able to describe the local relationship in a suitable area [8–10]. The problem is that challenges for these three step are

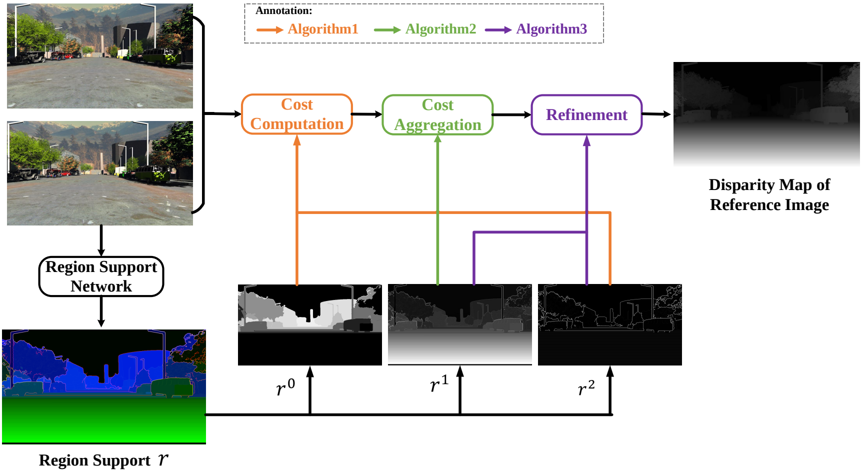

Fig. 1. The input stereo image pair is fed to a region support network to generate the region support r. The region support has three channel to help design the stereo matching. With the region support, the three steps: cost computation, cost aggregation, and refinement are reformulated by new strategies.

different, which makes the requirement for the two keys is inconsistent. Global or semi-global stereo matching methods are always designed on explicit constraints, where structural continuity are proven to be an effective constraint. To unify the different requirements by the structual constraint, we present the region support by an effective learning scheme which can infer structural information in disparity space.

To satisfy the requirements of all three steps, the region support propose new hypotheses for the aforementioned two keys. For the pixels in the same pixel set, pixels should share simialr disparity. Similarly, the local relationship should also be described directly according to the disparity relationship. The traditional determinations of these two keys are mainly based on the relationship of pixels in RGB space. But this can lead to a failure because the pixels which share similar colorful information are higly possible to have totally different disparity. Based on these two hypotheses in view of disparity space, the reformulated stereo matching method is able to leverage the region support as a coarse depth guidance to effectively and efficiently realize the stereo task.

The region support is shown in Figure.1(a). To obtain this kind of region support, we propose a novel neural network called the region support network (RSN) which can extract the instructive information from disparity space. This network takes a single image and generates the required region support. The ground-truth of the training process for the region support is obtained only from the original stereo disparity map according to the particular requirement from the stereo matching. During the testing process, the stereo matching method with the region support only takes into the stereo image pair, which can be treated as a coarse-to-fine strategy.

We reformulate the stereo matching method based on the region support. To test the effectiveness of usages of the region support for different step, we independently employ the region support in the three steps. By applying the region support for all three steps, the final stereo matching method reaches state-of-the-art results on SceneFlow and KITTI datasets.

The contributions of our work are two-fold:

- We propose a novel variable support called the region support for the stereo matching. In addition, we leverage the region support to present a novel stereo matching method with three reformulated steps.
- A novel deep neural network is proposed to obtain the region support. With the region support, a novel learning-based stereo matching method is presented.

2 Related Work

2.1 The Variable Support for the Local Stereo Matching

The variable support is a classical concept in the stereo matching [1, 7], which aims to determine the best set of pixels for the initialized matching result to aggregate with. Most of these variable support cost aggregation methods can be improved based on two different approaches. The former is allowing the set of pixels to have unconstrained shape. To obtain flexible shape, some classical works vary the window size and offsets or select more than one window [1, 11–13]. Meanwhile, some works use segment-based or cross-based methods to determine the set of pixels [14, 10, 15]. The latter assigns adaptive or dynamic weights to the pixels belonging to the set. Generally, the weights are computed based on the color and spatial relationship among the pixels in the same set [16–19]. These works demonstrate that the colorful and structural information in RGB space is useful to express the relationship in disparity space in a way.

As for cost computation, a lot of methods can be seen as the continuation of the variable support concept. The patchmatch and segment-based methods can always be treated as the determination of the sets of pixels sharing potential disparity [20, 9, 10]. Recent works using deep neural networks for cost computation also can be seen as the application of the variable support, where the pixel sets are determined by the receptive field of the networks and the weights are the learnable filters [5, 21, 6]. The results of these works show that a suitable receptive field is crucial to obtain the effective representation. A too large receptive field may generate a representation which is indistinguishable for the similarity measure, which can lead to a blurring result on edges. A too small receptive field is hard to handle the textureless and occluded areas.

With the respect to refinement methods, the filtering based methods always require a suitable filter setting, which can be inferred from the variable support [8]. To remove the outliers in textureless areas, the variable support can offer the guidance to describe the local relationship among the same set. The methods using slanted-plane model always need a pre-defined pixel sets [22, 23]. Compared to the randomized strategy like patchmatch, the segment-based or object-based methods are proved to be more effective [23, 22]. But the determination is always based on the inference in the RGB space. Displet [24] propose to use the estimation from disparity space, but the proposals are estimated based on the pre-defined 3D model, which is limited the generalization of the estimation from disparity space.

Global or semi-global stereo methods generally use energy function to iteratively optimize the stereo matching results [25–27]. The constraints in energy function are always defined based on smoothness or continuity, where the structural or geometry information is crucial. The region information like segmentation and occlusion detection results are proven to be effective for the smoothness constraint [28, 24].

In this paper, the determination of pixel sets and descriptor of local relationship is reformulated by a learning-based scheme. In addition, without the strong prior knowledge from pre-defined models, the learning process directly endows the region support with geometry information from disparity space. With the region support, a novel learning-based stereo matching method is designed, where all the three steps are reformulated by the region support.

3 Region Support for the Stereo Matching

3.1 Outline

In this section, we will focus on the discussion of specific requirement and employment of the region support. Eventually, the formulation of an effective region support is acquired for the stereo matching. In the following subsections, we find out requirements from all the three steps and figure out the commonness between them. In each subsection, we first determine the particular challenge to deal with and propose the novel solutions with the help of region support. The concrete realization of the region support will be discussed in Section. 4.

3.2 Cost Computation

Cost computation aims to generates a cost volume V from the reference image R and target image T, in which the element represents the matching cost of corresponding pixels. The whole process can be expressed as

$$V(w, h, d) = f(R(w, h), T(w - d, h)), \tag{1}$$

where the f indicates the cost computation function and w, h, d donates the position of the cost volume.

There are mainly two challenges in this process. The former is to generate a powerful representation for the pixels. The later is to design an effective similarity measure. Driven by the powerful representation from the deep neural

Algorithm 1: Cost Computation

```
Input: Support Regions $r^0$,Local Relationship Matrices $r^1$, Edges $r^2$
Output: Initial Cost Volume $V^i$
Step1:Generation of $r$ from the region support netowrk(RSN)
$r^0$,$r^1$,$r^2$= $RSN(R)$
Step2:Pre-Matching
$d^0$= $match(r^0)$
Step3:Representive Pixels
$e$,$o$= $select(r^0$,$r^2$,$d^0)$
Step4:Feature Extraction from Siamese Network
$F(e), F(o) = siamese(e, o)$
Step5:Matching Cost Computation
$V^i$=$f(F(e))$,$f(F(0))$
```

networks, the former challenge is highly resolved [6, 29, 30]. But it is still difficult to judge whether the view field is suitable. As for similarity measure, the widely used cosine distance reaches an efficient result [5, 21]. Using networks as similarity measure raises the accuracy but cut down the speed. In addition, the high computational burden of cost volume greatly limits the whole stereo matching method.

Requirement for Region Support To handle the aforementioned problems, we propose the region support to guide the cost computation process. First, it offers a more economical form to storage the matching results. Compared to the slanted plane based methods [23, 22], each pixel set i.e. support region determined by the region support consists of pixels at the similar disparity. As a result, we are able to just compute a subset of each pixel set. Then, it handles the view field problem by an additional division of pixels. For the pixels at the edges, the small view field is required, which is because a large view field will lead to a blurring matching result. In contrast, the pixels at textureless areas need a large view field, so it can capture the supportive information of objects. The region support divides these two-class pixels and feeds them to different neural network with different view field. Finally, it improves the similarity measure. Based on the cosine distance, the region support offers a local relationship descriptor for each pixel, which can be used as an additional local feature to enhance the deep feature.

Reformulation by Region Support The process of cost computation is shown in Algorithm.1. We first divide the reference image into support regions. Then, we generate representative pixels for each support region as the candidate subset for cost computation. This operation will lead to a sparse matching cost result, while the dense disparity is obtained in the refinement part of Section.3.4. Before generating the representation, we pre-match support regions to obtain a coarse region disparity. The pre-matching algorithm carries out using a template based shift matching. We shift each support each on the image to find the best matching one according to the rate of overlapping areas. Then this region dis-

Algorithm 2: Cost Aggregation

```
Input: Local Relationship Matrices $r^1$, Two Set of Pixels $e, o$,Cost Volume $V^i$
Output: Cost Volume $V^a$
while $d < D$ do
    $V^d = V^i(, , d)$
    for $t$ $in$ $[e, o]$ do
        for $p$ $in$ $t$ do
            %Determine Related Pixels by a Selection function $select$ based on $r^1$
            $P = select(p, r^1)$
            $V^d(p) = A(p, P) \odot V^d(P)$
        end
    end
end
```

parity is used to construct the corresponding pixels, which can highly reduce the search space. The matching pixel pairs at edge i.e. e and among objects i.e. o are fed into different networks to obtain the representation. Finally, we propose a novel similarity function f to compute the matching cost, which is expressed as

$$\begin{aligned} f(w,h,d) = \cos(F(R(w,h)), F(T(w-d,h))) \\ +\|U(R(w,h)) - U(T(w-d,h))\|_1, \end{aligned} \tag{2}$$

where F is the representation generation method and U is the local relationship descriptor.

3.3 Cost Aggregation

For each pixel on the cost volume, cost aggregation method aggregates the cost value of pixels related to this pixel with adaptive weights. Most methods determine the related pixels and adaptive weights on the basis of RGB space [31, 2, 15]. These methods assume that pixels which are similar in RGB space are also similar in disparity space. For some simple situations, this hypothesis reaches sufficient performance, but on high-texture areas, this can lead to a failure. In addition, the determination of related pixels and adaptive weights are highly computational expensive.

Requirement of Region Support Traditional hand-designed variable support cannot offer guidance from the disparity space, therefore, a learning mechanism is indispensable to infer the information in disparity space. With the region support, we simply employ the classical strategy for cost aggregation. But aggregating with all related pixels for each pixel brings into the redundant computation. As we have discussed above, the region support generates representative pixels and local descriptor. We can just conduct the aggregation on the representative pixels to reach a sufficient performance. It is worth pointing out that for most cost aggregation methods, we only focus the two-dimensional cost aggregation which is based on the fronto-parallel. In summary, the region support determines the related pixels with unconstrained shape and adaptive weights for the cost aggregation.

Algorithm 3: Refinement

```
Input: Local Relationship Matrices $r^1$, Edges $r^2$, Pixel Sets $e, o$,Initial Disparity Map $W^i$
Output: Disparity Map $W^r$
Step1:Remove Outliers
$V^d = V^i(,,d)$
for $p$  in  $[e + o]$ do
    $P = select(p, r^1)$
    if $|W^i(p) - average(W^i(P))| > thres$ then
        $W^i(p) = 0$
    end
end
Step2:Determination of Unmeasured Pixels
$e^r = near(e)$, $o^r = near(o)$
Step3:Interpolation
for $p$  in  $e^r$ do
    $P = nearest4(p, r^1, r^2)$
    $W^i(p) = bilinear(P)$
end
for $p$  in  $o^r$ do
    $P = nearest2(p, r^1, r^2)$
    $W^i(p) = bilinear(P)$
end
Step4:Final Refine
for $t$  in  $[e^r, o^r]$ do
    for $p$  in  $t$ do
        $P = nearest4(p, r^1)$
        $W^i(p) = average(P)$
    end
end
```

Reformulation by Region Support The cost aggregation with the region support is shown in Algorithm.2. The cost aggregation is applied iteratively along the depth($\mathbb{D}$) dimension, which means we conduct the two-dimensional cost aggregation D times for each support region. The cost aggregation is only applied to the representative pixels of support regions. The adaptive weights are computed based on the region support, which can be shown as

$$A(x, x) = 1 - \lambda \left\| r^1(x) - r^1(x^{'}) \right\|_1, \tag{3}$$

where x and $x^{'}$ represent the pixels need to aggregate with and the λ is defined by

$$\lambda = \begin{cases} \left| r^1(x) - r^1(x^{'}) \right| & if\, r^0(x) = r^0(x^{'})\, and\, r^2(x) = r^2(x^{'}) \\ 0.3 \times \left| r^1(x) - r^1(x^{'}) \right| & if\, r^0(x) = r^0(x^{'})\, and\, r^2(x) \neq r^2(x^{'}) \\ 0 & if\, r^0(x) \neq r^0(x^{'}) \end{cases} \tag{4}$$

3.4 Refinement

The refinement methods generally use filtering, left-right (L-R) check and interpolation to reach the accurate sub-pixel level disparity map. The filtering based methods use a static geometry distribution to remove outliers. The interpolation methods or slanted plane methods infer the linear relationship in disparity space. The key of refinement method is to adequately infer the underlying relationship in disparity space.

Requirement of Region Support Because of the proposed cost computation method, the obtained disparity map is in a sparse form, which requires an interpolation method to obtain the dense disparity map. For RGB images, the interpolation needs to ensure the smoothness and continuity at the same time keeping the high-textured areas. As for disparity images, the variation depth is much more straightforward and the inference can be generally summarized into two situations. The former is the pixels among object i.e. o, where the depth variation is continuous in all directions. The inference based on distance measure can ensure a continuous and smooth performance. The latter is the pixels at edges i.e. e, where obtaining the direction of the edges and the related pixels is crucial because continuity is only kept along the direction of edges. The spare form solves the occlusion problem in a way because even the pixels are occluded, the continuity remains locally. In addition, the region support removes the outliers before interpolation by the local relationship.

Reformulation by Region Support The operation of refinement is shown in Algorithm.3. We first remove the outliers according to the local relationship from the region support. Then we separately carry out interpolation following e and o. For the unmeasured pixels related to e, we find the four nearest measured pixels to interpolate the value. For the other unmeasured pixels, we simple interplate the value by the two nearest measured pixels on the horizontal direction. Finally, we refine the interpolation value by averaging each pixel with the four nearest measured pixels.

4 Region Support Network

4.1 Formulation

From the discussion of the last section, we can see the inference in disparity space is crucial for the region support, which is extremely difficult to achieve by hand-designed manner. In this paper, we propose a region support network (RSN) to learn the required region support.

The region support needs to determine the support regions in which pixels share similar disparity. In addition, the region support divides the pixels into two classes: the pixels at edges and the pixels among the object. Besides, the region support generates a local descriptor for each pixel.

The proposed network is presented in Section.4.2. The RSN takes the reference and target image separately and generates the region support for both images. To equip the support regions with the required attributes, we propose a novel loss function shown in Section.4.4.

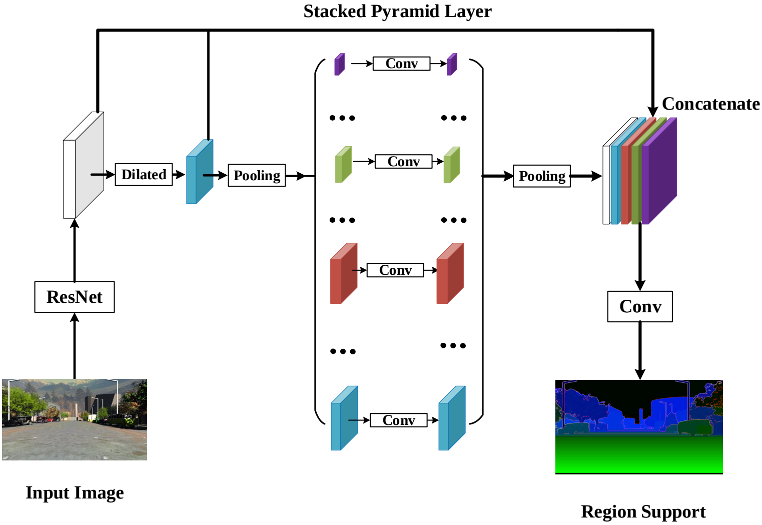

Fig. 2. The Support Region Network. First, we extract features from a residual network and dilated network. Then the sub-sampled feature maps are fed into a stacked pyramid layer to extract multi-scale features. After that, the up-sampling and concatenating operation is employed to get the final feature map. Finally, several convolutional layers with softmax function are employed to get the region support.

4.2 Region support Archetecture

The overview of the region support network is shown in Fig.2. First, we use a residual network [32] to extract the local feature for each pixel with original resolution. Then we use dilated strategy to obtain a larger view field for the feature map [33, 34] and meanwhile reduce the resolution to 1/4. After that, we use a stacked pyramid network [35] to conduct the inference in disparity space. The obtained local feature map is sub-sampled into six resolution level of $1/16, 1/32, 1/64, 1/128, 1/256$ and $1/512$ by an average pooling operation. Acquired with the feature map with different resolution, we use a bilinear interpolation operation to up-sample all the feature into original resolution. After fixing the resolution problem, we concatenate all the feature maps and feed the concatenated feature map through two convolutional layers to get the final deep representation for the region support. The layer setting is shown in **Supplementary Material**.

The output of the region support network is r, which has three channels. The first channel is the index for support regions. To obtain this prediction, the representation is fed to a softmax function to get the classification result. Before the softmax, a 1*1 convolutional layer is applied to adjust the channel number. The second is the detailed classification in a certain support region, which indicates the local relationship. We separately apply 1*1 convolutional layer with softmax function for each support region determined by channel one. The third channel donates whether the pixel is at the edge. We add the results of channel one and two to form the initial region support. Then we apply three convolutional layers with an additional three-class softmax to determine the edges at support regions, edges among support regions and others.

4.3 The Stereo Matching Network With Region Support

Using the algorithms proposed in Section.3, we conduct the whole stereo matching with the learning-based region support. We reuse the residual network of the region support network to extract the feature for stereo matching by a siamese structure. The input is a pair of patches which is centered at the representative pixel determined by the region support. The cost aggregation is independently carried out following the order of support regions. Before the refinement, we use a soft-argmin strategy [6, 30] to compute the disparity value. After that, we gain a sparse disparity map, and then the final refinement is carried out by the proposed interpolation operation on the sparse disparity map. The detailed implementation is shown in **Supplementary Material**.

4.4 Training

The training process of our stereo matching method is achieved separately by training the proposed two networks. First, we train the region support network with a novel supervised joint loss function:

$$L_S = L_R + L_r + L_E, \tag{5}$$

$$L_R = \frac{1}{N} \sum_{n=1}^{N} \hat{r}_n^0 \log(r_n^0), \tag{6}$$

$$L_r = \frac{1}{S} \sum_{s=1}^{S} \frac{1}{M} \sum_{m=1}^{M} \hat{r}_m^1 \log(r_m^1), \tag{7}$$

$$L_E = \frac{1}{N} \sum_{n=1}^{N} \hat{r}_n^2 \log(r_n^2). \tag{8}$$

r_n represents the computed the region support and the $\hat{r}_n$ represents the ground-truth for training. The N represents the number of pixels of the whole image, S

represents the number of support regions and M represents the number of pixels among a particular support region.

After obtaining the region support, we hold still the region support network and train the proposed stereo matching method simply by a regression loss,

$$L_D = \frac{1}{N}\sum_{n=1}^{N}\left\|d_n - \hat{d_n}\right\|, \tag{9}$$

where d_n indicates the disparity of the pixel. During this training process, the region support is obtained from the ground-truth of the region support. In the end, we jointly train the whole stereo matching method by $L_S + L_D$ by replacing the region support by the result from the region support network.

5 Experiments

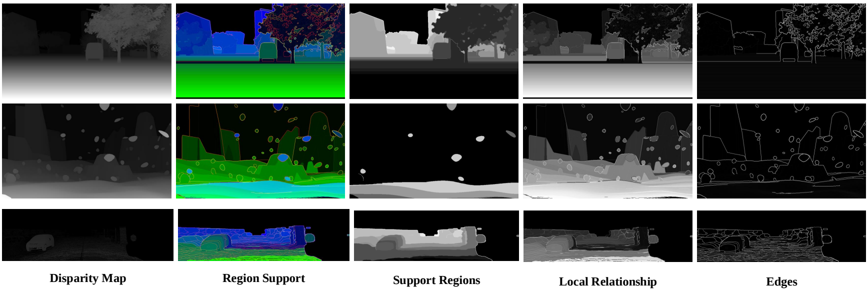

Fig. 3. The region support in SceneFlow and KITTI. The first two rows are the region support in SceneFlow. The Third row is the region support in KITTI. We compare the region support with the disparity map, we can see the region support network actually infers the information in disparity space. The support regions, local relationship and edges are the three channels of region support, which are the reuqired instructive information for the stereo matching.

In this section, we persent qualitative and quntitative results to demonstrate the effectiveness of the region support. Firstly, in Section.5, we show the construction of the region support from disparity map, which provides the ground-truth to train the region support network. In Section.5.2, we compare the results of our approach with the state-of-the-art methods on KITTI [36, 37] and SceneFlow [38]. Finally, we figure out the effectiveness of the region support for each steps in Section.5.3

The proposed the region support network and stereo matching method are implemented by PyTorch [39]. All models are trained by the Adam Optimization[40]. The learning rate is initialized by 0.01 and reduce by a half after each

Table 1. Comparisons on the SceneFlow

Method	$> 1px$	$> 3px$	$> 5px$	EPE	Param.	Time(ms)
SceneFlowNet [38]	–	–	–	7.87	–	**60**
MC-CNN-fst [44]	–	–	–	10.12	–	800
SGM [25]	–	–	–	23.01	–	1100
GC-Net [6]	16.9	9.34	7.22	2.51	3.5M	950
iResNet [45]	**9.28**	**4.57**	3.32	2.45	43.34M	148
Ours	10.26	4.93	**3.02**	**2.37**	**2.7M**	112

epoch. We train the region support network with batchsize 1. All of the images are normalized into 0-1 intensity. For each dataset, we train 8 epoches for the region support network, 5 epoches for the stereo matching network and 2 epoches for the joint training. The training of KITTI dataset uses the pre-trained model on the Driving dataset. The detailed setting of network and hyper-parameter setting are shown in **Supplementary Material**.

5.1 Training Data Preparation

The obtained region support is shown in Fig.3. The region support has three channels, which are support regions, local relationship, and edges. These three instructive guidances are required by the stereo matching methods. Since the region support is learning-based, for the training purpose, we extract the region support from the disparity map. During training, the obtained region support from disparity is used as the ground-truth of the RSN and the initial region support for the stereo matching network. After training the region support network, we replace the region support for stereo matching with the learned region support from the RSN.

To obtain the ground-truth, we extract the region support The first channel is the support regions, each of which is consisted of pixels in similar disparity. We apply the Felzenszwalbs efficient graph-based segmentation [41] to the disparity map to obtain the initial support regions. After that, we apply an aggregation operation between the regions according to the disparity continuity. Then we divide the aggregation results into 32 sets according to the average disparity, which is the final support regions. The second channel is the local relationship. It is the combination of the results of original Felzenszwalbs segmentation and Sobel edges detection [42]. The third channel is the edges in disparity space, which is obtained by the canny edge detection and Sobel detection [43].

5.2 Benchmark Results

We test our stereo matching method on two datasets: SceneFLow [38] and KITTI [36, 37]. The SceneFlow is a synthetic dataset which consists of three datasets Driving, FlyingThings3D and Monkaa. These three datasets are constructed in different scenes. The Driving dataset is a mostly naturalistic street scene from

Table 2. Comparisons on the FlyingThings3D

Method	3PE	EPE	Param.	Time(ms)
CRL [46]	6.20	1.32	78.77M	162
SGM [25]	12.54	4.50	–	–
MC-CNN-fst [44]	13.70	3.79	–	–
iResNet [45]	–	1.4	43.34M	90
SceneFlowNet [38]	–	2.02	–	**60**
Ours	**4.79**	**1.24**	**2.7M**	112

Table 3. Comparisons on KITTI2012

Model	>2px		> 5 px		Mean Error		Time(s)
	Non-Occ	All	Non-Occ	All	Non-Occ	All	
PSMNet	**2.44**	**3.01**	**0.90**	**1.15**	**0.5**	0.6	1.3
GC-Net [6]	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 [24]	3.43	4.46	1.72	2.17	0.7	0.8	265
L-ResMatch [29]	3.64	5.06	1.50	2.26	0.7	1.0	48
MC-CNN [44]	3.90	5.45	1.64	2.39	0.7	0.9	67
iResNet-i2e2 [45]	2.69	3.34	1.06	1.32	0.5	0.6	**0.12**
Our model	2.70	3.23	1.07	1.27	**0.5**	**0.6**	0.18

the viewpoint of a driving car, made to resemble the KITTI datasets. It has 8830 training images which we use to analyze the effectiveness of region support, and the results are shown in Table.5. The average evaluation on SceneFlow is shown in Table.1.We compare with SceneFlowNet [38], GC-Net [6], iResNet [45] and MC-CNN [44], where we reach the best performance on 5*px* error rate and endpoint error(EPE) with the smallest model parameter. The evaluation of FlyingThings3D is shown in Table.2. Comparing to CRL [46], iResNet [45]and SceneFlowNet [38], where we reach the best performance on EPE and 3PE. The endpoint-error(EPE) is the average Euclidean distance between the prediction and ground-truth and the three-pixel-error(3PE) is the percentage of EPE value more than 3 [46].

The KITTI dataset is the real scene dataset on a driving car. The KITTI dataset contains 194 training and 195 testimage pair consist of images of challenging and varied road scene obtained from LIDAR data. We use the pre-trained model on Driving dataset and fine-tune on KITTI to obtain the final model. The comparation with GC-Net [6], PSMNet, SegStereo, iResNet-i2e2 [45], MC-CNN [44], Displetv v2 [24] and Kandao are shown in Table.3 and Table.3. From the evaluation, we can see the proposed method reaches the state-of-art performance both on KITTI2012 and KITTI2015.

5.3 Effectis Analysis

To evaluate the effectiveness of the proposed three algorithms of region support, we separately test them on the Driving dataset. The result is shown in Table.5.

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[6]	2.21	6.16	2.87	2.02	5.58	2.61	0.9
MC-CNN[5]	2.89	8.88	3.89	2.48	7.64	3.33	67
Displetv v2[24]	3.00	5.56	3.43	2.73	4.95	3.09	265
PSMNet	**1.86**	4.62	**2.32**	**1.71**	4.31	**2.14**	0.41
SegStereo	2.16	4.02	2.47	2.01	3.62	2.28	0.6
iResNet-i2e2[45]	2.10	3.64	2.36	1.94	**2.55**	2.15	**0.1**
Kandao	2.14	**3.45**	2.36	1.98	2.92	**2.14**	0.22
Our model	2.11	3.51	2.34	1.92	3.23	2.15	0.18

Table 5. Analysis on SceneFlow Driving dataset

RSN	Cost Computation	Cost Aggregation	Refinement	EEP	3PE	Time(ms)
×	√	√	√	7.32	11.56	48
√	√	×	×	14.37	3.74	97
√	×	√	×	13.70	6.72	103
√	×	×	√	16.32	2.96	105
√	√	√	√	9.77	8.61	112

We firstly use the region support obtained from ground-truth for the stereo matching, which reaches a significant performance both on speed and accuracy. Then we use RSN to provide the region support and test the effectiveness of the region support by separately applying the proposed algorithms. From the results, we can see that the application for cost computation can highly reduce the computational time. Applying for cost aggregation can lead to a low EPE which means a fine result in detail, while the application for refinement leads to a more smooth effects on disparity map.

6 Discussion

The region support significantly improves the stereo matching. We can see the reformulated pipeline gains a sufficient speed on computation, it should be noticed that the limitation of computational time comes from the region support network. During training, using the region support from ground-truth can lead an impressive speed with low computation requirement. The strategy to use region support is general even without the region support network. For example, the algorithm applied to compute ground-truth from disparity can also be employed to the sparse laser data, which we test on KITTI. Then the region support can be used as the fusion of laser data and stereo matching results. In the future, we will find a more effective region support network to speed up the determination of region support. Besides, we will find out the more general and effective usages of the region support not only for the stereo matching but also for other applications.

7 Conclusion

In this paper, we presented the region support for stereo matching. The region support was designed to handle particular challenges existing in different stereo matching steps. And we presented an effective solution to obtain the region support, which was built based on the commonness of these requirements to deal with the challenges. With the proposed region support, a novel stereo matching pipeline was reformulated in an effective an efficient manner. In addition, we proposed the region support network to generate the desired region support for stereo matching. The network was able to conduct the inference in disparity space and provided coarse-depth guidance for stereo matching. Furthermore, this network was also shown the ability for the geometry inference in disparity space.

References

1. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International journal of computer vision **47**(1-3) (2002) 7–42
2. Min, D., Lu, J., Do, M.N.: 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, IEEE (2011) 1567–1574
3. Hosni, A., Bleyer, M., Gelautz, M., Rhemann, C.: Local stereo matching using geodesic support weights. In: Image Processing (ICIP), 2009 16th IEEE International Conference on, IEEE (2009) 2093–2096
4. Zhang, K., Fang, Y., Min, D., Sun, L., Yang, S., Yan, S., Tian, Q.: Cross-scale cost aggregation for stereo matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2014) 1590–1597
5. Zbontar, J., LeCun, Y.: Computing the stereo matching cost with a convolutional neural network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 1592–1599
6. Kendall, A., Martirosyan, H., Dasgupta, S., Henry, P., Kennedy, R., Bachrach, A., Bry, A.: End-to-end learning of geometry and context for deep stereo regression. (2017)
7. Tombari, F., Mattoccia, S., Di Stefano, L., Addimanda, E.: Classification and evaluation of cost aggregation methods for stereo correspondence. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE (2008) 1–8
8. Ma, Z., He, K., Wei, Y., Sun, J., Wu, E.: Constant time weighted median filtering for stereo matching and beyond. In: Proceedings of the IEEE International Conference on Computer Vision. (2013) 49–56
9. Tombari, F., Mattoccia, S., Di Stefano, L.: Segmentation-based adaptive support for accurate stereo correspondence. Advances in Image and Video Technology (2007) 427–438
10. Mei, X., Sun, X., Dong, W., Wang, H., Zhang, X.: Segment-tree based cost aggregation for stereo matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2013) 313–320
11. Adhyapak, S., Kehtarnavaz, N., Nadin, M.: Stereo matching via selective multiple windows. Journal of Electronic Imaging **16**(1) (2007) 013012
12. Kim, J.C., Lee, K.M., Choi, B.T., Lee, S.U.: A dense stereo matching using two-pass dynamic programming with generalized ground control points. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. Volume 2., IEEE (2005) 1075–1082
13. Gong, M., Yang, R.: Image-gradient-guided real-time stereo on graphics hardware. In: 3-D Digital Imaging and Modeling, 2005. 3DIM 2005. Fifth International Conference on, IEEE (2005) 548–555
14. Medioni, G., Nevatia, R.: Segment-based stereo matching. Computer Vision, Graphics, and Image Processing **31**(1) (1985) 2–18
15. Zhang, K., Lu, J., Lafruit, G.: Cross-based local stereo matching using orthogonal integral images. IEEE transactions on circuits and systems for video technology **19**(7) (2009) 1073–1079
16. Darrell, T.: A radial cumulative similarity transform for robust image correspondence. In: Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on, IEEE (1998) 656–662

17. Wang, L., Liao, M., Gong, M., Yang, R., Nister, D.: High-quality real-time stereo using adaptive cost aggregation and dynamic programming. In: 3D Data Processing, Visualization, and Transmission, Third International Symposium on, IEEE (2006) 798–805
18. Xu, Y., Wang, D., Feng, T., Shum, H.Y.: Stereo computation using radial adaptive windows. In: Pattern Recognition, 2002. Proceedings. 16th International Conference on. Volume 3., IEEE (2002) 595–598
19. Yoon, K.J., Kweon, I.S.: Adaptive support-weight approach for correspondence search. IEEE Transactions on Pattern Analysis and Machine Intelligence **28**(4) (2006) 650–656
20. Wei, Y., Quan, L.: Region-based progressive stereo matching. In: Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on. Volume 1., IEEE (2004) I–I
21. Luo, W., Schwing, A.G., Urtasun, R.: Efficient deep learning for stereo matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 5695–5703
22. Bleyer, M., Rhemann, C., Rother, C.: Patchmatch stereo-stereo matching with slanted support windows. In: Bmvc. Volume 11. (2011) 1–11
23. Sinha, S.N., Scharstein, D., Szeliski, R.: Efficient high-resolution stereo matching using local plane sweeps. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2014) 1582–1589
24. Guney, F., Geiger, A.: Displets: Resolving stereo ambiguities using object knowledge. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 4165–4175
25. Hirschmuller, H.: Stereo processing by semiglobal matching and mutual information. IEEE Transactions on pattern analysis and machine intelligence **30**(2) (2008) 328–341
26. Slossberg, R., Wetzler, A., Kimmel, R.: Deep stereo matching with dense crf priors. arXiv preprint arXiv:1612.01725 (2016)
27. Knöbelreiter, P., Reinbacher, C., Shekhovtsov, A., Pock, T.: End-to-end training of hybrid cnn-crf models for stereo. arXiv preprint arXiv:1611.10229 (2016)
28. Yamaguchi, K., McAllester, D., Urtasun, R.: Efficient joint segmentation, occlusion labeling, stereo and flow estimation. In: European Conference on Computer Vision, Springer (2014) 756–771
29. Shaked, A., Wolf, L.: Improved stereo matching with constant highway networks and reflective confidence learning. arXiv preprint arXiv:1701.00165 (2016)
30. Yu, L., Wang, Y., Wu, Y., Jia, Y.: Deep stereo matching with explicit cost aggregation sub-architecture. arXiv preprint arXiv:1801.04065 (2018)
31. Yang, Q.: A non-local cost aggregation method for stereo matching. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE (2012) 1402–1409
32. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 770–778
33. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. arXiv preprint arXiv:1606.00915 (2016)
34. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122 (2015)

35. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). (2017) 2881–2890
36. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE (2012) 3354–3361
37. Menze, M., Geiger, A.: Object scene flow for autonomous vehicles. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 3061–3070
38. Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T.: A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 4040–4048
39. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch. (2017)
40. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
41. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. International journal of computer vision **59**(2) (2004) 167–181
42. Gao, W., Zhang, X., Yang, L., Liu, H.: An improved sobel edge detection. In: Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on. Volume 5., IEEE (2010) 67–71
43. Canny, J.: A computational approach to edge detection. In: Readings in Computer Vision. Elsevier (1987) 184–203
44. Zbontar, J., LeCun, Y.: Stereo matching by training a convolutional neural network to compare image patches. Journal of Machine Learning Research **17**(1-32) (2016) 2
45. Liang, Z., Feng, Y., Guo, Y., Liu, H., Qiao, L., Chen, W., Zhou, L., Zhang, J.: Learning deep correspondence through prior and posterior feature constancy. arXiv preprint arXiv:1712.01039 (2017)
46. Pang, J., Sun, W., Ren, J., Yang, C., Yan, Q.: 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)