WaveletStereo: Learning Wavelet Coefficients of Disparity Map in Stereo Matching

Menglong Yang, Fangrui Wu and Wei Li
School of Aeronautics and Astronautics, Sichuan University
Chengdu, Sichuan, PR China
mlyang@scu.edu.cn, wufangrui@stu.scu.edu.cn, li.wei@scu.edu.cn

Abstract

Some stereo matching algorithms based on deep learning have been proposed and achieved state-of-the-art performances since some public large-scale datasets were put online. However, the disparity in smooth regions and detailed regions is still difficult to accurately estimate simultaneously. This paper proposes a novel stereo matching method called WaveletStereo, which learns the wavelet coefficients of the disparity rather than the disparity itself. The WaveletStereo consists of several sub-modules, where the low-frequency sub-module generates the low-frequency wavelet coefficients, which aims at learning global context information and well handling the low-frequency regions such as textureless surfaces, and the others focus on the details. In addition, a densely connected atrous spatial pyramid block is introduced for better learning the multi-scale image features. Experimental results show the effectiveness of the proposed method, which achieves state-of-the-art performance on the large-scale test dataset Scene Flow.

1. Introduction

As a convenient and cheap means of obtaining object depth, stereo vision acts a more and more pivotal part in computer vision, with the increasing applications of virtual (augmented) reality, 3D object detection and recognition, motion sense game and unmanned aerial vehicle, etc. Scholars have paid many attentions to stereo matching, which is a key step for stereo vision. Recent years, the research on stereo matching has got satisfying achievement since some public datasets were published online, such as Middlebury [39] and KITTI stereo benchmark [10, 12], which is convenient for researchers to compare their algorithms against state-of-the-art algorithms. However, stereo matching for the complex environment including amounts of ill-posed regions, such as textureless or detailed regions, is still a challenging topic[32].

As Scharstein and Szeliski [39] summarized, there were usually four steps in a typical traditional stereo matching algorithm, i.e., matching cost computation, cost aggregation, optimization, and disparity refinement, respectively. Traditional stereo matching algorithms can be mainly split into two categories, i.e., the local methods and the global (semi-global) methods.

Most local stereo matching methods were more interested at studying the first two steps [16, 53], and often accurately estimated the disparity in regions with high-frequency details, but frequently failed in low-frequency areas such as textureless and saturated regions.

To improve the performance in low-frequency region, many global (semi-global) methods, such as graph cuts [24, 36, 33], belief propagation (BP) [23, 52, 51, 60, 49] and Semi-global matching (SGM) algorithm [15], focused more on the research of latter two steps. A Conditional Random Field (CRF) model was often constructed in global (semi-global) algorithms, which contained the assumptions of photo-consistency and smoothness. The photo-consistency expects that the matching pixels have similar appearance features, and the smoothness constrains the divergence of neighboring pixels' label except some disparity-jump places, which measures the cost of assigning labels to neighboring pixels, such as the pairwise smoothness [23, 52, 51, 60] or the higher-order smoothness [49, 47, 25]. Compared with local methods, these methods improved the performance in the low-frequency regions, but solving the CRF model is often time consuming and they may still falsely predict the disparity for pixels in high-frequency regions.

Recent works based on deep learning use convolutional neural networks to learn similarity computation and contextual information, and vastly improve the accuracy and robustness of disparity estimation [55, 20]. However, there are a lot of mismatches in ill-posed regions, especially the high-frequency regions, such as thin surfaces, occlusion areas and repeated patterns, although these regions account for only a small proportion and the false predictions in them

have minor impact on the total evaluation of accuracy. Intuitively, the disparity estimation in low-frequency regions is more dependent on the global contextual information, but the estimation in high-frequency regions depends more on the image details. Therefore, it is difficult to train a network to simultaneously predict these regions accurately. In order to resolve this problem, this paper proposes a novel stereo matching algorithm based on learning wavelet coefficients of the disparity map, with main contributions summarized as follows.

Firstly, we proposed an end-to-end architecture for stereo matching, called WaveletStereo, which incorporates a mechanism of multi-resolution wavelet reconstruction. WaveletStereo contains several predictors of wavelet coefficients, which estimate the wavelet coefficients with different resolutions of the disparity map and calculate the disparity map through multi-resolution wavelet reconstruction, where the low-frequency wavelet predictors focus on learning the global contextual information of the disparity map, while the high-frequency wavelet predictors concentrate on generating the details of the disparity map.

Secondly, we proposed a densely connected atrous spatial pyramid block, which effectively captures multi-scale contextual information with relatively few parameters. A feature representation with a very deep and complex structure can benefit the final performance for the stereo algorithms, but it can also increase the computational time and the complexity of training. Based on the proposed block, this work obtained good accuracy without the computational burden.

Finally, we adopted the proposed WaveletStereo algorithm to achieve the state-of-the-art performance on the large-scale stereo benchmark Scene Flow.

2. Related work

Recent years, the public stereo datasets [9, 38, 31] have spawned many learning-based stereo algorithms. Early learning-based stereo algorithms put the learning mechanisms into the traditional stereo matching framework. For example, some works[42, 14] trained a model to automatically estimate the confidence of the computed matching cost. As a pioneer of using convolutional neural network (CNN) in stereo, J. Zbontar and Y. LeCun [55] used a CNN to learn similarity computation for a pair of image patches to improve the robustness to image noise and illumination variation, and refined the disparity using a traditional Semi-global matching algorithm [15]. It outperformed on KITTI benchmark, although it frequently suffered from mismatch in ill-posed regions. The extensive works [30, 48] dramatically speed-up the calculation of matching cost by using cosine similarity or Euclidean distance of the embedding features to compute the matching cost, instead of the forward propagation of several fully connected layers. Besides learning matching cost, some methods [2, 50] added a model to train smoothness cost, in order to reduce the roughness and over-smoothness when using SGM algorithm to refine the disparity. In addition, some other methods addressed refining matching cost [26, 27] or learning the parameters of optimization model [56, 37, 38].

Many end-to-end learning methods have successively achieved the state-of-the-art performances since the emergence of Scene Flow [31], which is a large-scale synthetic stereo dataset. Kendall et al. [20] used deep feature representations to form a cost volume and adopted cost filtering by a series of 3D convolutions to learn contextual information. It achieved sub-pixel accuracy with a differentiable soft argmin operation. Since then, many similar approaches have been proposed. Yu et al. [54] used a mechanism of multiple cost aggregation proposals to refine the cost volume. The pyramid stereo matching network (PSMNet) [5] was proposed, where a spatial pyramid pooling module aggregates context in different scales and locations to form a cost volume, and a stacked multiple hourglass network learns to regularize cost volume. Guo et al. [13] presented a method called group-wise correlation to improve the representations for measuring feature similarities. The idea of left-right consistency check was adopted in LRCR [18] and [4] to refine the disparity estimation. Song et al. [41] improved the details in disparity maps by utilizing a multi-task learning in conjunction with edge detection task. A two-stages cascade CNN architecture was adopted in [35], in which the first stage added the up-convolution modules into DispNet [31] to improve the details in disparities maps and the second stage learned the multi-scale residuals to further refine the disparity. Similarly, Khamis et al. [21] employed a learned edge-aware upsampling function to refine the disparity predicted from a very low resolution cost volume. The low resolution cost volume gives it speed advantage and the upsampling operations refine details, but it is unlikely to recover the details if the detailed signals are completely missing from the low resolution cost volume. In addition, some other works focused on self-supervised methods to learn from the open-world unlabeled data [57, 59].

This work solves the stereo by learning the wavelet coefficients of disparity map, rather than directly learning the disparity. Spectral approaches have been widely applied in image processing tasks [29, 43, 3, 44] for many years. Recently, there are some works exploring the possible advantages of learning the filters by deep learning for image analysis in the wavelet domain [7] or incorporating a spectral approach into CNNs [46]. Especially, Huang et al. proved the feasibility of solving the over-smoothed problem and improving the textural details in the image super-resolution application, by learning the wavelet coefficients of the high-resolution image. To the best of our knowledge we are the first to study the wavelet learning algorithm for stereo

matching.

3. WaveletStereo

We describe the proposed algorithm that predicts disparity for a rectified pair of stereo images in this section. The architecture can be mainly divided into three modules including deep representation, multi-resolution cost volumes and multi-resolution reconstruction, as shown in Fig. 1. The following sections respectively describe these main modules, the detailed network configuration is elaborated in the supplementary material.

3.1. Deep representation

The deep representation aims at learning to encode local and global contextual information, which extracts unary features from a pair of stereo images with a shared weight Siamese network, to form a cost volume. We bring the resolution to a fourth by using two downsampling modules, each of which is followed by a densely connected atrous spatial pyramid block in order to encode the contextual information more efficiently. The downsampling is simply performed by using convolutions with $3 \times 3 \times 32$ filters and a stride of 2. The last layer of the deep representation is a convolution operation with $3 \times 3 \times 32$ filters, and the outputs are $\frac{1}{4}H \times \frac{1}{4}W \times 32$ features for the left and right image, respectively. Each convolution is followed by a batch normalization and ReLU activation except the output layer.

Inspired by DenseNet [6, 17], we propose the densely connected atrous spatial pyramid (DCASPP) block, the structure of which is shown in Fig. 2. To learn different scales of contextual information, we use multiple atrous convolutions with different dilated rates and concatenate their results to group an inception layer, similar to the operation of atrous spatial pyramid pooling (ASPP) in [6]. In addition, we adopt the densely connected structure to make use of the advantages presented in [17], e.g. encouraging feature reuse and substantially reducing the number of parameters, and further expanding the scale of learnt contextual information without exaggeratively extending the dilated rate.

At the first downsampling step, we use two inception layers, each of which includes four atrous convolutions with $3 \times 3 \times 4$ filters and dilated rates of 1, 2, 4 and 8, respectively. For the second downsampling step, we use four inception layers, each of which includes two atrous convolutions with $3 \times 3 \times 8$ filters and dilated rates of 1 and 2, respectively.

3.2. Multi-resolution cost volumes

The module of multi-resolution cost volumes is relatively simple, which forms several cost volumes with different resolutions from the unary features, and lays the foundation for multi-resolution wavelet reconstruction.

In this work, we first construct a cost volume with a fourth resolution by concatenating the left and right unary features with shifted disparities, similar to [20], which is followed by two consecutive downsampling operations. We finally obtain three volumes with the resolutions of fourth, eighth and sixteenth, respectively. The downsampling is simply performed using 3D convolution with $3 \times 3 \times 3 \times 32$ filters and stride of 2, which has certain effects of cost filtering as well.

3.3. Multi-resolution wavelet reconstruction

The third module learns the regressions from multi-resolution cost volumes to the wavelet coefficients with different resolutions, performs the wavelet reconstruction level by level through inverse wavelet transforms and finally acquires a disparity map. The multi-resolution reconstruction iteratively repeats a similar process, i.e., mapping a cost volume to the wavelet coefficients, which is finished by a CNN composed of cost filtering and wavelet regression, as shown in Fig. 3. Note it does not compute the low-frequency wavelet approximation (the yellow regions) except for the lowest resolution. The low-frequency wavelet approximation incorporates wider context, while the high-frequency wavelet coefficients describe details for disparity map.

In this work, we use CNNs to learn the Haar wavelet coefficients of the disparity map, which are sufficient to depict scene information of different frequencies. There are four wavelet coefficients with the same resolution at each level, i.e., the low-frequency approximation, the horizontal high frequency wavelet, the vertical high frequency coefficient and the diagonal high frequency information, respectively. The four wavelet coefficients are used to reconstruct the low-frequency approximation of the higher resolution by inverse wavelet transform, which is iteratively repeated until the disparity map with the full resolution is reconstructed. More specifically, a CNN is used to map the cost with the lowest resolution (1/16) into the low-frequency wavelet approximation and corresponding high-frequency coefficients with the resolution of 1/8 (it includes an upsampling operation in the CNN). Then the low-frequency wavelet approximation and high-frequency coefficients are used together to get wavelet approximation with the higher resolution (1/4) by inverse wavelet transform. Similar steps are iteratively performed level by level until getting the disparity map with the full resolution.

Here the cost filtering includes a series of 3D convolutions and transposed convolutions, the network configuration of which can be seen in the supplementary material in detail. An important issue is described below, which is how wavelet coefficients of the disparity map are regressed from the filtered cost volumes.

Recently, a regression operation proposed in [20], i.e., soft argmin, is widely used in disparity estimation after cost

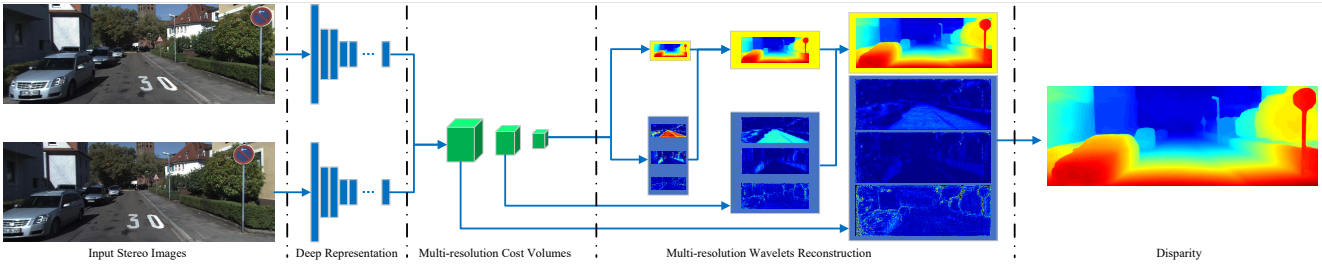

Figure 1. WaveletStereo Network architecture pipeline.

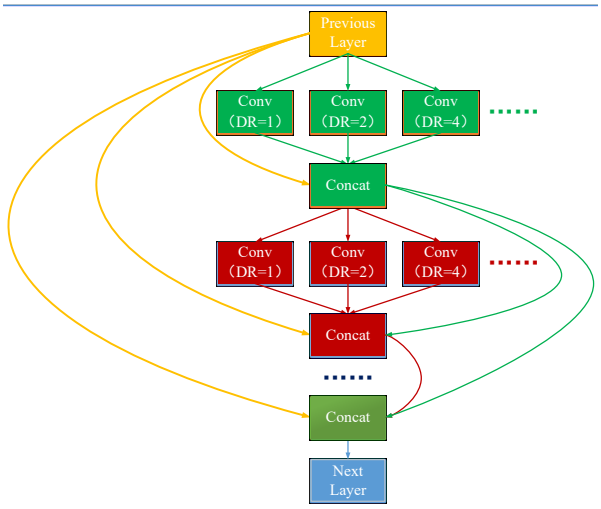

Figure 2. The structure of densely connected atrous spatial pyramid module, where "DR" denotes the dilated rate.

filtering. For each pixel i, the disparity d_i is regressed as a weighted softmax function:

$$d_i = \sum_{d=0}^{D_{max}} d \times \frac{e^{-y_i(d)}}{\sum_{d'=0}^{D_{max}} e^{-y_i(d')}} \tag{1}$$

where y_i is the filtered cost at pixel i and D_{max} is the predefined maximum disparity. The softmax operation to the filtered cost at pixel i, i.e.,

$$p_i(d) = \frac{e^{-y_i(d)}}{\sum_{d'=0}^{D_{max}} e^{-y_i(d')}} \tag{2}$$

can be regarded as the probability of $d_i = d$. The formulation (1) can be abbreviated as

$$D = \sum_{d=0}^{D_{max}} d \times P(d) \tag{3}$$

where D is the estimated disparity map, and $P(d)$ is the softmax operation to the filtered cost. Making a wavelet transform to the disparity map, it has

$$f(D) = f\left(\sum_{d=0}^{D_{max}} d \times P(d)\right) = \sum_{d=0}^{D_{max}} d \times f\left(P(d)\right) \tag{4}$$

where $f(\cdot)$ is the wavelet transform. If the wavelet coefficient of the disparity map is denoted as ψ, it can be calculated as

$$\psi = \sum_{d=0}^{D_{max}} d \times \psi(d) \tag{5}$$

The disparity map D can be reconstructed by inverse wavelet transform with the wavelet coefficients, the estimation of the disparity D can be thus transformed into the prediction of the wavelet coefficients. In fact, $\psi(d)$ in (5) can be regarded as the wavelet coefficients of $P(d)$. This wavelet transform can be iteratively performed, i.e., multi-resolution wavelet decomposition, which is shown as Figure 4.

If the maximum value of the disparity map is D_{max}, the value range of its low-frequency approximation at first level decomposition is $[0, 2D_{max}]$, and the value range of its high-frequency coefficients at first level decomposition is $[-D_{max}, D_{max}]$, after Haar decomposition. And so on, the value ranges of low-frequency and high-frequency wavelet coefficients at l-th level decomposition are $[0, 2^l D_{max}]$ and $[-2^{l-1}D_{max}, 2^{l-1}D_{max}]$, respectively. Therefore, the wavelet coefficients can be calculated as weighted softmax functions, i.e., the low-frequency coefficent at pixel i of level l is

$$\bar{\psi}_i^l = \sum_{d=0}^{D_{max}} 2^l d \times \frac{e^{-y_i(d)}}{\sum_{d'=0}^{D_{max}} e^{-y_i(d')}}, \tag{6}$$

and the high-frequency coefficients at pixel i of level l are

$$\tilde{\psi}_i^l = \sum_{d=0}^{D_{max}} 2^{l-1} d \times \left(\frac{e^{-\epsilon_i(d)}}{\sum_{d'=0}^{D_{max}} e^{-\epsilon_i(d')}} - \frac{e^{-\eta_i(d)}}{\sum_{d'=0}^{D_{max}} e^{-\eta_i(d')}} \right), \tag{7}$$

where y_i, ϵ_i and η_i are the filtered costs at pixel i. Note we use two variables ϵ_i and η_i, the softmax values of which are both $[0, 2^{l-1}D_{max}]$, to ensure the right value range of the high-frequency coefficients.

3.4. Loss

We train our model with supervised learning using groundtruth disparity data, where the loss function contains two terms.

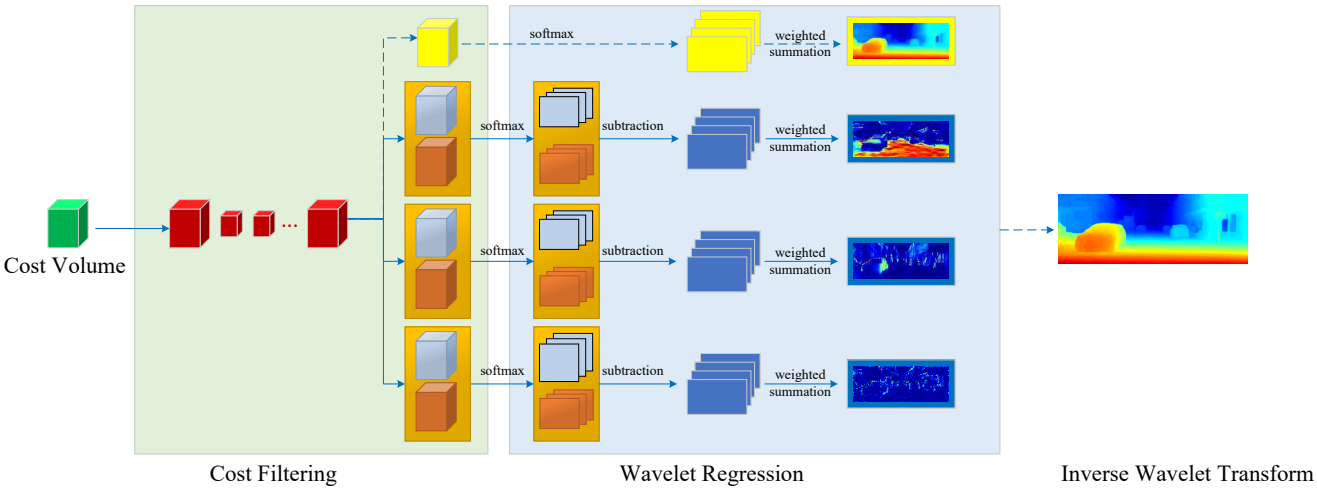

Figure 3. The pipeline from a cost volume to the wavelet coefficients.

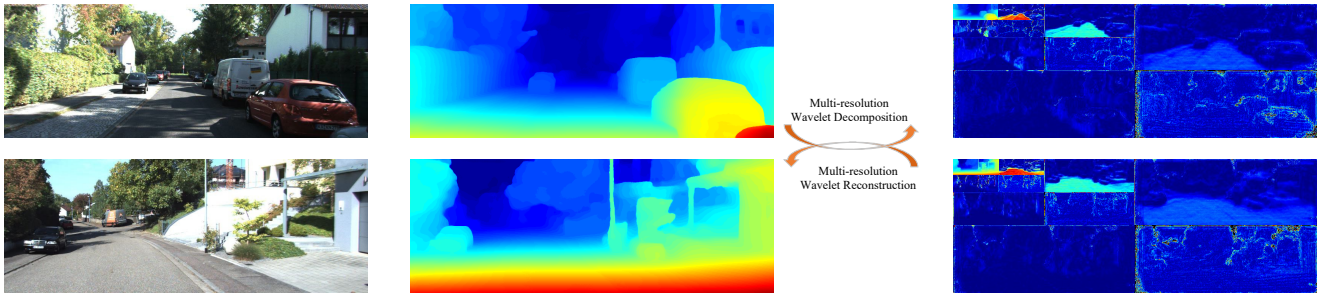

Figure 4. The prediction of the wavelet coefficients of the disparity map is equivalent to the estimation of the disparity itself, in terms of wavelet transform. Left column is the reference images, the middle column is the final estimated disparities, and the last column is the predicted wavelet coefficients.

The first term aims at training the predictors of the wavelet coefficients. Similar to [41], we use the smooth L_1 loss between the predicted wavelet coefficients ψ_i^l (including low-frequency and high-frequency coefficients) and the ground truth wavelet coefficients $\hat{\psi}_i^l$ for each labeled pixel i, for its low sensitivity to outliers, which is defined as

$$\mathcal{L}_1 = \frac{1}{N}\sum_{l=1}^{L}\sum_{i=1}^{N} smooth_{L_1}\left(\psi_i^l - \hat{\psi}_i^l\right), \tag{8}$$

in which

$$smooth_{L_1}(\epsilon) = \begin{cases} 0.5\epsilon^2, & \text{if } |\epsilon| < 1 \\ |\epsilon| - 0.5, & \text{otherwise} \end{cases}, \tag{9}$$

where N is the number of the pixels. The groundtruth wavelet coefficients are obtained from decomposing the groudtruth disparity map by 2-D fast wavelet transform [34, 28].

We adopt second term to supervise the final disparity map. The loss is defined as

$$\mathcal{L}_2 = \frac{1}{N}\sum_{i=1}^{N} smooth_{L_1}\left(\hat{d}_i - d_i\right) \tag{10}$$

where d_i and $\hat{d}_i$ are the predicted disparity and the groundtruth disparity value of the pixel i, respectively.

Finally, we train the model using an end-to-end supervised learning mechanism with following loss function.

$$\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2 \tag{11}$$

4. Experiment

Experimental setup and results are presented in this section. We not only evaluate the performances of the proposed method on public stereo benchmarks and compare them with some state-of-the-art stereo methods, but also analyze the effectiveness of each proposed module by ablation studies.

4.1. Implementation details

Datasets. We evaluate our method on the Scene Flow [12], KITTI 2012 [10] and KITTI 2015 [32] datasets in this work.

(i) **Scene Flow** [31] is a large synthetic dataset containing $35,454$ stereo pairs for training and $4,370$ for testing. SceneFlow has a large scale training data, which

is beneficial to the training of deep-learning-based method. In addition, the groundtruth of Scene Flow has no measurement error unlike that of other realistic datasets, because it is completely synthetic. Combined with large scale test data, it thus facilitates testing the accuracy of the algorithms more thoroughly and precisely.

(ii) **KITTI** is a real-world dataset with dynamic street views from the perspective of a driving car, in which the groundtruth depth maps for training and evaluation are sparsely obtained from LIDAR data. It includes KITTI 2012 and KITTI 2015, where KITTI 2012 provides 194 stereo pairs for training and 195 for evaluation through its online leaderboard, and KITTI 2015 provides 200 pairs for training and 200 pairs for evaluation.

Training. We adopted TensorFlow [1] to train the convolutional neural networks with the stochastic optimization algorithm of Adam [22], where $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. For Scene Flow dataset, we used an initial learning rate of 0.001 which was kept constant for the first 10 epochs and then set to be 0.0001 until the end (approximately 25 epochs). For KITTI dataset, we fine-tuned the model pre-trained on Scene Flow for a further 200 epochs with a constant learning rate of 0.0001. The parameters of the networks were initialized from random, and trained on a Nvidia GeForce Titan RTX GPU with a batch size of 2. The pixel intensities of each image are normalized into $[-1, 1]$. The maximum disparity was set as $D_{max} = 192$.

4.2. Ablation studies

In this section, we conduct several experiments on the Scene Flow dataset to compare some different model variants and justify the effectiveness of our design choices, where we test our networks with two widely used metrics for evaluation:

(i) Endpoint error (EPE): the average Euclidean distance between the pixels of estimated disparity and the groundtruth.

(ii) Three-pixel error (> 3 pixel): the percentage of pixels with endpoint error more than 3.

We first study the effectiveness of the deep representation module by comparing the proposed densely connected atrous spatial pyramid (DCASPP) block with Vortex Pooling[4] and atrous spatial pyramid pooling (ASPP) [6]. The experimental results are shown in first half of Table 1, where we can see the effectiveness of DCASPP. The difference of calculation time between ASPP and DCASPP is little, but the performance of DCASPP is significantly better than ASPP.

Then we study the effectiveness of the wavelet learning by several ablation experiments. First, we use CNNs to learn only the low-frequency coefficients, which is basically equivalent to multi-scale upsampling refinement mechanism similar to StereoNet [21]. Next, we adopt CNNs to learn the high-frequency coefficients of 1st level, 2nd level and 3rd level step by step. The experimental results are shown in second half of Table 1, where we can see the effectiveness of wavelet learning. Here we denote the low-frequency as 'LF' and the high-frequency as 'HF'.

Table 1. Results on the Scene Flow dataset. We compare different architecture variants to justify the effectiveness of our design choices.

Model	> 3 px (%)	EPE	Time
Vortex Pooling[4]	5.68	1.20	0.41s
ASPP[6]	5.48	1.07	0.26s
DCASPP	4.13	0.84	0.27s
Low-frequency Only	13.54	1.855	0.1s
LF + HF of Level 3	4.47	0.89	0.13s
LF + HF of Level 2 + 3	4.42	0.856	0.18s
Full Model	4.13	0.84	0.27s

Figure 5 shows the predicted disparity map of some examples on Scene Flow test data. The low-frequency regions, such as the textureless surfaces, are accurately estimated due to the accurate prediction of the low-frequency coefficients. Some sharp regions, like the thin surfaces, can be also correctly estimated, which is credited to the high-frequency predictors.

4.3. Comparisons with the state-of-the-art methods

In this section, we compare the proposed algorithm with the state-of-the-art methods. Firstly we compare the our method with the state-of-the-art algorithms on SceneFlow dataset, including PSMNet [5], DispFulNet [35], CRL [35], GC-Net [20], DRR [11], Edge Stereo [41], StereoNet [21] , DeepPruner [8] and Stereo-DRNet [4], as presented in Table 2. Our approach outperforms previous methods in terms of two evaluation metrics. The methods whose results are closest to ours are Stereo-DRNet [4] and DeepPruner [8]. Stereo-DRNet [4] focused on the refinement of the disparity. It predicts the disparities of left and right views simultaneously, and utilizes left-right image consistence and disparity consistence to further refine the disparity. However, our method outperforms this architecture without such a mechanism of left-right consistency check. DeepPruner [8] adopted a differentiable module to discard most disparities without requiring full cost volume evaluation, in order to speed up the estimation of the disparity. It achieved a comparable result with Stereo-DRNet [4] on Scene Flow, which was slightly weaker than the proposed algorithm.

Then we evaluate our model on KITTI. Table 3 and table 4 compare the error rates of some published state-of-the-art

Table 2. Comparisons of stereo matching algorithms on the Scene Flow test set.

metric	PSMNet [5]	DispFulNet [31]	CRL [35]	GC-Net [20]	DRR [11]
> 3 pixel (%)	-	8.61	6.20	7.20	7.21
EPE	1.09	1.75	1.32	-	-
metric	EdgeStereo [41]	StereoNet [21]	Stereo-DRNet [4]	DeepPruner [8]	Ours
> 3 pixel (%)	4.99	-	-	-	**4.13**
EPE	1.12	1.10	0.86	0.86	**0.84**

Table 3. KITTI 2012 test set results [10]. This benchmark contains 194 train and 195 test image pairs.

Method	Out-Noc	Out-All	Avg-Noc	Avg-All	Runtime
PSMNet [5]	1.49 %	1.89 %	0.5 px	0.6 px	0.41 s
Stereo-DRNet [4]	1.42 %	1.83 %	0.5 px	0.5 px	0.23 s
EdgeStereo [41]	1.73 %	2.18 %	0.5 px	0.6 px	0.48 s
GC-NET [20]	1.77 %	2.30 %	0.6 px	0.7 px	0.9 s
PDSNet [45]	1.92 %	2.53 %	0.9 px	1.0 px	0.5 s
SGM-Net [2]	2.29 %	3.50 %	0.7 px	0.9 px	67 s
SsSMnet [58]	2.30 %	3.00 %	0.7 px	0.8 px	0.8 s
PBCP [40]	2.36 %	3.45 %	0.7 px	0.9 px	68 s
Displets v2 [12]	2.37 %	3.09 %	0.7 px	0.8 px	265 s
Ours	1.66 %	2.18 %	0.5 px	0.6 px	0.27 s

Table 4. KITTI 2015 test set results [32]. This benchmark contains 200 training and 200 test color image pairs. The qualifier 'bg' refers to background pixels which contain static elements, 'fg' refers to dynamic object pixels, while 'all' is all pixels (fg+bg). The results show the percentage of pixels which have error greater than three pixels or 5% disparity error from all 200 test images.

Method	D1-bg	D1-fg	D1-all	Time
PSMNet [5]	1.86 %	4.62 %	2.32 %	0.41 s
Stereo-DRNet [4]	1.72 %	4.95 %	2.26 %	0.23 s
EdgeStereo [41]	2.27 %	4.18 %	2.59 %	0.27 s
CRL [35]	2.48 %	3.59 %	2.67 %	0.47 s
GC-NET [20]	2.21 %	6.16 %	2.87 %	0.9 s
LRCR [19]	2.55 %	5.42 %	3.03 %	49.2 s
DRR [11]	2.58 %	6.04 %	3.16 %	0.4 s
SsSMnet [58]	2.70 %	6.92 %	3.40 %	0.8 s
Displets v2 [40]	3.00 %	5.56 %	3.43 %	265 s
PBCP [40]	2.58 %	8.74 %	3.61 %	68 s
SGM-Net [2]	2.66 %	8.64 %	3.66 %	67 s
Ours	2.12 %	5.34 %	2.65 %	0.27 s

algorithms comparable to the proposed method on KITTI 2012 and 2015 datasets, respectively. Visual results are not shown here for conciseness. We recommend the readers to go to the KITTI website [10] for more details.

The proposed method achieves an error rate of 2.18% in KITTI 2012 and 2.65% in KITTI 2015 (three-pixel error), which is comparable to EdgeStereo [41] but inferior than Stereo-DRNet [4] and PSMNet [5]. However WaveletStereo significantly outperforms these methods on Scene Flow test set. We think this difference is mainly attributed to the fact that groundtruth of KITTI training set is sparsely labeled, and it is difficult for which to compute the high-frequency wavelet coefficients in too many regions. The training of the high-frequency wavelet coefficient prediction is required to large scale training data with amount of details, and the training of the high-frequency predictors is thus not comprehensive. In other words, the loss function (8) almost does not work on KITTI. In addition, the sparsely labeled data of KITTI, especially the data often unlabeled in the high-frequency regions, may also affect the evaluation rank of the proposed method. Scene Flow dataset contains more than 30 thousands training images, hence the well-learned WaveletStereo achieves the state-of-the-art performance on this dataset.

5. Conclusion

This paper proposes a novel end-to-end deep learning architecture with a multi-resolution wavelet reconstruction mechanism for stereo vision, which consists of multi-scale predictors of wavelet coefficients. The low-frequency predictors exploit global context information and well handles the low-frequency regions such as textureless surfaces, and the high-frequency predictors generate details. In addition, a module of densely connected atrous spatial pyramid is proposed and used in the deep representation for better learning different scales of contextual information. Experimental results demonstrates the efficacy of the proposed method.

This work is an attempt to combine the traditional image processing technique with deep learning, for stereo matching. The understanding of the wavelet will contribute to applying the stereo algorithm more flexible. In some applications, the high-frequency predictors can be removed to obtain better timeliness, if the details are not required.

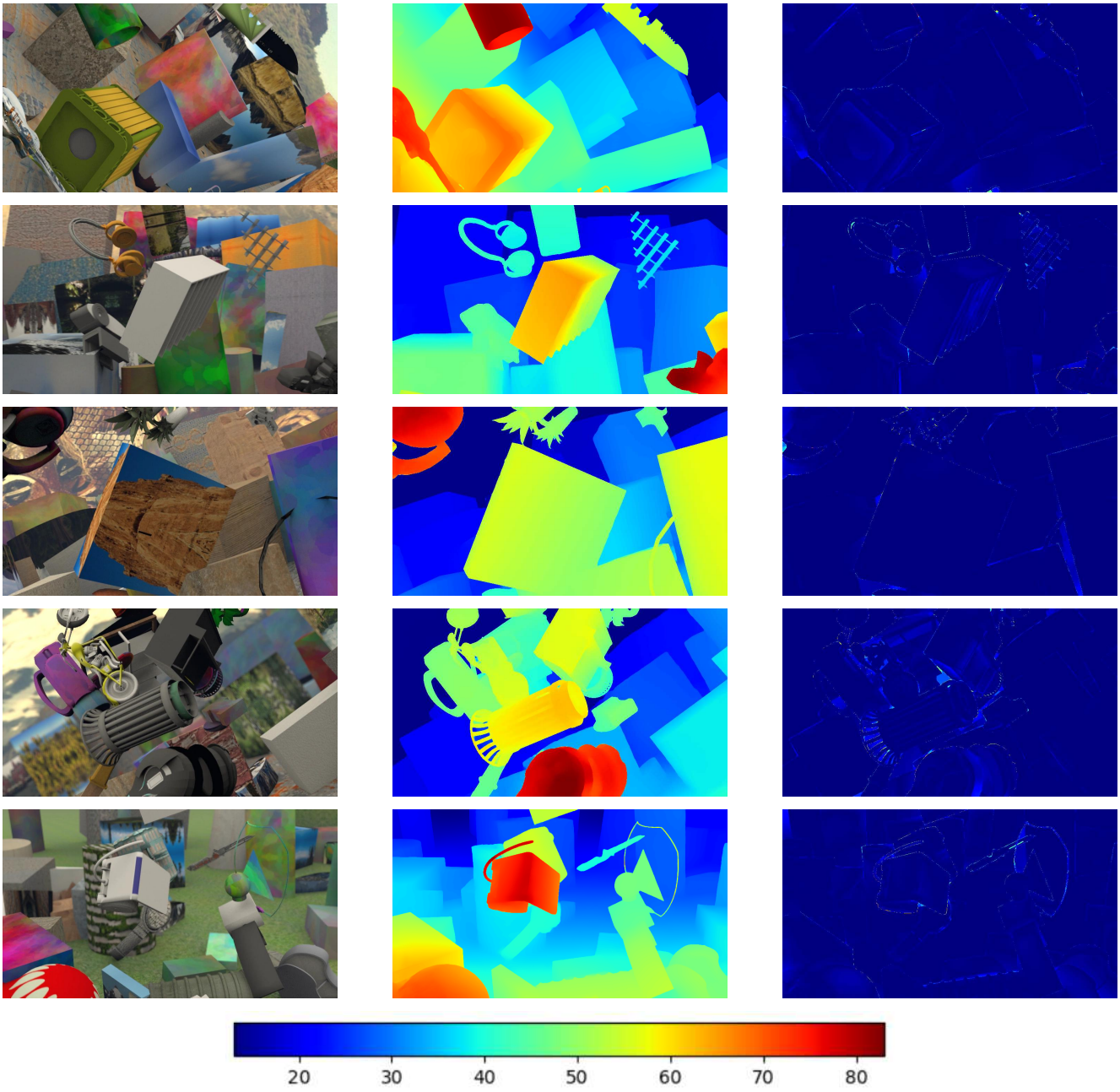

Figure 5. Qualitative results of Scene Flow test set. From left to right: left stereo input image, predicted disparity and the errors. The last row is the color bar for error maps.

We can make a reasonable trade-off between accuracy and speed according to application requirements.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (Grant No. U1933134, U19A2071 and 61860206007), Sichuan Science and Technology Program (Grant No. 18YYJC1287) and the funding from Sichuan University (Grant No. 2018SCUH0042).

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv: Distributed, Parallel, and Cluster Computing*, 2015.

[2] S. Akihito and P. Marc. Sgm-nets: Semi-global matching with neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 231–240, June 2017.

[3] E. J. Candes. The curvelet transform for image denoising. 1:7, 2001.

[4] R. Chabra, J. Straub, C. Sweeney, R. A. Newcombe, and H. Fuchs. Stereodrnet: Dilated residual stereo net. *arXiv: Computer Vision and Pattern Recognition*, 2019.

[5] 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.

[6] L. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv: Computer Vision and Pattern Recognition*, 2017.

[7] F. Cotter and N. G. Kingsbury. Deep learning in the wavelet domain. *arXiv: Computer Vision and Pattern Recognition*, 2018.

[8] S. Duggal, S. Wang, R. H. Wei-Chiu Ma, and R. Urtasun. Deeppruner: Learning efficient stereo matching via differentiable patchmatch. In *International Conference on Computer Vision*, pages 4384–4393, 2019.

[9] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

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

[11] S. Gidaris and N. Komodakis. Detect, replace, refine: Deep structured prediction for pixel wise labeling. *computer vision and pattern recognition*, pages 7187–7196, 2017.

[12] F. Guney and A. Geiger. Displets: Resolving stereo ambiguities using object knowledge. 2015.

[13] X. Guo, K. Yang, W. Yang, X. Wang, and H. Li. Group-wise correlation stereo network. *arXiv: Computer Vision and Pattern Recognition*, 2019.

[14] R. Haeusler, R. Nair, and D. Kondermann. Ensemble learning for confidence measures in stereo vision. pages 305–312, 2014.

[15] H. Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341, 2008.

[16] A. Hosni, M. Bleyer, M. Gelautz, and C. Rhemann. Local stereo matching using geodesic support weights. In *International Conference on Image Processing*, pages 2093–2096, 2009.

[17] G. Huang, Z. Liu, L. V. Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *IEEE Conference onComputer Vision and Pattern Recognition*, pages 2261–2269, 2017.

[18] Z. Jie, P. Wang, Y. Ling, B. Zhao, Y. Wei, J. Feng, and W. Liu. Left-right comparative recurrent model for stereo matching. In *IEEE Conference onComputer Vision and Pattern Recognition*, pages 3838—-3846, 2018.

[19] Z. Jie, P. Wang, Y. Ling, B. Zhao, Y. Wei, J. Feng, and W. Liu. Left-right comparative recurrent model for stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3838–3846, 2018.

[20] 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. *CoRR, vol. abs/1703.04309*, 2017.

[21] 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.

[22] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *international conference on learning representations*, 2015.

[23] A. Klaus, M. Sormann, and K. Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *IEEE International Conference on Pattern Recognition*, volume 3, pages 15–18, 2006.

[24] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions using graph cuts. In *IEEE International Conference on Computer Vision*, volume 2, pages 508–515, 2001.

[25] N. Komodakis and N. Paragios. Beyond pairwise energies: Efficient optimization for higher-order mrfs. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2985–2992, 2009.

[26] D. Kong and H. Tao. A method for learning matching errors for stereo computation. In *Proceedings of the British Machine Vision Conference*, pages 11.1–11.10. BMVA Press, 2004. doi:10.5244/C.18.11.

[27] D. Kong and H. Tao. H.: Stereo matching via learning multiple experts behaviors. *In: BMVC*, pages 97–106, 2006.

[28] G. R. Lee, R. Gommers, F. Wasilewski, K. Wohlfahrt, and A. O'Leary. Pywavelets: A python package for wavelet analysis. *Journal of Open Source Software*, 4(36):1237, 2019.

[29] H. Li, B. S. Manjunath, and S. K. Mitra. Multisensor image fusion using the wavelet transform. *Graphical Models and Image Processing*, 57(3):235–245, 1995.

[30] W. Luo, A. G. Schwing, and R. Urtasun. Efficient deep learning for stereo matching. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[31] 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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[32] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[33] D. Miyazaki, Y. Matsushita, and K. Ikeuchi. Interactive shadow removal from a single image using hierarchical graph cut. *Asian Conference on Computer Vision*, pages 234–245, 2010.

[34] S. Naik and N. Patel. Single image super resolution in spatial and wavelet domain. *The International Journal of Multimedia & Its Applications*, 5(4):23–32, 2013.

[35] J. Pang, W. Sun, J. S. Ren, C. Yang, and Q. Yan. Cascade residual learning: A two-stage convolutional neural network for stereo matching. In *ICCV Workshops*, volume 7, 2017.

[36] N. Papadakis and V. Caselles. Multi-label depth estimation for graph cuts stereo problems. *Journal of Mathematical Imaging and Vision*, 38(1):70–82, 2010.

[37] M. Peris, S. Martull, A. Maki, Y. Ohkawa, and K. Fukui. Towards a simulation driven stereo vision system. In *Proceedings of the 21st International Conference on Pattern Recognition*, pages 1038–1042, 2012.

[38] D. Scharstein and C. Pal. Learning conditional random fields for stereo. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[39] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47((1/2/3)):7–42, 2002.

[40] A. Seki and M. Pollefeys. Patch based confidence prediction for dense disparity map. *British Machine Vision Conference (BMVC)*, 2016.

[41] 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.

[42] A. Spyropoulos, N. Komodakis, and P. Mordohai. Learning to detect ground control points for improving the accuracy of stereo matching. pages 1621–1628, 20014.

[43] J. Starck, E. J. Candes, and D. L. Donoho. The curvelet transform for image denoising. *IEEE Transactions on Image Processing*, 11(6):670–684, 2002.

[44] H. Tong, M. Li, H. Zhang, and C. Zhang. Blur detection for digital images using wavelet transform. 1:17–20, 2004.

[45] S. Tulyakov, A. Ivanov, and F. Fleuret. Practical deep stereo (pds): Toward applications-friendly deep stereo matching. *arXiv preprint arXiv:1806.01677*, 2018.

[46] T. Williams and R. Li. Wavelet pooling for convolutional neural networks. 2018.

[47] O. Woodford, P. Torr, I. Reid, and A. Fitzgibbon. Global stereo reconstruction under second-order smoothness priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12):2115–2128, 2009.

[48] M. Yang, Y. Liu, and Z. You. The euclidean embedding learning based on convolutional neural network for stereo matching. *Neurocomputing*, 267:195–200, 2017.

[49] M. Yang, Y. Liu, Z. You, X. Li, and Y. Zhang. A homography transform based higher-order mrf model for stereo matching. *Pattern Recognition Letters*, 40:66–71, 2014.

[50] M. Yang and X. Lv. Learning both matching cost and smoothness constraint for stereo matching. *Neurocomputing*, 2018.

[51] Q. Yang, L. Wang, R. Yang, H. Stewénius, and D. Nistér. Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(3):492–504, 2009.

[52] Q. Yang, R. Yang, J. Davis, and D. Nistér. Spatial-depth super resolution for range images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[53] K.-J. Yoon and I.-S. Kweon. Adaptive support-weight approach for correspondence search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(4):650–656, 2006.

[54] L. Yu, Y. Wang, Y. Wu, and Y. Jia. Deep stereo matching with explicit cost aggregation sub-architecture. *arXiv: Computer Vision and Pattern Recognition*, 2018.

[55] J. Zbontar and Y. LeCun. Stereo matching by training a convolutional neural network to compare image patches. *Submitted to JMLR*, 2015.

[56] L. Zhang and S. M. Seitz. Estimating optimal parameters for mrf stereo from a single image pair. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2):331–342, 2007.

[57] Y. Zhang, S. Khamis, C. Rhemann, J. Valentin, A. Kowdle, V. Tankovich, M. Schoenberg, S. Izadi, T. Funkhouser, and S. Fanello. Activestereonet: End-to-end self-supervised learning for active stereo systems. *arXiv preprint arXiv:1807.06009*, 2018.

[58] Y. Zhong, Y. Dai, and H. Li. Self-supervised learning for stereo matching with self-improving ability. *arXiv preprint arXiv:1709.00930*, 2017.

[59] Y. Zhong, H. Li, and Y. Dai. Open-world stereo video matching with deep rnn. In *European Conference on Computer Vision*, pages 104–119, 2018.

[60] C. Zitnick and S. Kang. Stereo for image-based rendering using image over-segmentation. *International Journal of Computer Vision*, 75(1):49–65, 2007.