Group-wise Correlation Stereo Network

Xiaoyang Guo[1] Kai Yang[2] Wukui Yang[2] Xiaogang Wang[1] Hongsheng Li[1]
[1] The Chinese University of Hong Kong [2] SenseTime Research
{xyguo, xgwang, hsli}@ee.cuhk.edu.hk {yangkai, yangwukui}@sensetime.com

Abstract

Stereo matching estimates the disparity between a rectified image pair, which is of great importance to depth sensing, autonomous driving, and other related tasks. Previous works built cost volumes with cross-correlation or concatenation of left and right features across all disparity levels, and then a 2D or 3D convolutional neural network is utilized to regress the disparity maps. In this paper, we propose to construct the cost volume by group-wise correlation. The left features and the right features are divided into groups along the channel dimension, and correlation maps are computed among each group to obtain multiple matching cost proposals, which are then packed into a cost volume. Group-wise correlation provides efficient representations for measuring feature similarities and will not lose too much information like full correlation. It also preserves better performance when reducing parameters compared with previous methods. The 3D stacked hourglass network proposed in previous works is improved to boost the performance and decrease the inference computational cost. Experiment results show that our method outperforms previous methods on Scene Flow, KITTI 2012, and KITTI 2015 datasets. The code is available at `https://github.com/xy-guo/GwcNet`

1. Introduction

Accurate depth sensing plays an important role in many computer vision applications like odometry, robot navigation, pose estimation, and object detection [10, 28, 6, 16]. Unlike monocular depth estimation [3, 5] or active depth sensing [24], stereo matching estimates depth by matching pixels from rectified image pairs captured by two cameras.

Traditional stereo pipelines usually consist of all or portion of the following four steps, matching cost computation, cost aggregation, disparity optimization, and post-processing [23]. Matching cost computation provides initial similarity measures for left image patches and possible corresponding right patches, which is a crucial step of stereo matching. Common matching costs include absolute difference (SAD), sum of squared difference (SSD), and normalized cross-correlation (NCC). The cost aggregation and the optimization steps incorporate contextual matching costs and priors to obtain more robust disparity predictions.

Learning-based methods explore different feature representations and aggregation algorithms for matching costs. DispNetC [19] computes a correlation volume from the left and right image features and utilizes a CNN to directly regress disparity maps. GC-Net [9] and PSMNet [2] construct concatenation-based feature volume and incorporate a 3D CNN to aggregate contextual features. There are also works [1, 25] trying to aggregate evidence from multiple hand-crafted matching cost proposals. However, the above methods have several drawbacks. The full correlation [19] provides an efficient way for measuring feature similarities, but it loses much information because it produces only a single-channel correlation map for each disparity level. The concatenation volume [9, 2] requires more parameters in the following aggregation network to learn the similarity measurement function from scratch. [1, 25] stills utilizes traditional matching costs and cannot be optimized end-to-end.

In this paper, we propose a simple yet efficient operation called group-wise correlation to tackle the above drawbacks. Multi-level unary features are extracted to form high-dimensional feature representations $\mathbf{f}_l, \mathbf{f}_r$ for a left-right image pair. Then, the features are split into multiple groups along the channel dimension, and the ith left feature group is correlated with the corresponding ith right feature group over all disparity levels to obtain group-wise correlation maps. At last, all the correlation maps are packed to form a 4D cost volume. The unary features can be treated as groups of structured vectors [32], so the correlation maps for a certain group can be seen as a matching cost proposal. In this way, we can leverage the power of traditional correlation matching cost and provide better similarity measures for the following 3D aggregation network compared with [9, 2]. The multiple matching cost proposals also avoid the information loss like full correlation [19].

The 3D stacked hourglass aggregation network proposed in PSMNet [2] is modified to further improve the performance and decrease the inference computational cost.

$1\times1\times1$ 3D convolutions are employed in the shortcut connections within each hourglass module without increasing too much computational cost.

Our main contributions can be summarized as follows. 1) We propose group-wise correlation to construct cost volumes to provide better similarity measures. 2) The stacked 3D hourglass refinement network is modified to improve the performance without increasing the inference time. 3) Our method achieves better performance than previous methods on Scene Flow, KITTI 2012, and KITTI 2015 datasets. 4) Experiment results show that when limiting the computational cost of the 3D aggregation network, the performance reduction of our proposed network is much smaller than previous PSMNet, which makes group-wise correlation a valuable way to be implemented in real-time stereo networks.

2. Related Work

2.1. Traditional methods

Generally, traditional stereo matching consists of all or portion of the following four steps: matching cost computation, cost aggregation, disparity optimization, and some post-processing steps [23]. In the first step, the matching costs of all pixels are computed for all possible disparities. Common matching costs include sum of absolute difference (SAD), sum of squared difference (SSD), normalized cross-correlation (NCC), and so on. Local methods [37, 34, 20] explore different strategies to aggregate matching costs with neighbor pixels and usually utilize the winner-take-all (WTA) strategy to choose the disparity with minimum matching cost. In contrast, global methods minimize a target function to solve the optimal disparity map, which usually takes both matching costs and smoothness priors into consideration, such as belief propagation [30, 13] and graph cut [15]. Semi-global matching (SGM) [7] approximates the global optimization with dynamic programming. Local and global methods can be combined to obtain better performance and robustness.

2.2. Learning based methods

Besides hand-crafted methods, researchers also proposed many learned matching costs [36, 18, 27] and cost aggregation algorithms [1, 25]. Zbontar and Lecun [36] first proposed to compute matching costs using neural networks. The predicted matching costs are then processed with traditional cross-based cost aggregation and semi-global matching to predict the disparity map. The matching cost computation was accelerated in [18] by correlating unary features. Batsos *et al.* proposed CBMV [1] to combine evidence from multiple basic matching costs. Schonberger *et al.* [25] proposed to classify scanline matching cost candidates with a random forest classifier. Seki *et al.* proposed SGM-Nets [26] to provide learned penalties for SGM. Knobelreiter *et al.* [14] proposed to combine CNN-predicted correlation matching costs and CRF to integrate long-range interactions.

Following DispNetC (Mayer *et al.* [19]), there are a lot of works directly regressing disparity maps from correlation cost volumes [22, 17, 29, 33]. Given the left and the right feature maps $\mathbf{f}_l$ and $\mathbf{f}_r$, the correlation cost volume is computed for each disparity level d,

$$\mathbf{C}_{corr}(d,x,y) = \frac{1}{N_c}\langle \mathbf{f}_l(x,y), \mathbf{f}_r(x-d,y)\rangle, \qquad (1)$$

where $\langle\cdot,\cdot\rangle$ is the inner product of two feature vectors and N_c denotes the number of channels. CRL [22] and iResNet [17] followed the idea of DispNetC with stack refinement sub-networks to further improve the performance. There are also works integrating additional information such as edge features [29] and semantic features [33].

Recent works employed concatenation-based feature volume and 3D aggregation networks for better context aggregation [9, 2, 35]. Kendall *et al.* proposed GC-Net [9] and was the first to use 3D convolution networks to aggregate context for cost volumes. Instead of directly giving a cost volume, the left and the right feature $\mathbf{f}_l$, $\mathbf{f}_r$ are concatenated to form a 4D feature volume,

$$\mathbf{C}_{concat}(d,x,y,\cdot) = \text{Concat}\left\{\mathbf{f}_l(x,y), \mathbf{f}_r(x-d,y)\right\}. \quad (2)$$

Context features are aggregated from neighbour pixels and disparities with 3D convolution networks to predict a disparity probability volume. Following GC-Net, Chang *et al.* [2] proposed the pyramid stereo matching network (PSMNet) with a spatial pyramid pooling module and stacked 3D hourglass networks for cost volume refinement. Yu *et al.* [35] proposed to generate and select multiple cost aggregation proposals. Zhong *et al.* [38] proposed a self-adaptive recurrent stereo model to tackle open-world data.

LRCR [8] utilized left-right consistency check and recurrent model to aggregate cost volumes predicted from [27] and refined unreliable disparity predictions. There are also other works focusing on real-time stereo matching [11] and application friendly stereo [31].

3. Group-wise Correlation Network

We propose group-wise correlation stereo network (GwcNet), which extends PSMNet [2] with group-wise correlation cost volume and improved 3D stacked hourglass networks.

3.1. Network architecture

The structure of the proposed group-wise correlation network is shown in Figure 1. The network consists of four

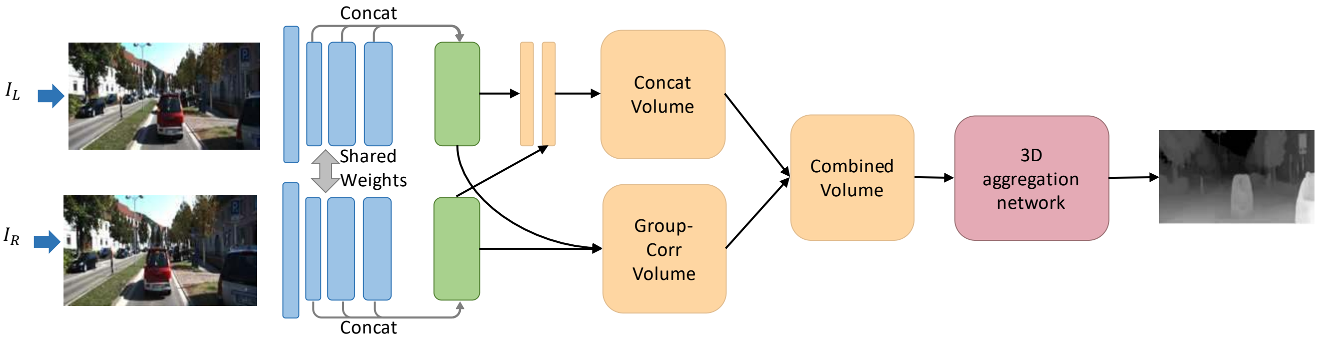


Figure 1: The pipeline of the proposed group-wise correlation network. The whole network consists of four parts, unary feature extraction, cost volume construction, 3D convolution aggregation, and disparity prediction. The cost volume is divided into two parts, concatenation volume (*Cat*) and group-wise correlation volume (*Gwc*). Concatenation volume is built by concatenating the compressed left and right features. Group-wise correlation volume is described in Section 3.2.

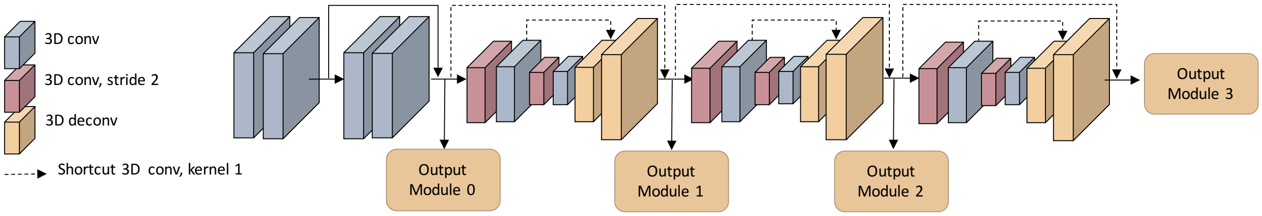


Figure 2: The structure of our proposed 3D aggregation network. The network consists of a pre-hourglass module (four convolutions at the beginning) and three stacked 3D hourglass networks. Compared with PSMNet [2], we remove the shortcut connections between different hourglass modules and output modules, thus output modules 0,1,2 can be removed during inference to save time. $1\times1\times1$ 3D convolutions are added to the shortcut connections within hourglass modules.

parts, unary feature extraction, cost volume construction, 3D aggregation, and disparity prediction (details in Table 1).

For feature extraction, we adopt the ResNet-like network used in PSMNet [2] with the half dilation settings and without its spatial pyramid pooling module. The last feature maps of conv2, conv3, and conv4 are concatenated to form 320-channel unary feature maps.

The cost volume is composed of two parts, a concatenation volume and a group-wise correlation volume. The concatenation volume is the same as PSMNet [2] but with fewer channels, before which the unary features are compressed into 12 channels with two convolutions. The proposed group-wise correlation volume will be described in details in Section 3.2. The two volumes are then concatenated as the input to the 3D aggregation network.

The 3D aggregation network is used to aggregate features from neighboring disparities and pixels, which consists of a pre-hourglass module and three stacked 3D hourglass networks. As shown in Figure 2, the pre-hourglass module consists of four 3D convolutions with batch normalization and ReLU. Three stacked 3D hourglass networks are followed to refine low-texture ambiguities and occlusion parts by encoder-decoder structures. Compared with 3D aggregation network of [2], we have several important modifications to improve the performance and increase the inference speed, and details are described in Section 3.3.

The pre-hourglass and three stacked 3D hourglass networks are connected to output modules. Each output module predicts a disparity map. The structure of the output module and the loss function are described in Section 3.4.

3.2. Group-wise correlation volume

The left unary features and the right unary features are denoted by $\mathbf{f}_l$ and $\mathbf{f}_r$ with N_c channels and in $1/4$ the size of original images. In previous works [19, 9, 2], the left and right features are correlated or concatenated at different disparity levels to form the cost volume. However, both correlation volume and concatenation volume have drawbacks. The full correlation provides an efficient way for measuring feature similarities, but it loses much information because it produces only a single-channel correlation map for each disparity level. The concatenation volume contains no information about the feature similarities, so more parameters are required in the following aggregation network to learn the similarity measurement function from scratch. To solve the above issues, we propose group-wise correlation by combining advantages of the concatenation volume and the correlation volume.

The basic idea behind group-wise correlation is splitting the features into groups and computing correlation maps group by group. We denote the channels of unary features as N_c. All the channels are evenly divided into N_g groups along the channel dimension, and each feature group therefore has N_c/N_g channels. The gth feature group $\mathbf{f}_l^g, \mathbf{f}_r^g$ consists of the $g\frac{N_c}{N_g}, g\frac{N_c}{N_g}+1, \ldots, g\frac{N_c}{N_g}+(\frac{N_c}{N_g}-1)$th channels of the original feature $\mathbf{f}_l, \mathbf{f}_r$. The group-wise correlation is then computed as

$$\mathbf{C}_{gwc}(d, x, y, g) = \frac{1}{N_c/N_g}\langle \mathbf{f}_l^g(x, y), \mathbf{f}_r^g(x-d, y)\rangle, \tag{3}$$

where $\langle\cdot,\cdot\rangle$ is the inner product. Note that the correlation is computed for all feature groups g and at all disparity levels d. Then, all the correlation maps are packed into a matching cost volume of the shape $[D_{max}/4, H/4, W/4, N_g]$, where D_{max} denotes the maximum disparity and $D_{max}/4$ corresponds to the maximum disparity for the feature. When N_g=1, the group-wise correlation becomes full correlation.

Group-wise correlation volume $\mathbf{C}_{gwc}$ can be treated as N_g cost volume proposals, and each proposal is computed from the corresponding feature group. The following 3D aggregation network aggregates multiple candidates to regress disparity maps. The group-wise correlation successfully leverages the power of traditional correlation matching costs and provides rich similarity-measure features for the 3D aggregation network, which alleviates the parameter demand. We will show in Section 4.5 that we explore to reduce the channels of the 3D aggregation network, and the performance reduction of our proposed network is much smaller than [2]. Our proposed group-wise correlation volume requires less 3D aggregation parameters to achieve favorable results.

To further improve the performance, the group correlation cost volume can be combined with the concatenation volume. Experiment results show that the group-wise correlation volume and the concatenation volume are complementary to each other.

3.3. Improved 3D aggregation module

In PSMNet [2], a stacked hourglass architecture was proposed to learn better context features. Based on the network, we apply several important modifications to make it suitable for our proposed group-wise correlation and improve the inference speed. The structure of the proposed 3D aggregation is shown in Figure 2 and Table 1.

1) First, we add one more auxiliary output module (see output module 0 in Figure 2) after the pre-hourglass module. The extra auxiliary loss makes the network learn better features at lower layers, which benefits the final prediction. 2) The residual connections between different output modules are removed, thus auxiliary output modules (output module 0, 1, 2) can be removed during inference to save computational cost. 3) $1\times1\times1$ 3D convolutions are added to the shortcut connections within each hourglass module (see dashed lines in Figure 2) to improve the performance without increasing much computational cost. Since the $1\times1\times1$ 3D convolution only has 1/27 multiplication operations compared with $3\times3\times3$ convolutions, it runs very fast and the time can be neglected.

Name	Layer properties	Output size
Cost Volume		
unary_l/r	N/A, S2	$^{H}/_{4}\times{}^{W}/_{4}\times 320$
volume_g	group-wise cost volume	$^{D}/_{4}\times{}^{H}/_{4}\times{}^{W}/_{4}\times 40$
volume_c	concatenation cost volume	$^{D}/_{4}\times{}^{H}/_{4}\times{}^{W}/_{4}\times 24$
volume	volume_g,volume_c: Concat	$^{D}/_{4}\times{}^{H}/_{4}\times{}^{W}/_{4}\times 64$
Pre-hourglass		
conv1	[32×32, 3×3×3, S1] ×2	$^{D}/_{4}\times{}^{H}/_{4}\times{}^{W}/_{4}\times 32$
conv2	[32×32, 3×3×3, S1] ×2	$^{D}/_{4}\times{}^{H}/_{4}\times{}^{W}/_{4}\times 32$
output	conv1,conv2: Add	$^{D}/_{4}\times{}^{H}/_{4}\times{}^{W}/_{4}\times 32$
Hourglass Module 1, 2, 3		
input	N/A	$^{D}/_{4}\times{}^{H}/_{4}\times{}^{W}/_{4}\times 32$
conv1a	32×64, 3×3×3, S2	$^{D}/_{8}\times{}^{H}/_{8}\times{}^{W}/_{8}\times 64$
conv1b	64×64, 3×3×3, S1	$^{D}/_{8}\times{}^{H}/_{8}\times{}^{W}/_{8}\times 64$
conv2a	64×128, 3×3×3, S2	$^{D}/_{16}\times{}^{H}/_{16}\times{}^{W}/_{16}\times 128$
conv2b	128×128, 3×3×3, S1	$^{D}/_{16}\times{}^{H}/_{16}\times{}^{W}/_{16}\times 128$
deconv1*	128×64, 3×3×3, S2, deconv	$^{D}/_{8}\times{}^{H}/_{8}\times{}^{W}/_{8}\times 64$
shortcut1*	conv1b: 64×64, 1×1×1, S1	$^{D}/_{8}\times{}^{H}/_{8}\times{}^{W}/_{8}\times 64$
plus1	deconv1,shortcut1: Add&ReLU	$^{D}/_{8}\times{}^{H}/_{8}\times{}^{W}/_{8}\times 64$
deconv0*	64×32, 3×3×3, S2, deconv	$^{D}/_{4}\times{}^{H}/_{4}\times{}^{W}/_{4}\times 32$
shortcut0*	input: 32×32, 1×1×1, S1	$^{D}/_{4}\times{}^{H}/_{4}\times{}^{W}/_{4}\times 32$
output	deconv0,shortcut0: Add&ReLU	$^{D}/_{4}\times{}^{H}/_{4}\times{}^{W}/_{4}\times 32$
Output Module 0, 1, 2, 3		
input	N/A	$^{D}/_{4}\times{}^{H}/_{4}\times{}^{W}/_{4}\times 32$
conv1	32×32, 3×3×3, S1	$^{D}/_{4}\times{}^{H}/_{4}\times{}^{W}/_{4}\times 32$
conv2**	32×1, 3×3×3, S1	$^{D}/_{4}\times{}^{H}/_{4}\times{}^{W}/_{4}\times 1$
score	Upsample	$D\times H\times W\times 1$
prob	Softmax (at disparity dimension)	$D\times H\times W\times 1$
disparity	Soft Argmin (Equ. 4)	$H\times W\times 1$

Table 1: Structure details of the modules. H, W represents the height and the width of the input image. S1/2 denotes the convolution stride. If not specified, each 3D convolution is with a batch normalization and ReLU. * denotes the ReLU is not included. ** denotes convolution only.

3.4. Output module and loss function

For each output module, two 3D convolutions are employed to generate a 1-channel 4D volume, and then the volume is upsampled and converted into a probability volume with softmax function along the disparity dimension. Detailed structures are shown in Table 1. For each pixel, we have a D_{max}-length vector which contains the probability p for all disparity levels. Then, the disparity estimation $\widetilde{d}$ is

Model	Concat Volume	Group Corr Volume	Stack Hour-glass	Groups × Channels	Init Volume Channel	>1px (%)	>2px (%)	>3px (%)	EPE (px)	Time (ms)
Cat64-Base	✓			-	64	12.78	8.05	6.33	1.308	117.1
Gwc1-Base		✓		1×320	1	13.32	8.37	6.62	1.369	104.0
Gwc10-Base		✓		10×32	10	11.82	7.31	5.70	1.230	112.8
Gwc20-Base		✓		20×16	20	11.84	7.29	5.67	1.216	116.3
Gwc40-Base		✓		40×8	40	11.68	7.18	5.58	1.212	122.2
Gwc80-Base		✓		80×4	80	11.69	7.17	5.57	1.214	133.3
Gwc160-Base		✓		160×2	160	11.58	7.08	5.49	1.188	157.3
Gwc40-Cat24-Base	✓	✓		40×8	40+24	**11.26**	**6.87**	**5.31**	**1.127**	135.1
PSMNet [2]	✓		[2]	-	64	9.46	5.19	3.80	0.887	246.1
Cat64-original-hg	✓		[2]	-	64	9.47	5.13	3.74	0.876	241.0
Cat64	✓		Ours	-	64	8.41	4.63	3.41	0.808	198.3
Gwc40 (GwcNet-g)		✓	Ours	40×8	40	8.18	4.57	3.39	0.792	200.3
Gwc40-Cat24 (GwcNet-gc)	✓	✓	Ours	40×8	40+24	**8.03**	**4.47**	**3.30**	**0.765**	210.7

Table 2: Ablation study results of proposed networks on the Finalpass of Scene Flow datasets [19]. *Cat*, *Gwc*, *Gwc-Cat* represent only concatenation volume, only group-wise correlation volume, or the both. *Base* denotes the network variants without stacked hourglass networks. The time is the inference time for 480×640 inputs on a single Nvidia TITAN Xp GPU. The result of PSMNet [2] is trained with published code with our batch size, evaluation settings for fair comparison.

Model	KITTI 12 EPE (px)	KITTI 12 D1-all(%)	KITTI 15 EPE (px)	KITTI 15 D1-all (%)
PSMNet [2]	0.713	2.53	0.639	1.50
Cat64-original-hg	0.740	2.72	0.652	1.76
Cat64	0.691	2.41	0.615	1.55
Gwc40	0.662	2.30	**0.602**	**1.41**
Gwc40-Cat24	**0.659**	**2.10**	0.613	1.49

Table 3: Ablation study results of our networks on KITTI 2012 validation and KITTI 2015 validation sets.

given by the soft argmin function [9],

$$\widetilde{d} = \sum_{k=0}^{D_{max}-1} k \cdot p_k, \tag{4}$$

where k and p_k denote a possible disparity level and the corresponding probability. The predicted disparity maps from the four output modules are denoted as $\widetilde{\mathbf{d}}_0, \widetilde{\mathbf{d}}_1, \widetilde{\mathbf{d}}_2, \widetilde{\mathbf{d}}_3$. The final loss is given by,

$$L = \sum_{i=0}^{i=3} \lambda_i \cdot \text{Smooth}_{L_1}(\widetilde{\mathbf{d}}_i - \mathbf{d}^*), \tag{5}$$

where λ_i denotes the coefficients for the ith disparity prediction and $\mathbf{d}^*$ represents the ground-truth disparity map. The smooth L1 loss is computed as follows,

$$\text{Smooth}_{L_1}(x) = \begin{cases} 0.5x^2, & \text{if } |x| < 1 \\ |x| - 0.5, & \text{otherwise} \end{cases} \tag{6}$$

4. Experiment

In this section, we evaluate our proposed stereo models on Scene Flow datasets [19] and the KITTI dataset [4, 21]. Datasets and implementation details are described in Section 4.1 and Section 4.2. The effectiveness and the best settings of group-wise correlation are explored in Section 4.3. The performance improvement of the new stacked hourglass module is discussed in Section 4.4. We also explore the performance of group-wise correlation when the computational cost is limited in Section 4.5.

4.1. Datasets and evaluation metrics

Scene Flow datasets are a dataset collection of synthetic stereo datasets, consisting of Flyingthings3D, Driving, and Monkaa. The datasets provide 35,454 training and 4,370 testing images of size 960×540 with accurate ground-truth disparity maps. We use the Finalpass of the Scene Flow datasets, since it contains more motion blur and defocus and is more like real-world images than the Cleanpass. **KITTI 2012** and **KITTI 2015** are driving scene datasets. KITTI 2012 provides 194 training and 195 testing images pairs, and KITTI 2015 provides 200 training and 200 testing image pairs. Both datasets provide sparse LIDAR ground-truth disparity for the training images.

For Scene Flow datasets, the evaluation metrics is usually the end-point error (EPE), which is the mean average disparity error in pixels. For KITTI 2012, percentages of erroneous pixels and average end-point errors for both non-occluded (Noc) and all (All) pixels are reported. For KITTI 2015, the percentage of disparity outliers *D1* is evaluated for background, foreground, and all pixels. The outliers are defined as the pixels whose disparity errors are larger than $\max(3\text{px}, 0.05d^*)$, where d^* denotes the ground-truth disparity.

4.2. Implementation details

Our network is implemented with PyTorch. We use Adam [12] optimizer, with $\beta_1 = 0.9$, $\beta_2 = 0.999$. The batch size is fixed to 16, and we train all the networks with 8 Nvidia TITAN Xp GPUs with 2 training samples on each GPU. The coefficients of four outputs are set as $\lambda_0 = 0.5$, $\lambda_1 = 0.5$, $\lambda_2 = 0.7$, $\lambda_3 = 1.0$.

For Scene Flow datasets, we train the stereo networks for 16 epochs. The learning rate is set to 0.001 and down-scaled by 2 after epoch 10, 12, and 14. To test on Scene Flow datasets, the full images of size 960×540 are input to the network for disparity prediction. We set the maximum disparity value as $D_{max} = 192$ following PSMNet [2] for Scene Flow datasets. To evaluate our networks, we remove all the images with less than 10% valid pixels ($0 \leq d < D_{max}$) in the test set. For each valid image, the evaluation metrics are computed with only valid pixels.

For KITTI 2015 and KITTI 2012, we fine-tune the network pre-trained on Scene Flow datasets for another 300 epochs. The initial learning rate is 0.001 and is down-scaled by 10 after epoch 200. For testing on KITTI datasets, we first pad zeros on the top and the right side of the images to make the inputs in size 1248×384.

4.3. The effectiveness of Group-wise correlation

In this section, we explore the effectiveness and the best settings for the group-wise correlation. In order to prove the effectiveness of the proposed group-wise correlation volume, we conduct several experiments on the *Base* model, which removes the stacked hourglass networks and only preserves the pre-hourglass module and the output module 0. *Cat-Base*, *Gwc-Base*, and *Gwc-Cat-Base* are the base models with only concatenation volume, only group-wise correlation volume, or both volumes.

Experiment results in Table 2 show that the performance of the *Gwc-Base* network increases as the group number increases. When the group number is larger than 40, the performance improvement becomes minor and the end-point error stays around 1.2px. Considering the memory usage and the computational cost, we choose 40 groups with each group having 8 channels as our network structure, which corresponds to the *Gwc40-Base* model in Table 2.

All the *Gwc-Base* models except *Gwc1-Base* outperform the *Cat-Base* model which utilizes concatenation volume, which shows the effectiveness of the group-wise correlation. The *Gwc40* model reduces the end-point error by 0.1px and the 3-pixel error rate by 0.75%, and the time consumption is almost the same. The performance can be further improved by combining group-wise correlation volume with concatenation volume (see *Gwc40-Cat24-Base* model in Table 2). The group-wise correlation could provide accurate matching features, and the concatenation volume provides complementary semantic information.

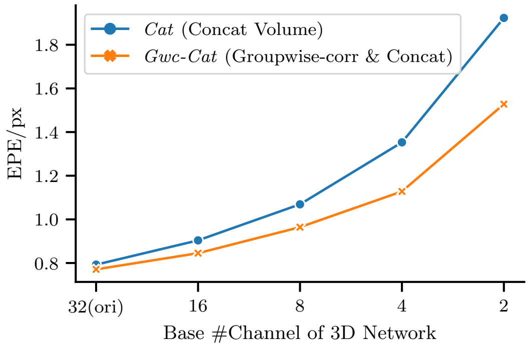


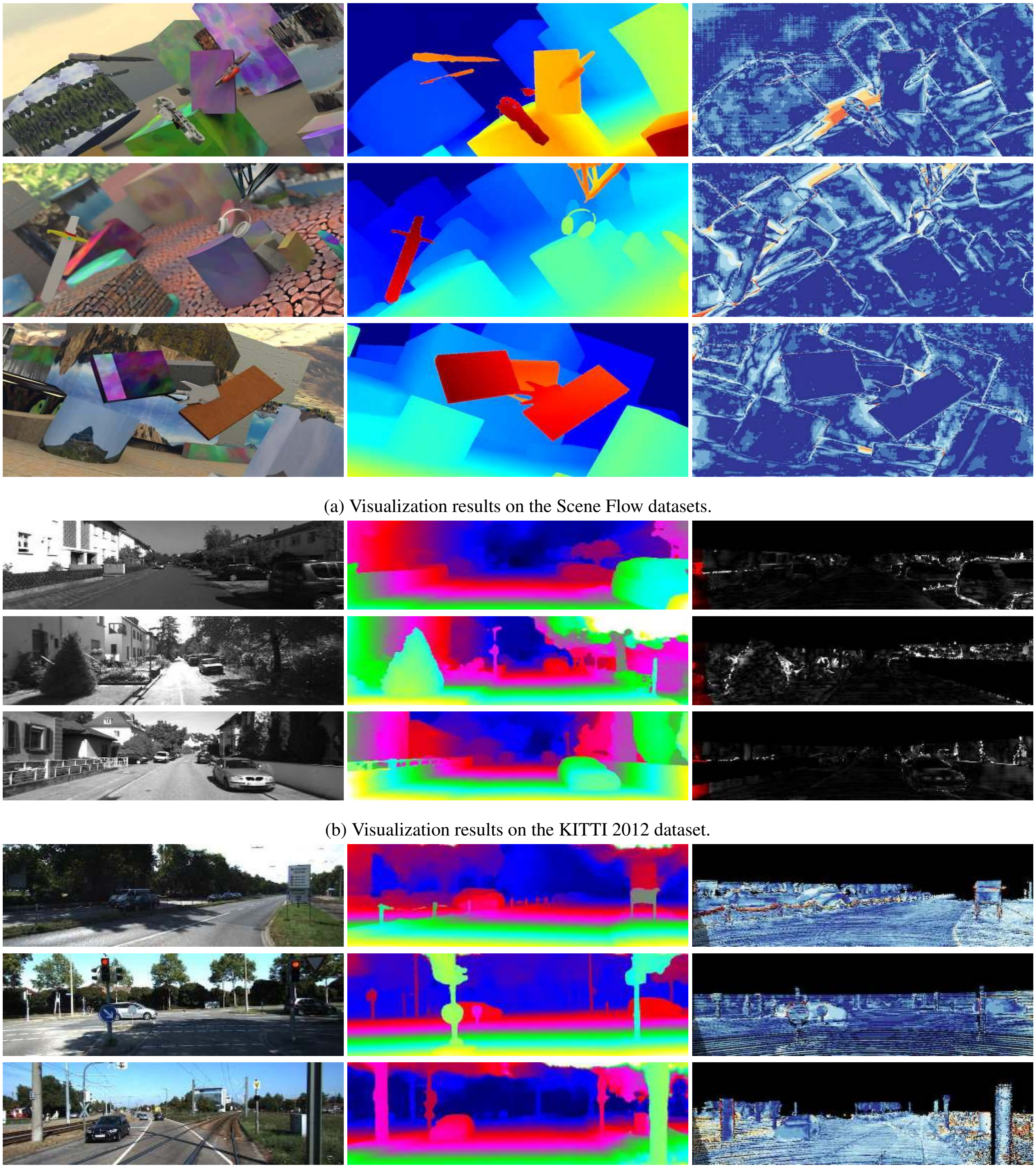
Figure 3: Our model *Gwc-Cat* achieves much better performance than *Cat* when the number of channels decreases. The models with 32 base channels correspond to the *Cat64* model (concatenation volume) and the *Gwc40-Cat24* model (group-wise correlation and concatenation volume). The channels of the cost volume and all 3D convolutions decrease by the same factor as the base channel.

4.4. Improved stacked hourglass

In this paper, we applied several modifications to the stacked hourglass networks proposed in [2] to improve the performance of cost volume aggregation. From Table 2 and Table 3, we can see that the model with the proposed hourglass networks (*Cat64*) increases EPE by 7.8% on Scene Flow datasets and 5.8% on KITTI 2015 compared with the model *Cat64-original-hg* (with the hourglass module in [2]). The inference time for 640×480 inputs on a single Nvidia TITAN Xp GPU also decreases by 42.7ms, because the auxiliary output modules can be removed during inference to save time.

4.5. Limit the computational cost of 3D network

We explore to limit the computational cost by decreasing channels in the 3D aggregation network to verify the effectiveness of the proposed group-wise group correlation. The results are shown in Figure 3. The base number of channels are modified from the original 32 to 2, and the channels of the cost volume and all 3D convolutions are reduced with the same factor. As the number of channels decreasing, our models with group-wise correlation volume (*Gwc-Cat*) perform much better than the models with only concatenation volume (*Cat*). The performance gain enlarges as more channels reduced. The reason for this is that the group-wise



(a) Visualization results on the Scene Flow datasets.

(b) Visualization results on the KITTI 2012 dataset.

(c) Visualization results on the KITTI 2015 dataset.

Figure 4: Depth visualization results on the test sets of Scene Flow [19], KITTI 2012 [4] and KITTI 2015 [21] datasets. From left to right, input left images, predicted disparity maps, and error maps.

	All (%)			Noc (%)			Time
	D1-bg	D1-fg	D1-all	D1-bg	D1-fg	D1-all	(s)
DispNetC [19]	4.32	4.41	4.34	4.11	3.72	4.05	0.06
GC-Net [9]	2.21	6.16	2.87	2.02	5.58	2.61	0.9
CRL [22]	2.48	3.59	2.67	2.32	3.12	2.45	0.47
iResNet-i2e2 [17]	2.14	3.45	2.36	1.94	3.20	2.15	0.22
PSMNet [9]	1.86	4.62	2.32	1.71	4.31	2.14	0.41
SegStereo [33]	1.88	4.07	2.25	1.76	3.70	2.08	0.6
GwcNet-g (Gwc40)	**1.74**	**3.93**	**2.11**	**1.61**	**3.49**	**1.92**	0.32

Table 4: KITTI 2015 test set results. The dataset contains 200 images for training and 200 images for testing.

	>2px (%)		>3px (%)		>5px (%)		Mean Error (px)		Time
	Noc	All	Noc	All	Noc	All	Noc	All	(s)
DispNetC [19]	7.38	8.11	4.11	4.65	2.05	2.39	0.9	1.0	0.06
MC-CNN-acrt [36]	3.90	5.45	2.43	3.63	1.64	2.39	0.7	0.9	67
GC-Net [9]	2.71	3.46	1.77	2.30	1.12	1.46	0.6	0.7	0.9
iResNet-i2 [17]	2.69	3.34	1.71	2.16	1.06	1.32	0.5	0.6	0.12
SegStereo [33]	2.66	3.19	1.68	2.03	1.00	1.21	0.5	0.6	0.6
PSMNet [9]	2.44	3.01	1.49	1.89	0.90	1.15	0.5	0.6	0.41
GwcNet-gc (Gwc40-Cat24)	**2.16**	**2.71**	**1.32**	**1.70**	**0.80**	**1.03**	**0.5**	**0.5**	0.32

Table 5: KITTI 2012 test set results. The dataset contains 194 images for training and 195 images for testing.

correlation provides good matching cost representations for the 3D aggregation network, while the aggregation network with only concatenation volume as inputs needs to learn the matching similarity function from scratch, which usually requires more parameters and computational cost. As a result, the proposed group-wise correlation could be a valuable method to be implemented in real-time stereo networks where the computational costs are limited.

4.6. KITTI 2012 and KITTI 2015

For KITTI stereo 2015 [21], we split the training set into 180 training image pairs and 20 validation image pairs. Since the results on the validation set are not stable, we fine-tune the pretrained model for 3 times and choose the model with the best validation performance. From Table 3, the performance of both *Gwc40-Cat24* and *Gwc40* is better than the models without group-wise correlation (*Cat64*, *Cat64-original-hg*). We submit the *Gwc40* model (without concatenation volume) with the lowest validation error to the evaluation server, and the results on the test set are shown in Table 4. Our model surpasses the PSMNet [2] by 0.21% and SegStereo [33] by 0.14% on D1-all.

For KITTI 2012 [4], we split the training set into 180 training images and 14 validation image pairs. The results on the validation set are shown in Table 3. We submit the best *Gwc40-Cat24* model on the validation set to the evaluation server. The evaluation results on the test set are shown in Table 5. Our method surpasses PSMNet [2] by 0.19% on 3-pixel-error and 0.1px on mean disparity error.

5. Conclusion

In this paper, we proposed GwcNet to estimate disparity maps for stereo matching, which incorporates group-wise correlation to build up the cost volumes. The group-wise correlation volumes provide good matching features for the 3D aggregation network, which improves the performance and reduces the parameter requirements of the aggregation network. We showed that when the computational cost is limited, our model achieves larger gain than previous concatenation-volume based stereo networks. We also improved the stacked hourglass networks to further improve the performance and reduce the inference time. Experiments demonstrated the effectiveness of our proposed method on the Scene Flow datasets and the KITTI dataset.

Acknowledgements

This work is supported in part by SenseTime Group Limited, in part by the General Research Fund through the Research Grants Council of Hong Kong under Grants CUHK14202217, CUHK14203118, CUHK14205615, CUHK14207814, CUHK14213616, CUHK14208417, CUHK14239816, and in part by CUHK Direct Grant.

References

[1] Konstantinos Batsos, Changjiang Cai, and Philippos Mordohai. Cbmv: A coalesced bidirectional matching volume for disparity estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2060–2069, 2018.

[2] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5410–5418, 2018.

[3] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014.

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

[5] Xiaoyang Guo, Hongsheng Li, Shuai Yi, Jimmy Ren, and Xiaogang Wang. Learning monocular depth by distilling cross-domain stereo networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 484–500, 2018.

[6] Saurabh Gupta, Ross Girshick, Pablo Arbeláez, and Jitendra Malik. Learning rich features from rgb-d images for object detection and segmentation. In *European Conference on Computer Vision*, pages 345–360. Springer, 2014.

[7] Heiko Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 807–814. IEEE, 2005.

[8] Zequn Jie, Pengfei Wang, Yonggen Ling, Bo Zhao, Yunchao Wei, Jiashi Feng, and Wei 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.

[9] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 66–75, 2017.

[10] Christian Kerl, Jürgen Sturm, and Daniel Cremers. Robust odometry estimation for rgb-d cameras. In *2013 IEEE International Conference on Robotics and Automation*, pages 3748–3754. IEEE, 2013.

[11] Sameh Khamis, Sean Fanello, Christoph Rhemann, Adarsh Kowdle, Julien Valentin, and Shahram Izadi. Stereonet: Guided hierarchical refinement for real-time edge-aware depth prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 573–590, 2018.

[12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[13] Andreas Klaus, Mario Sormann, and Konrad Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 3, pages 15–18. IEEE, 2006.

[14] Patrick Knobelreiter, Christian Reinbacher, Alexander Shekhovtsov, and Thomas Pock. End-to-end training of hybrid cnn-crf models for stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2339–2348, 2017.

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

[16] Hongyang Li, Bo Dai, Shaoshuai Shi, Wanli Ouyang, and Xiaogang Wang. Feature Intertwiner for Object Detection. In *ICLR*, 2019.

[17] Zhengfa Liang, Yiliu Feng, Yulan Guo Hengzhu Liu Wei Chen, and Linbo Qiao Li Zhou Jianfeng Zhang. Learning for disparity estimation through feature constancy. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2811–2820, 2018.

[18] Wenjie Luo, Alexander G Schwing, and Raquel Urtasun. Efficient deep learning for stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5695–5703, 2016.

[19] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4040–4048, 2016.

[20] Xing Mei, Xun Sun, Weiming Dong, Haitao Wang, and Xiaopeng Zhang. Segment-tree based cost aggregation for stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 313–320, 2013.

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

[22] Jiahao Pang, Wenxiu Sun, Jimmy SJ Ren, Chengxi Yang, and Qiong Yan. Cascade residual learning: A two-stage convolutional neural network for stereo matching. In *ICCV Workshops*, volume 7, 2017.

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

[24] Daniel Scharstein and Richard Szeliski. High-accuracy stereo depth maps using structured light. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 1, pages I–I. IEEE, 2003.

[25] Johannes L Schonberger, Sudipta N Sinha, and Marc Pollefeys. Learning to fuse proposals from multiple scanline optimizations in semi-global matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 739–755, 2018.

[26] Akihito Seki and Marc Pollefeys. Sgm-nets: Semi-global matching with neural networks. In *Proceedings of the IEEE*

Conference on Computer Vision and Pattern Recognition, pages 231–240, 2017.

[27] Amit Shaked and Lior Wolf. Improved stereo matching with constant highway networks and reflective confidence learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4641–4650, 2017.

[28] Jamie Shotton, Ross Girshick, Andrew Fitzgibbon, Toby Sharp, Mat Cook, Mark Finocchio, Richard Moore, Pushmeet Kohli, Antonio Criminisi, Alex Kipman, et al. Efficient human pose estimation from single depth images. *IEEE transactions on pattern analysis and machine intelligence*, 35(12):2821–2840, 2013.

[29] Xiao Song, Xu Zhao, Hanwen Hu, and Liangji Fang. Edgestereo: A context integrated residual pyramid network for stereo matching. In *Asian Conference on Computer Vision (ACCV)*, 2018.

[30] Jian Sun, Nan-Ning Zheng, and Heung-Yeung Shum. Stereo matching using belief propagation. *IEEE Transactions on pattern analysis and machine intelligence*, 25(7):787–800, 2003.

[31] Stepan Tulyakov, Anton Ivanov, and Francois Fleuret. Practical deep stereo (pds): Toward applications-friendly deep stereo matching. In *Advances in Neural Information Processing Systems*, pages 5875–5885, 2018.

[32] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018.

[33] Guorun Yang, Hengshuang Zhao, Jianping Shi, Zhidong Deng, and Jiaya Jia. Segstereo: Exploiting semantic information for disparity estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 636–651, 2018.

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

[35] Lidong Yu, Yucheng Wang, Yuwei Wu, and Yunde Jia. Deep stereo matching with explicit cost aggregation subarchitecture. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[36] Jure Zbontar and Yann LeCun. Computing the stereo matching cost with a convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1592–1599, 2015.

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

[38] Yiran Zhong, Hongdong Li, and Yuchao Dai. Open-world stereo video matching with deep rnn. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 101–116, 2018.