On the Over-Smoothing Problem of CNN Based Disparity Estimation

Chuangrong Chen
Sun Yat-sen University
chenchr5@mail2.sysu.edu.cn

Xiaozhi Chen
DJI
cxz.thu@gmail.com

Hui Cheng*
Sun Yat-sen University
chengh9@mail.sysu.edu.cn

Abstract

Currently, most deep learning based disparity estimation methods have the problem of over-smoothing at boundaries, which is unfavorable for some applications such as point cloud segmentation, mapping, etc. To address this problem, we first analyze the potential causes and observe that the estimated disparity at edge boundary pixels usually follows multimodal distributions, causing over-smoothing estimation. Based on this observation, we propose a single-modal weighted average operation on the probability distribution during inference, which can alleviate the problem effectively. To integrate the constraint of this inference method into training stage, we further analyze the characteristics of different loss functions and demonstrate that cross entropy loss with Gaussian distribution further improves the performance consistently. For quantitative evaluation, we propose a novel metric that measures the disparity error in the local structure of edge boundaries. Experiments on various datasets using various networks show our method's effectiveness and general applicability.

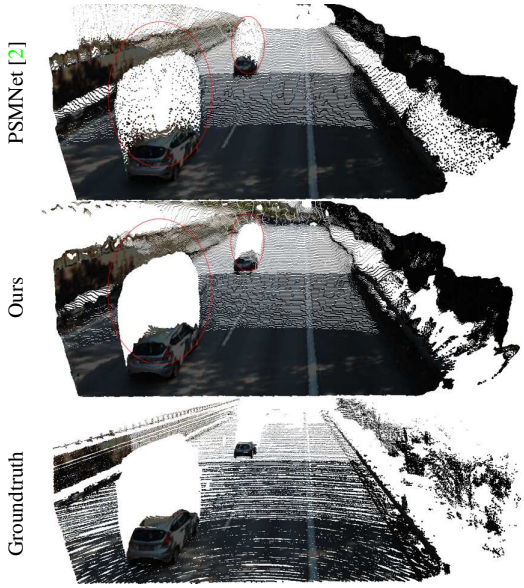


Figure 1. Point cloud converted from disparity. Please zoom in for more details.

1. Introduction

Given a calibrated stereo-rig, the problem of disparity estimation is to estimate per-pixel horizontal displacement from left image to right image or vice versa. If the intrinsic of the stereo-rig is known, per-pixel depth can be calculated by $depth = \frac{f \cdot b}{disp}$, where b is stereo baseline, f is the focal length and $disp$ is the estimated disparity. Stereo disparity plays an important role in many areas like robotics, autonomous driving, and augmented reality as it provides an economical way to obtain the depth of the scene, compared with expensive depth sensors such as LIDAR.

The pipeline of disparity estimation usually consists of four components: feature extraction, cost computation, cost aggregation, and disparity refinement. Traditional approaches use hand-crafted features and energy minimization methods to estimate disparity. Recent methods resort to data-driven approach using Convolutional Neural Network (CNN). MC-CNN [27] uses CNN for feature extraction and cost computation, and use traditional approaches for the remaining parts. Some recent works formulate disparity estimation into an end-to-end manner. These methods can be divided into two categories: 2D CNN based [15, 17, 10, 26] and 3D CNN based [11, 2, 23, 7]. These two categories have two major differences in design. First, given extracted left and right features, 2D CNN based methods use *inner product* or *euclidean distance* for cost computation and 2D convolution are applied for further process, while 3D CNN based methods use *concatenation* operation and 3D convolution for cost computation and aggregation. Second, most 2D CNN based methods directly regress the disparity while 3D CNN based methods predict probability distribution over enumerated disparity and the final result is obtained by a weighted average operation.

While CNN based methods have achieved large improvement on disparity estimation, they usually suffer from severe over-smoothing problem at edge boundaries. While the estimated disparity map looks good, when converted to point cloud, they usually have the "long tail" effect at

*Corresponding author

boundaries. As shown in Fig. 1, 3D CNN based method PSMNet [2] fails to estimate disparity correctly at the boundaries of foreground and background regions. Note that in the point cloud plenty of points are adhering to the boundaries. Those over-smoothing estimations have a negative influence on some robotic applications like mapping, local structure inference, and path planning.

In this work, we analyze the problem for 3D CNN based methods. We observe that in most cases the over-smoothing disparity is caused by the ambiguity between the locality of estimation network and the weighted average operation. Inspired by this, we first propose a simple yet effective strategy to address this problem. Specifically, after acquiring the probability output of estimation network, we replace the original full-band weighted average with a single-modal weighted average operation. With this simple change, the over-smoothing problem is alleviated significantly.

As the proposed single-modal operation is only for post-processing, an intuitive consideration is to integrate the single-modal constraint into training stage as in [23]. To this end, we further analyze the characteristics of regression based and cross entropy based loss functions. We found that using cross entropy with gaussian distribution has more stable and fine-grained supervision signal in training stage, and it can further improve the estimation performance.

For evaluation, the commonly evaluation metric [16], *End Point Error (EPE)*, concentrates on overall performance in a point-to-point manner. In experiments, we found that it can not reflect the quality of disparity around boundaries appropriately. To this end, we propose a novel metric, *Soft Edge Error (SEE)*, in this work. *SEE* computes error only at edge regions in a point-to-patch manner and can better reflect the performance on over-smoothing problem.

Compared with currently prevalent $smooth\ \ell_1$ loss function, using cross entropy with gaussian distribution during training and single-modal weighted average operation consistently improves the performance on over-smoothing problem and overall estimation, which is validated on various datasets [15, 16, 1, 18], for various networks [2, 11, 23].

We summarize the contributions of our work as follows:

- A simple yet effective strategy is proposed to address the over-smoothing problem suffered by CNN based disparity estimation methods.
- An analysis on regression based and cross entropy based loss functions, which shows that cross entropy is more appropriate for training of disparity network.
- A novel metric is proposed for the evaluation of the quality of disparity estimation at boundary regions.
- We validate the effectiveness and general applicability of the proposed method on various public datasets using various networks.

2. Related Work

We briefly review recent works on CNN based disparity estimation.

Hybrid Method. MCCNN [27] and ContentCNN [14] utilize CNN for feature extraction traditional approaches for cost aggregation and result refinement. PBCP [19] uses CNN to predict confidence of disparity estimation. Based on the cost volume constructed using MCCNN, it fuses the confidence into SGM optimization process and achieves better accuracy. SGMNet [20] uses CNN to predict penality term in SGM [9] optimization process. For those methods that use traditional method for cost aggregation, the disparity is acquired by applying a winner-take-all strategy on cost volume. Post refinement is used for sub-pixel estimation.

2D CNN Based Method. DispNet [15] is an end-to-end network for dispariry estimation. It uses an encoder-decoder hourglass network architecture like FlowNet [5]. DispNetC [15] uses correlation of CNN features to build cost volume before feeding to hourglass network. CRL [17] proposes to use cascade residual learning to refine the disparity iteratively by stacking two hourglass network. iResNet [13] emphasizes the constraint on left-right feature constancy. Recently, DispNet3.0 [10] proposes to jointly estimate disparity, occlusion, and depth boundary in a generic network. PWCNet [22] does warping on feature instead of on image, for smaller model size and more efficient inference for optical flow estimation. Similar strategy is used for disparity estimation in UnDepthflow [24] and HD^3 [26]. SegStereo [25] exploits semantic information from joint training of semantic segmentation and disparity estimation. EdgeStereo [21] improves the performance by combining disparity estimation with edge detection network.

3D CNN Based Method. GCNet [11] first utilizes 3D convolution for end-to-end learning of disparity regression. First, like 2D CNN based methods, it uses 2D convolution for feature extraction. Then, instead of directly calculating the cost by *correlation* or *euclidean distance*, it stacks the left and right feature that corresponds to specified disparity, leading to a 4D cost volume. 3D convolution is then used for joint learning of geometry and context. Similar to the architecture of GCNet, many works have been proposed recently. PSMNet [2] applies spatial pyramid module [8] and dilated convolution [3] in feature extractor to exploit multi-scale representation and a stack hourglass 3D network for residual learning. For efficiency, StereoNet [12] proposes doing 3D convolution in lower resolution and iteratively refining the disparity by image-guided residual learning. PDSNet [23] proposes reducing channels of 3D convolution for fast inference and training by sub-pixel cross entropy loss for stable disparity range adaptation.

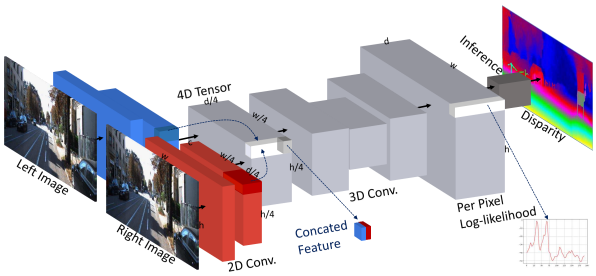


Figure 2. Typical architecture of 3D CNN based network. Note that only the d, h, w dimension of the 4D tensor is visualized.

Network	Region	The proportion of #Modal (%)			
		1	2	3	Others
Basic	All	95.70	2.70	1.01	0.59
	Edge	83.30	12.30	2.98	1.42
Shg.	All	97.30	1.49	0.48	0.73
	Edge	87.14	10.56	1.31	0.99

Table 1: #Modal statistics of result from basic and stack-hourglass(shg.) version of PSMNet [2], on Sceneflow [15].

3. Methodology

3.1. 3D CNN based Method Revisited

As shown in Fig. 2, for disparity estimation, 3D CNN based method first extracts feature map of left and right image in $\frac{1}{16}$ resolution, using a custom 2D convolution based submodule, then a 4D tensor is constructed by concatenating left and right feature on location corresponding to specified disparity value. The 4D tensor has shape of $2c \times \frac{d_{max}}{4} \times \frac{h}{4} \times \frac{w}{4}$, where c is the dimension of 2D feature map, d_{max} is the user-specified maximum diparity and (h, w) is the size of image. This 4D tensor is then fed into a 3D convlution based submodule for cost calculation and aggregation. The output of 3D submodule is a per-pixel log-likelihood for every possible disparity value, in the form of $\frac{d_{max}}{4} \times \frac{h}{4} \times \frac{w}{4}$ feature map. This low resolution feature map is then trilinear upsampled to full resolution. By applying softmax on the per-pixel log-likelihood, one can get the probability distribution $p(\cdot)$. Finally, the estimated disparity is calculated using full-band weighted average operation, as

$$\hat{d} = \sum_{d=0}^{d_{max}} d \cdot p(d) \tag{1}$$

3.2. The Over-Smoothing Problem

With weighted average operation as in Equation (1), one is able to achieve sub-pixel estimation directly. However, as shown in Fig. 1, 3D CNN based method PSMNet [2]'s disparity estimation has severe over-smoothing problem on edge boundaries. By visualizing the predicted probability distribution of boundary pixels, we found that a large part of those pixels' distribution is a multimodal distribution,

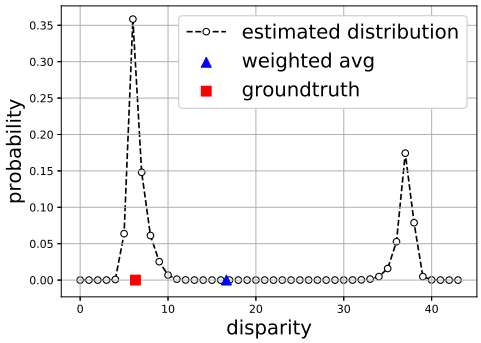


Figure 3. The multimodal distribution of pixel's disparity estimation. After applying full-band weighted average, the green triangle—the estimated disparity lies on between two modals.

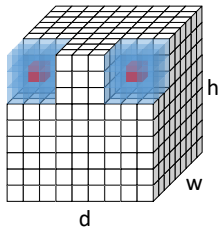


Figure 4. Visualization of $3 \times 3 \times 3$ 3D convolution operation on a single feature map with size $d \times h \times w$, which has strong locality. The connection between two red indices far apart on d dimension, which corresponds to pixel's distribution output, is too weak.

which disobeys the assumption that the full-band weighted average operation supposes. As shown in Fig. 3, the distribution of pixel's disparity presents with two modals. We further count the number of distribution's modals on Sceneflow [15] dataset for two networks presented in [2]. The statistics in Table 1 coincides with our observation. Bimodal distribution takes up a secondary proportion, which is more obvious on edge regions. After weighted average operation, the estimated disparity deviates far from the groundtruth value, and the corresponding 3D point cloud lies on between foreground and background as in Fig. 1, which is the over-smoothing problem.

From another point of view, first, as 3D CNN methods extract feature of $\frac{1}{16}$ resolution, part of spatial accuracy is lost and it's hard to determine whether pixel on edge boundary belongs to foreground or background. Second, although the 3D convolution submodule may use hourglass network to enlarge the perceptive field, as show in Fig. 4, it still can not cover the whole range of $\frac{d_{max}}{4} \times \frac{h}{4} \times \frac{w}{4}$ cost volume. This means that the output log-likelihood at location $(10, 100, 200)$ indeed has few connection with the one at location $(70, 100, 200)$, in which (x, y, z) is the coordinate of cost volume with size $d_{max} \times h \times w$. Therefore, we consider that the estimated distribution has strong locality, and the modes of distribution correspond to locations where the left and right features have strong similarity.

3.3. Single-Modal Weighted Average

Based on the assumption that the estimated distribution has strong locality, we propose single-modal weighted average operation to alleviate the over-smoothing problem. Specifically, when calculating the estimated disparity during inference, instead of using full-band weighted average operation as in Equation (1) on whole disparity range, we only apply weighted average operation on the modal with maximum probability, as

$$\hat{d} = \sum_{d=d_l}^{d_r} d \cdot \hat{p}(d) \tag{2}$$

in which d_l and d_r specify the range of modal with maximum probability. We first locate the maximum probability index, which should also be a local maximum, then march from this index to left and right respectively until it doesn't descend monotonically. $\hat{p}(\cdot)$ is normalized probability distribution, as

$$\hat{p}(d) = \begin{cases} \dfrac{p(d)}{\sum_{i=d_l}^{d_r} p(i)} & d_l \le d \le d_r \\ 0 & otherwise \end{cases} \tag{3}$$

We first locate the modal with maximum probability and the corresponding range, then normalize distribution on this range, and apply weighted average operation only on this range with the normalized distribution $\hat{p}(\cdot)$. By applying single-modal weighted average, we aim to do one more inference step on the posterior distribution of network output. This single-modal operation is only applied during inference.

3.4. Fine-Grained Supervision by Cross Entropy

Although the single-modal weighted average in Section 3.3 is able to alleviate the over-smoothing problem, it's a post-processing operation, without imposing any constraint on the output of network, which may affect the performance on other aspects. Inspired by PDSNet [23], here we analyze the characteristics of different loss function when training 3D CNN based disparity network.

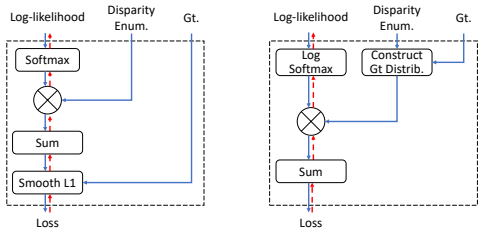
During training stage, currently, most works use smoothed regression based loss function L_{reg} defined as

$$L_{reg}(\hat{d}, d) = \begin{cases} 0.5(\hat{d} - d)^2 & |\hat{d} - d| \le 1 \\ |\hat{d} - d| - 0.5 & otherwise \end{cases} \tag{4}$$

in which $\hat{d}$ is estimated disparity and d is groundtruth value. Another is cross entropy based loss function L_{ce} as

$$L_{ce}(p_{gt}, p) = -\sum_{d=0}^{d_{max}} p_{gt}(d) \cdot \log p(d) \tag{5}$$

where $p_{gt}(\cdot)$ is a constructed groundtruth distribution. As listed in Table 1, most pixels' distribution has only one modal. Therefore, we construct Laplace and Gaussian distribution as groundtruth. The computation graph of these two types of loss function is shown in Fig. 5.



a) Regression b) Cross Entropy

Figure 5. Computation graph of two types of loss function.

As both softmax and log softmax operation do not change the relative scale of input values during forward, the main differences are the multiplication operation and the ℓ_1 loss. For an intuitive explanation, we generate a 100-dimension random vector as log-likelihood to be optimized and set the grountruth at index 30. Then we optimize this vector by using regression based loss function and cross entropy based loss function with constructed groundtruth gaussian distribution centering at target index respectively. The variation of log-likelihood, negative gradient, and probability during optimization are presented in Fig. 6.

As elaborated in Section 3.2, the modes of distribution corrrespond to locations where the left and right feature have strong similarity. Therefore, the magnitude of output log-likelihood at index corresponding to groundtruth disparity should be the largest, so as the negative gradient (i.e. the updated magnitude) during training. In Fig. 6, for regression based method, the magnitude of negative gradient at groundtruth is smaller than values in one side, which conflicts with our intuition, while for cross entropy based one, it coincides with our analysis. There is a limitation of regression based loss function. From the computation graph in Fig. 5, as regression based one uses weighted average estimation as proxy and ℓ_1 loss, the gradient back propagated is just a scalar. And as the probability is multiplied with disparity enumeration from 0 to d_{max} in weighted average operation, the gradient back propagated to log-likelihood needs to be multiplied with disparity enumeration too, causing the slanted phenomenon in Fig. 6. As cross entropy loss directly imposes constraint on the entire distribution, it is able to produce more stable and fine-grained supervision signal, which coincides with results in Fig. 6.

3.5. The Soft Edge Error

The End Point Error (EPE) metric is commonly used to evaluate the performance of disparity estimation. Given estimated disparity map $\hat{d}$ and groundtruth disparity map d, the EPE is defined as

$$E(\hat{d}, d) = \frac{1}{N} \sum_{i=1}^{N} |\hat{d}_i - d_i| \tag{6}$$

Another commonly used metric is computing the average number of erroneously estimated pixels given specific

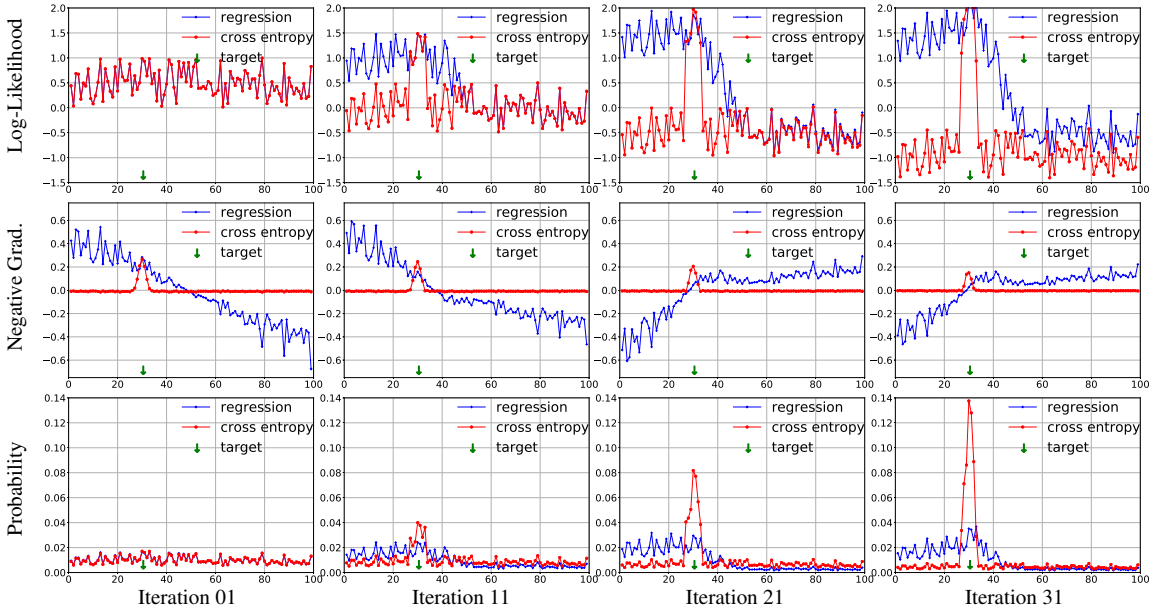


Figure 6. The variation of log-likelihood, negative gradient, and probability during optimization using regression based and cross entropy based loss function. From left to right, columns are values at iteration $n \in \{1, 11, 21, 31\}$. For all subfigures, the x-axis is disparity index.

disparity error threshold [6]. These metrics have two limitations. First, for pixels with groundtruth available, the error is calculated in a point-to-point manner, which can not reflect the over-smoothing problem in local structure. Second, these metrics count all the pixels, thus can not reflect the quality of disparity at boundaries which would be dominated by other regions. To this end, we propose the *Soft Edge Error(SEE)* metric, which only counts disparity error for pixels around edge boundaries $Edge(d)$. Formally, it's computed by

$$SEE_k(\hat{d}, d) = \frac{1}{N}\sum_{i=1}^{N} se_k(\hat{d}_i, d_i) \;\; i \in Edge(d), \qquad (7)$$

where $se_k(\cdot,\cdot)$ is *Soft Error* defined as:

$$se_k(\hat{d}_i, d_i) = \min|\hat{d}_i - d_j| \;\; j \in N_k(i). \qquad (8)$$

$N_k(i)$ denotes the local $k \times k$ neighbourhoods of point i. Note that when $k = 1$, $se_1(\cdot,\cdot)$ is the same as vanilla point-to-point error.

By definition, *Soft Error* $se_k(\hat{d}_i, d_i)$ is the minimum absolute error between the estimated disparity and it's corresponding local groundtruth patch. We match the groundtruth pixel in a patch because the disparity values at the exact boundary pixels could be uncertain. Practically, minor misalignment artifact of disparity at boundaries is acceptable as it hardly affects the local structure and over-smoothing artifact is much more undesirable. With *Soft Error*, we mean to relax the vanilla point-to-point error metric and impose more penalty on the damage of local structure. As shown in Fig. 7, for a naive 1D case, suppose the step function shown in red circle is the groundtruth, although misalignment artifact causes larger point-to-point error than

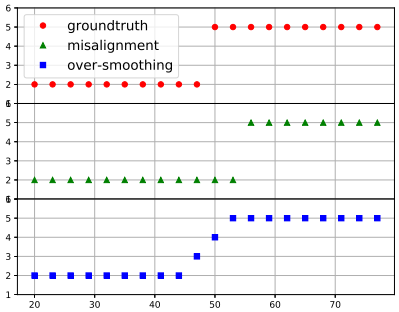


Figure 7. 1D artifact visualization.

over-smoothing artifact, it can still preserve the local structure while it's not the case for over-smoothing artifact. For over-smoothing artifact, local structure is corrupted and it has larger *Soft Error* than misalignment artifact.

$Edge(d)$ denotes the discontinuous disparity regions in the image. For datasets with dense per-pixel groundtruth such as sceneflow [15], we extract the edge boundaries by choosing pixel with absolute gradient on groundtruth disparity map exceeding a specific threshold, which is set to 2 in our benchmark. For datasets only having sparse groundtruth like KITTI 2015 [16], it's hard to determine the discontinuous disparity region precisely. We approximate it using semantic information for boundary extraction. As KITTI 2015 provides instance segmentation groundtruth, we extract the boundaries between object instances (e.g., vehicles) and background regions (e.g., wall), which are supposed to be spatially apart in the 3D world. For background regions such as road, sidewalk, and ground that are spatially connected, we treat them as one instance. The edge extracted is enlarged using dilation with 3×3 kernel.

Method	Loss	Infr.	Sceneflow [15]				Sintel [1]				Kitti 2015 [16]		Middlebury [18]	
			SEE		EPE		SEE		EPE		3SEE	3EPE	3SEE	3EPE
			Avg	3px	Avg	3px	Avg	3px	Avg	3px				
Shg [2]	L_{reg}	FB	1.57	9.40	0.89	3.13	2.82	10.80	3.17	8.86	7.82	2.00	23.97	26.36
	L_{reg}	SM	1.01	4.17	0.90	2.83	2.53	8.07	**3.16**	8.39	6.99	1.86	17.94	24.14
	L_{lap}	SM	0.99	3.20	0.89	2.35	2.32	8.13	3.27	**8.15**	6.85	1.77	14.92	22.11
	L_{gau}	SM	**0.79**	**2.53**	**0.77**	**2.21**	**2.29**	**7.33**	3.24	8.18	**6.66**	**1.70**	**11.94**	**19.05**
Basic [2]	L_{reg}	FB	1.83	10.97	1.11	4.20	3.84	12.68	5.26	11.87	8.28	2.47	27.49	33.11
	L_{reg}	SM	1.29	5.36	1.18	3.82	4.05	10.47	5.48	10.84	7.42	2.25	18.78	27.14
	L_{lap}	SM	1.16	4.14	1.13	3.28	3.45	9.25	**5.10**	10.49	8.41	2.76	16.02	24.06
	L_{gau}	SM	**1.01**	**3.60**	**1.02**	**3.12**	**3.35**	**9.12**	5.11	**10.12**	**7.04**	**2.23**	**14.32**	**21.68**
GCNet [11]	L_{reg}	FB	1.60	9.81	**0.89**	3.56	3.13	13.23	**3.18**	9.98	8.90	2.46	25.07	28.86
	L_{reg}	SM	1.27	5.39	1.24	3.51	**2.79**	**10.18**	3.37	**9.32**	**8.36**	2.65	18.30	25.10
	L_{lap}	SM	1.44	6.91	1.47	4.07	4.21	11.49	4.12	16.84	**8.36**	**2.40**	16.00	22.81
	L_{gau}	SM	**1.07**	**3.92**	1.07	**3.18**	3.94	11.12	3.95	10.06	8.71	2.50	**14.24**	**20.39**
PDSNet [23]	L_{reg}	FB	1.97	12.11	1.19	4.40	3.02	12.37	**2.60**	9.19	8.89	2.44	22.11	24.15
	L_{reg}	SM	1.32	6.01	1.20	3.97	**2.82**	9.93	2.63	8.82	8.16	2.34	17.66	22.64
	L_{lap}	SM	1.69	10.04	1.57	4.84	3.17	11.24	2.82	19.79	10.84	2.85	19.30	21.88
	L_{gau}	SM	**1.04**	**3.43**	**1.04**	**2.93**	5.14	**9.28**	2.69	**7.61**	**7.40**	**2.05**	**13.00**	**15.81**

Table 2: Evaluation of average (Avg) and 3px *SEE*, *EPE* on four datasets. Here we report SEE_5 with a tolerance of 2px edge misalignment. The loss functions are regression based L_{reg}, and cross entropy based methods with Laplace (L_{lap}) and Gaussian (L_{gau}) distribution. The inference (Infr.) methods include full band (FB) and single-modal (SM) weighted average.

4. Experiments

4.1. Implementation Detail

Datasets. For experiments, we use four datasets: Sceneflow [15], Sintel [1], KITTI 2015 [16], and Middlebury stereo [18]. Sceneflow has 35 454 training and 4370 testing images while Sintel stereo is a small dataset. Both Sceneflow and Sintel are synthesis dataset with per-pixel disparity groundtruth. KITTI 2015 is a real world city scene autonomous driving dataset with 200 training images and sparse groundtruth. Middlebury is a real-world indoor dataset with 15 training images and dense groundtruth.

Networks. We use four 3D CNN based disparity networks. Namely, the basic and stack-hourglass version of PSMNet [2], GCNet [11], and PDSNet [23]. Both PSMNet and PDSNet construct cost volume at $\frac{1}{16}$ resolution while GCNet at $\frac{1}{4}$ resolution. We trilinearly upsample the final per-pixel log-likelihood to full resolution for all networks.

Training. For all networks, we train 10 epochs on Sceneflow [15]. The learning rate(lr) is initially set to 0.001 and halved at epoch 6, 7, 8. For KITTI, we finetune with lr as 0.001 for first 200 and 0.01 for more 100 epochs. We use 160 images for finetune and 40 for validation. As Sintel is a small synthesis dataset and middlebury stereo only has 15 images, we use model trained on Sceneflow. We train all networks with regression based loss function $smooth\ \ell_1$ and cross entropy based loss function with Gaussian and Laplace groundtruth distribution. The variance of gaussian is set to 2 and the scale parameter of Laplace is set to 4. Except for loss function, the training config is set to the same for every network respectively for a fair comparison.

Evaluation Metrics. To evaluation the performance on over-smoothing problem, for synthesis datasets, we compute average and 3px *Soft Edge Error(SEE)* on Sceneflow testing split and Sintel. As 3px error is the proportion of outliers with error exceeds 3 pixels, it's more robust and practical. Therefore for real-world datasets we compute 3px SEE on KITTI validation split and Middlebury. To evaluate the overall performance we compute *End Point Error(EPE)*.

4.2. Results

The Performance Gap. From the result in Table 2, there is a large gap between *EPE* evaluated on whole image and *SEE* evaluated on discontinuous disparity regions for all methods. For stack-hourglass version of PSMNet [2], although it has 3.13 3px *EPE* on Sceneflow dataset, it's 3px *SEE* with $k = 5$, which means a tolerance of 2 pixel edge misalignmentis, is 9.40. It is larger than 3.13 by a margin that can not be overlooked. The performance on discontinuous disparity regions needs more attention.

Benefits of Single-Modal Weighted Average. From the first two rows of each block in Table 2, we can see that using single-modal weighted average is consistently better than full-band one on *SEE*. By applying single-modal weighted average operation on the probabilistic output, noteworthy improvement is achieved on discontinuous disparity regions compared with result of full-band weighted average. We further evaluate the average and 3px *SEE*, with k varying from 1 to 15, at tolerance of 0-7 pixels edge misalignment. From the results in Fig. 8, we can also see the consistent improvement from full-band to single-modal operation for various networks on various datasets.

Benefits of Cross Entropy Based Loss. As listed in Table 2 and Fig. 8, by training with cross entropy based loss function, all four networks get much lower *SEE* and *EPE*. Com-

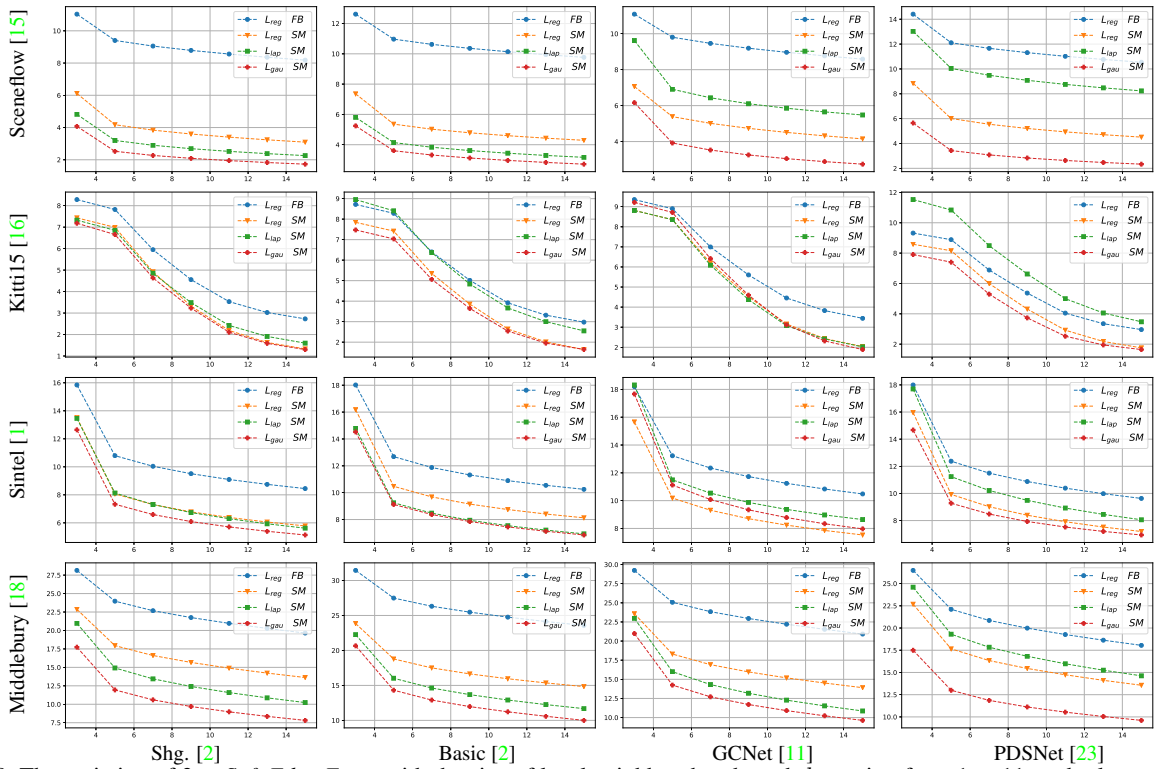


Figure 8. The variation of 3px *Soft Edge Error* with the size of local neighbourhood patch k varying from 1 to 11, and tolerance of edge misalignment from 0 to 5 pixels. For all subfigs the x-axis is k and y-axis is 3px *SEE*.

pared with regression based loss function, entropy based loss function can impose the constraint of output distribution during training and have more stable and fine-grained supervision signal during training as analyzed in Section 3.4. It is more suitable for 3D CNN based disparity estimation and achieves better performance on over-smoothing problem as well as on overall estimation. About the specified groundtruth distribution used for training, the result shows that Gaussian distribution based one is consistently more stable and better than Laplace distribution based.

The Sharpness of Gaussian Distribution. From the results in Table 2 and Fig. 8, networks training with Laplace distribution is suboptimal compared with Gaussian based. As Laplace distribution is usually used for long-tail data while Gaussian for short tails, here we study the relation between the sharpness of Gaussian distribution and the performance by controlling the variance parameter. In Fig. 9, on over-smoothing performance, we get better 3px *SEE* with lower variance. However, there is no obvious relation on overall performace. We consider that sharper groundtruth distribution helps to learn more discriminative feature, which is beneficial to challenging edge regions. However, as networks construct cost volume at low resolution, it will introduce artifacts when upsampled to full image size. Therefore, for overall performance, a sharper distribution may not be necessary.

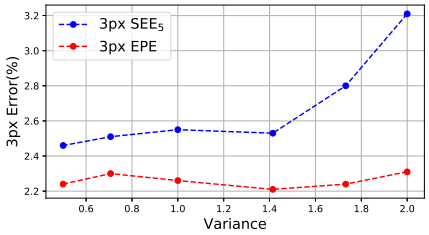


Figure 9. Performance of model trained with different variance.

Multimodal Distribution For Edge Regions. As listed in Table 1, a large proportion of pixels' probability on edge regions follows a bimodal distribution. Here we study the groundtruth distribution for edge regions. For pixels on edge regions, we use Gaussian mixture distribution for training. The modes of mixture distribution are set to edge pixels' groundtruth and groundtruth of it's neighbour which has maximum disparity difference. Table 3 lists the results training with different mixing coefficient. From 3px SEE_5 results, by choosing the mixing coefficient appropriately, one can achieve better performance on over-smoothing problem as it can better fit the real distribution. However, using Gaussian mixture distribution for training harms the overall performance from EPE results.

KITTI Benchmark. For benchmark of the proposed method's overall performance, we use the model with the best performance on validation set. Specifically, we use the

Figure 10. Point cloud converted from disparity with known intrinsic using MeshLab [4]. From top to bottom: the left input image, result from PSMNet [2], result from ours, and groundtruth. Please zoom in for more details.

Coefficient		SEE_5		EPE	
Gt.	Ngb.	Avg	3px	Avg	3px
1.0	0.0	**0.79**	2.53	**0.77**	**2.21**
0.9	0.1	0.90	2.65	0.96	2.63
0.8	0.2	0.92	2.56	1.10	2.79
0.7	0.3	0.82	2.19	0.94	2.52
0.6	0.4	0.83	**2.15**	0.98	2.69

Table 3: Results of models trained with different mixing coefficient on groundtruth(Gt) and neighbour(Ngb.) disparity.

	All (%)			Noc (%)		
Method	Bg	Fg	All	Bg	Fg	All
PDSNet [23]	2.29	4.05	2.58	2.09	3.68	2.36
PSMNet [2]	1.86	4.62	2.32	1.71	4.31	2.14
SegStereo [25]	1.88	4.07	2.25	1.76	3.70	2.08
EdgeStereo [21]	1.87	**3.61**	2.16	1.72	**3.41**	2.00
Ours	**1.54**	4.33	**2.14**	**1.70**	3.90	**1.93**

Table 4: Results on KITTI 2015 [16] benchmark over nonoccluded(Noc) and overall(All) regions.

	>2px (%)		>3px (%)		>4px (%)		>5px (%)	
Method	Noc	All	Noc	All	Noc	All	Noc	All
PDSNet [23]	3.82	4.65	1.92	2.53	1.38	1.85	1.12	1.51
SegStereo [25]	2.66	3.19	1.68	2.03	1.25	1.52	1.00	1.21
PSMNet [2]	2.44	3.01	1.49	1.89	1.12	1.42	0.90	1.15
EdgeStereo [21]	2.32	2.88	1.46	1.83	1.07	**1.34**	**0.83**	**1.04**
Ours	**2.17**	**2.81**	**1.35**	**1.81**	**1.04**	1.39	0.87	1.16

Table 5: Results on KITTI 2012 [6] benchmark.

stack hourglass version of PSMNet [2] trained using cross entropy based loss function with Gaussian distribution and applying single-modal weighted average during inference. The results on KITTI 2012 [6] and 2015 [16] testing sets are listed in Table 4 and 5. As listed, compared with the result of original PSMNet [2], by applying the proposed method, we achieve consistent non-trivial improvement, which even surpasses strong baselines [21, 25] that use semantic information of edge or segmentation.

Qualitative Results. We show some qualitative results in Fig. 10 and 1, where we visualize the 3D point cloud converted from estimated disparity. Our method is able to estimate more sharp disparity on discontinuous regions, with few over-smoothing estimations, while PSMNet [2] fails to estimate disparity at the boundaries of foreground (e.g. vehicles) and background (e.g. wall) regions.

5. Conclusions

In this work, we aim at addressing the over-smoothing problem of CNN based disparity estimation, which is unfavorable by many practical applications but seldom explored in previous work. For 3D CNN based methods, after analyzing the probability distributions of disparity, we propose a simple yet effective method to alleviate this problem. We then analyze the characteristics of different loss function and found cross entropy based one is more appropriate for 3D CNN based disparity estimation. By integrating the singe-modal constraint into training stage, further improvements are achieved, both on the over-smoothing problem and overall performance. As existing disparity metric can not reflect the error at local boundary structure, we propose a novel metric, *Soft Edge Error*, for evaluation. Experiments on various public datasets using various networks clearly validate the effectiveness and general applicability of the proposed method, which significantly reduces the over-smoothing effect and improves the overall performance. We hope this work could inspire further research in this direction.

Acknowledgment. This work is supported by Major Program of Science and Technology Planning Project of Guangdong Province (2017B010116003), and NSFC-Shenzhen Robotics Projects (U1613211).

References

[1] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *European Conf. on Computer Vision (ECCV)*, pages 611–625. Springer-Verlag, 2012. 2, 6, 7

[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. 1, 2, 3, 6, 7, 8

[3] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *The European Conference on Computer Vision (ECCV)*, September 2018. 2

[4] Paolo Cignoni, Marco Callieri, Massimiliano Corsini, Matteo Dellepiane, Fabio Ganovelli, and Guido Ranzuglia. MeshLab: an Open-Source Mesh Processing Tool. In Vittorio Scarano, Rosario De Chiara, and Ugo Erra, editors, *Eurographics Italian Chapter Conference*. The Eurographics Association, 2008. 8

[5] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2758–2766, 2015. 2

[6] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361, 2012. 5, 8

[7] Xiaoyang Guo, Kai Yang, Wukui Yang, Xiaogang Wang, and Hongsheng Li. Group-wise correlation stereo network. In *CVPR*, 2019. 1

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *European conference on computer vision*, pages 346–361. Springer, 2014. 2

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

[10] Eddy Ilg, Tonmoy Saikia, Margret Keuper, and Thomas Brox. Occlusions, motion and depth boundaries with a generic network for disparity, optical flow or scene flow estimation. In *The European Conference on Computer Vision (ECCV)*, September 2018. 1, 2

[11] 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. 1, 2, 6, 7

[12] 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 *The European Conference on Computer Vision (ECCV)*, September 2018. 2

[13] Zhengfa Liang, Yiliu Feng, Yulan Guo, Hengzhu Liu, Wei Chen, Linbo Qiao, Li Zhou, and Jianfeng Zhang. Learning for disparity estimation through feature constancy. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[14] Wenjie Luo, Alexander G. Schwing, and Raquel Urtasun. Efficient deep learning for stereo matching. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2

[15] 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. 1, 2, 3, 5, 6, 7

[16] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2, 5, 6, 7, 8

[17] 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. 1, 2

[18] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *German conference on pattern recognition*, pages 31–42. Springer, 2014. 2, 6, 7

[19] Akihito Seki and Marc Pollefeys. Patch based confidence prediction for dense disparity map. In *BMVC*, volume 2, page 4, 2016. 2

[20] Akihito Seki and Marc Pollefeys. Sgm-nets: Semi-global matching with neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[21] Xiao Song, Xu Zhao, Hanwen Hu, and Liangji Fang. Edgestereo: A context integrated residual pyramid network for stereo matching. *Asian Conference on Computer Vision*, 2018. 2, 8

[22] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[23] S. Tulyakov, A. Ivanov, and F. Fleuret. Practical Deep Stereo (PDS): Toward applications-friendly deep stereo matching. In *Proceedings of the international conference on Neural Information Processing Systems (NIPS)*, 2018. 1, 2, 4, 6, 7, 8

[24] Yang Wang, Zhenheng Yang, Peng Wang, Yi Yang, Chenxu Luo, and Wei Xu. Joint unsupervised learning of optical flow and depth by watching stereo videos. *arXiv preprint arXiv:1810.03654*, 2018. 2

[25] Guorun Yang, Hengshuang Zhao, Jianping Shi, Zhidong Deng, and Jiaya Jia. Segstereo: Exploiting semantic information for disparity estimation. In *The European Conference on Computer Vision (ECCV)*, September 2018. 2, 8

[26] Zhichao Yin, Trevor Darrell, and Fisher Yu. Hierarchical discrete distribution decomposition for match density estimation. In *CVPR*, 2019. 1, 2

[27] Jure Zbontar and Yann LeCun. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*, 17:1–32, 2016. 1, 2