

DrivingStereo: A Large-Scale Dataset for Stereo Matching in Autonomous Driving Scenarios

Guorun Yang^{1*} Xiao Song^{2*} Chaoqin Huang^{2,3} Zhidong Deng¹ Jianping Shi² Bolei Zhou⁴

¹Department of Computer Science and Technology, Tsinghua University[†]

²SenseTime Group Limited ³Shanghai Jiao Tong University ⁴The Chinese University of Hong Kong

{ygr13@mails, michael@mail}.tsinghua.edu.cn huangchaoqin@sjtu.edu.cn

{songxiao, shijianping}@sensetime.com bzhou@ie.cuhk.edu.hk

Abstract

Great progress has been made on estimating disparity maps from stereo images. However, with the limited stereo data available in the existing datasets and unstable ranging precision of current stereo methods, industry-level stereo matching in autonomous driving remains challenging. In this paper, we construct a novel large-scale stereo dataset named *DrivingStereo*. It contains over 180k images covering a diverse set of driving scenarios, which is hundreds of times larger than the *KITTI Stereo* dataset. High-quality labels of disparity are produced by a model-guided filtering strategy from multi-frame LiDAR points. For better evaluations, we present two new metrics for stereo matching in the driving scenes, i.e. a distance-aware metric and a semantic-aware metric. Extensive experiments show that compared with the models trained on *FlyingThings3D* or *Cityscapes*, the models trained on our *DrivingStereo* achieve higher generalization accuracy in real-world driving scenes, while the proposed metrics better evaluate the stereo methods on all-range distances and across different classes. Our dataset and code are available at <https://drivingstereo-dataset.github.io>.

1. Introduction

Depth estimation is one of the critical topics in autonomous driving perception. Though LiDAR is commonly used for the accurate depth and surrounding sensing, the LiDAR point clouds acquired are usually sparse especially for the objects in far range while the device itself is high-priced. Alternatively, the calibrated stereo camera system

is able to provide the affordable dense depth estimation. Recently deep learning methods greatly improve the performance of stereo matching, bringing in better and more accurate depth estimation. As a result, on KITTI Stereo 2015 leaderboard [21], the bad pixels rate (D1 error) reduces from 6.38% [13] to 1.74% [6], which shows the rapid progress of stereo matching in the driving scenes.

Though deep learning methods of stereo matching on existing driving benchmarks achieve good performance, they still suffer from unreliable adaptation and insufficient precision when applied to the real-world driving scenarios. The main issue lies in **data** and **metric**.

Training deep learning models requires massive labeled data. The existing driving datasets only comprise hundreds of images [11, 21], on which deep neural networks are prone to overfit. Moreover, the large-scale synthetic stereo datasets [20, 10] cannot reflect the real-world data distribution, so that the trained model is difficult to generalize. Thus a large-scale stereo dataset containing a diverse set of road scenarios is demanding. On the other hand, we find that the performance metrics for stereo matching, such as the end-point error (EPE) [2] and D1 error [21], are insufficient to evaluate capabilities of stereo matching algorithms in complex driving scenes. In our statistics, more than 80% valid pixels locate on short-range distances or background regions, which heavily affects the final measures of EPE or D1 error. In real-world driving, the different-distance regions as well as various moving objects such as the vehicles and pedestrian, should also be considered. As a result, new metrics tailored for driving scenarios are needed.

To address the issue of data and metric mentioned above, we construct a large-scale stereo matching dataset *DrivingStereo* for driving scenes. We build the data acquisition platform and collect over 180K frames with LiDAR point clouds. As shown in Fig. 1, our dataset contains a diverse range of driving scenarios, including urban, suburban, highway, elevated, and country roads, together

* indicates equal contribution.

[†] State Key Laboratory of Intelligent Technology and Systems, Beijing National Research Center for Information Science and Technology, and Center for Intelligent Connected Vehicles and Transportation.

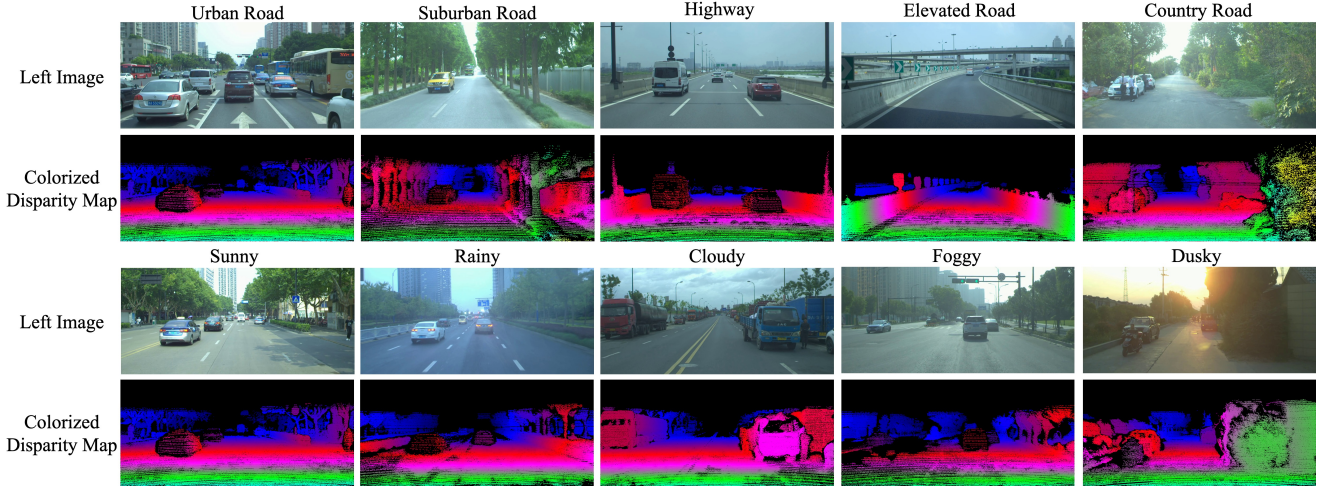


Figure 1. Examples of our DrivingStereo dataset. Our dataset contains a diverse range of driving scenarios. The image of the left camera and the disparity map are shown.

with scenarios under different climates like sunny, rainy, cloudy, foggy, and dusky weathers. Compared to existing synthetic [10, 20, 3] and real-world datasets [11, 21, 7], our DrivingStereo has many advantages such as real-world scenes, large quantity, high diversity, and high-quality disparity labels. A model-guided strategy is proposed to filter original disparity maps directly projected from multi-frame fused LiDAR point clouds.

We further define the distance-aware and semantic-aware metrics for the performance evaluation of stereo matching algorithms. The distance-aware metric is designed to measure the error between the disparity prediction and the ground-truth label on all ranges. In contrast to the overall metrics such as EPE and D1 error, we draw the distance-aware curve to distinguish the deviation of estimated disparity across all possible depths. Similarly, the semantic-aware metric is devised to compute the matching rate on various semantic categories, such as vehicle and human concerned in driving, via a semantic-aware radar map. Based on these two metrics, different properties of current stereo matching methods can be better uncovered in driving scenarios.

We benchmark stereo matching baselines, including eight state-of-the-art deep learning-based stereo matching models [20, 15, 23, 4, 17, 32, 28] and the classical semi-global matching (SGM) algorithm [12], on our DrivingStereo. In order to demonstrate the generalization ability of our dataset and metrics, deep stereo models are trained on FlyingThings3D [20], Cityscapes [7], and our DrivingStereo dataset respectively then compared. Results on KITTI [21] and our dataset demonstrates that our DrivingStereo makes stereo models more generalizable to real-world driving scenes. Rather than the previous metrics EPE or D1 error, our metrics reveal the perceptive deviation on all-range distances and the matching accuracy on specific objects for those stereo methods. Based on our dataset and

metrics, we further compare the stereo perception with LiDAR measurements [16] on different distances or objects so as to quantify their gap for future improvement.

We summarize the main contributions below:

- We construct a large-scale stereo dataset that comprises over 180K images covering a diverse range of driving scenarios, along with a model-guided filtering strategy for producing high-quality disparity labels.
- We define distance-aware and semantic-aware metrics that are suitable for stereo matching evaluation on farther ranges and various classes.
- Experimental results across different datasets and stereo methods demonstrate the capacity of our dataset and the effectiveness of the proposed metrics.

2. Related Work

Stereo Datasets. With the rapid development of deep learning, a couple of datasets emerge for autonomous driving [11, 21, 33, 14, 7, 22, 19]. A few of these datasets get involved in stereo collections [11, 21, 7, 14, 19]. KITTI stereo 2012 [11] and 2015 [21] release hundreds of image pairs, where the disparity labels are transformed from Velodyne LiDAR points. The main problem of KITTI stereo datasets is the small quantity, which is insufficient for training deep learning models. Cityscapes dataset [7] provides a considerable amount of stereo images, while the disparity labels are pre-computed by SGM method [13]. On the other hand, various synthetic stereo datasets are set up based on graphic techniques. The Scene Flow [20], virtual KITTI [10], and Sintel [3] datasets synthesize dense disparity maps, but it remains a huge gap between the synthetic domain and the real domain. In addition, Middlebury stereo dataset [25] and ETH 3D dataset [27] are not made for driving scenario.

Above all, it is demanding to have a large stereo dataset with high-quality disparity labels for driving scenarios.

In our DrivingStereo, all of stereo images are collected in real-world driving scenarios. The disparity labels are projected from multi-frame LiDAR point clouds. To reduce potential errors caused by moving objects or calibration deviation, we present a model-guided strategy to filter original disparity maps. Similarly, the KITTI Depth dataset [30] also employs SGM [13] to make choice of correct pixels. In our method, the filtering is conducted by a guided model which further ensures the quality of final disparity labels.

Stereo Matching Methods. Stereo matching problems have been studied for several decades. Scharstein *et al.* [26] summarize typical stereo algorithms as a four-step pipeline. SGM [12] is a classical algorithm that follows the pipeline. Zbontar and LeCun [34] first introduce CNN to describe image patches for stereo matching. Inspired by FCN used in semantic segmentation [18, 5], Mayer *et al.* [20] raise an encoder-decoder architecture called DispNet to enable end-to-end disparity regression. DispNet adopts a correlation operation as FlowNet [8] where the matching cost can be directly integrated to encoder volumes. Pang *et al.* [23] provide a cascade structure to optimize residues between prediction results and ground-truths. Liang *et al.* [17] also propose a two-stage pipeline to refine initially estimated disparity. A few methods introduce 3D convolution kernels to learn disparity. For example, Kendall *et al.* [15] integrate contextual information by 3D convolutions over a cost volume. Chang *et al.* [4] combine spatial pyramid network with 3D convolutional layers to incorporate global context. Cheng *et al.* [6] extend their convolutional spatial propagation network to 3D space for correlation. Lately, several methods attempt to combine other information to enhance stereo matching. In SegStereo [32], semantic cues are employed to improve stereo matching. EdgeStereo [28, 29] extracts edge cues to refine disparity estimation.

In this work, we evaluate several popular stereo matching methods on our dataset. Compared to existing models that are trained on FlyingThings3D [20] or Cityscapes [7], the models pretrained on our DrivingStereo outperform the others on our test set and KITTI Stereo dataset [21], which demonstrates the capacity of our dataset. Furthermore, all of the methods are evaluated on our proposed metrics. Such new metrics uncover more detailed results of stereo matching under all ranges and at different semantic classes, bringing insights to future study on stereo matching.

3. Dataset Construction

In this section, we first introduce the data acquisition system, where stereo images and LiDAR point clouds are simultaneously collected. Then a model-guided filtering strategy is presented to generate high-quality disparity labels from multi-frame point clouds.

3.1. Data Acquisition

Similar to the KITTI [11], we install several color cameras (Basler ACA1920-40GC), a 3D laser scanner (Velodyne HDL-64E S3) and a GPS/IMU navigation system (OXTS RT3003G) on our SUV. We select two cameras as stereo pairs, one is mounted on the top center, and the other is mounted on the top right. The baseline distance between such stereo pair is 54cm, and the field of view (FOV) is 50°. The LiDAR is also equipped behind the center camera, and the navigation unit is at the rear. For spatial alignment, the intrinsic and extrinsic parameters of cameras are at first calculated to rectify stereo images. Then we jointly make calibration of the cameras with LiDAR and GPS/IMU. More details of calibration are provided in the supplementary material. For time synchronization, all of the sensors adopt GPS clocks. When the LiDAR turns to the front view, the stereo cameras are triggered, and the corresponding vehicle pose is also saved. Limited by the LiDAR, the frequency of the collection system is set to 10Hz.

Totally, we collect 42 sequences with totally more than 1 million frames, which covers a wide range of driving scenarios. Some of the examples are shown in Fig. 1. Among the raw frames, we choose 18, 2188 frames to construct our DrivingStereo dataset, where 174, 437 frames from 38 sequences are used as training dataset. The remaining 7, 751 frames from 4 sequences are manually selected as testing dataset that has a higher quality of disparity labels. For the testing frames, we produce their pixel-level semantic labels by PSP-Net [35] trained from Cityscapes [7] with some post-processing. To our best knowledge, the DrivingStereo is the largest stereo dataset for the real-world driving scenario so far.

Apparently, the LiDAR point clouds acquired are sparse, particularly in the farther place. When single-frame points are projected onto an image plane, only a few pixels in the image have valid values. In order to increase the number of valid pixels in a single frame, we integrate point clouds from adjacent frames and fuse them together as shown in Fig. 3. The data fusion can be written as:

$$C_k^f = \sum_{i=-m}^n C_{(k+i)}^s \times T_{(k+i) \rightarrow k}, \quad (1)$$

where C_k^f and C_k^s denote the fused cloud and single cloud on frame k , respectively. m and n indicate the backward and forward seeking frames. $T_{(k+i) \rightarrow k}$ represents the mapping matrix from frame $k+i$ to k . We add a voxel grid [24] to reduce redundant points in fused cloud C^f . According to calibration matrices, each LiDAR point $p^l = [x, y, z, 1]^T$ in C^f is projected to camera pixel $p^c = [u, v, 1]^T$:

$$p^c = P_{rect} \times R_{rect} \times (R|T)_{l \rightarrow c} \times p^l, \quad (2)$$

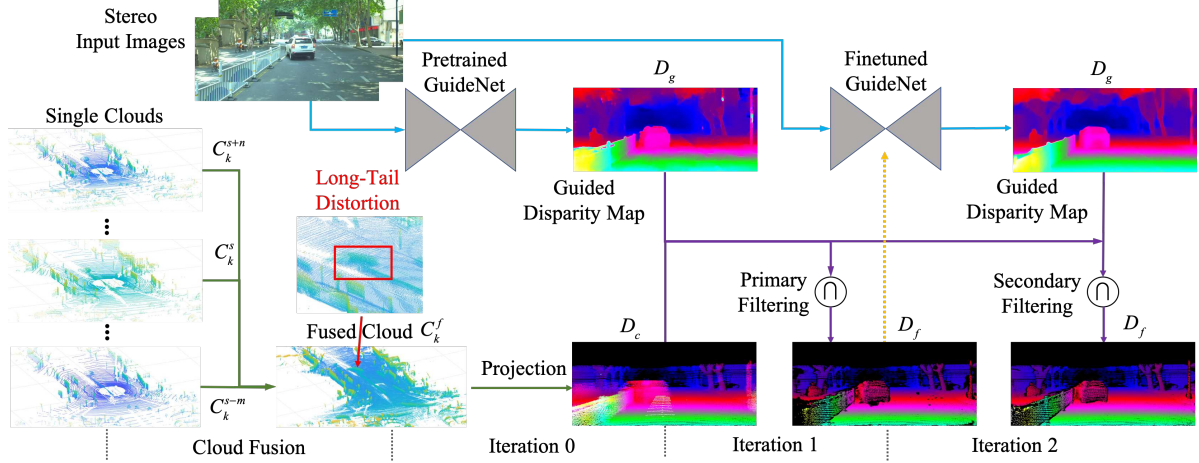


Figure 2. Diagram of model-guided filtering. From left to right, we perform cloud fusion, cloud projection, primary and secondary filtering. The GuideNet are utilized to output guided disparity map for filtering.

where the P_{rect} is the 3×4 intrinsic matrix of left referenced camera, and the R_{rect} is the 4×4 calibrated matrix between stereo cameras. The $(R|T)_{tr}$ is extrinsic matrix transformed from the Velodyne coordinate system to the image plane of the camera. For the problem of overlapping, we select the nearest LiDAR point for projection. After the cloud fusion and the point projection, we obtain original disparity map denoted as D_c . Although the fusion of multiple frames increases the number of valid pixels, more errors caused by dynamic objects are introduced in the D_c , which leads to the familiar problem of long-tail distortion as shown in Fig. 3. Furthermore, several inevitable factors, such as calibration error and accumulative deviation, also lower the quality of fused clouds. Hence, we present a guided filtering strategy to tackle the problem.

3.2. Model-Guided Filtering

The goal of guided filtering is to generate high-quality disparity labels from the fused cloud. We design a light-weight model named GuideNet for guided filtering. Following the encoder-decoder architecture like that in [18, 8, 20], the GuideNet employs 16 residual blocks as the encoder and 3 deconvolutional layers as the decoder. A correlation module [8] is embedded into the encoder to compute a matching cost between stereo features. We shrink the channel number in the convolutional layers to reduce model size. The inference time of our GuideNet is 20ms, which is about 3 times faster than DispNet [20]. The GuideNet is pretrained on the FlyingThings3D dataset [20] to predict a reasonable disparity map for initial filtering. Training implementations are described in Sec. 5.2, and the layer-by-layer definition is provided in supplementary material. Based on the disparity map predicted by GuideNet, we define the filtering rule

expressed as:

$$D_f = D_c \cap D_g, \quad (3)$$

$$d_c \cap d_g = \begin{cases} 1 & |d_c - d_g| \leq \delta_d \\ 0 & |d_c - d_g| > \delta_d \end{cases},$$

where D_c indicates the disparity map projected from fused clouds C^f , D_g denotes the predicted disparity map from the GuideNet. Since D_c is sparse and D_g is dense, the \cap operation is defined as each disparity value d_c on D_c is validated by d_g on D_g , and only consistent disparity whose deviation is less δ_d is retained on D_f . After primary filtering, the disparity maps are used to finetune the GuideNet, which makes the model better suitable for the driving case. In other words, the finetuned GuideNet can make the prediction of more accurate disparity maps, which are used to conduct secondary filtering for final disparity labels.

Instead of manual adjustment [21], our guided filtering enables the automatic generation of high-quality labels, which is the foundation for large-scale dataset construction. Meanwhile, the GuideNet is well-trained when making DrivingStereo. Experimental result in 5.3 shows that our light-weight GuideNet achieves similar accuracy to other stereo methods.

4. Evaluation Metrics

In driving perception, we should consider objects in different ranges and classes. For example, reliable detection of the objects in long-range distances benefits our velocity planning, while the detection of moving foreground objects plays a more important role in obstacle avoidance. First, we provide a statistical analysis of depth information and semantic distribution. We then define new distance-aware and semantic-aware metrics that can better recognize the accuracy on all-range distances and diverse classes.

4.1. Statistical Analysis

In a normal road image, most of pixels locate on nearby objects or background surroundings, such as roadway, building, and vegetation. Here, we conduct quantitative statistics of depth distribution on KITTI Stereo 2015 [21, 1]. We transform the disparity maps to depth images and compute the proportion of pixels in each depth interval. We find that more than 80% valid pixels are less than $40m$. When computing the overall EPE or the D1 error, the close regions or objects largely determine the final results. While for autonomous driving, the metrics need to strike a balance among short, middle and long distances. Similarly, the proportion of pixels distributed in different semantic categories are: ground(53.62%), nature(23.05%), construction(3.43%), vehicle(17.63%), human(0.04%) and others(2.23%). The pixels on ground and nature play a majority to the final results. During actual driving, we should pay more attention to the foreground objects, especially the moving vehicles and pedestrians. Therefore, we define the following metrics to evaluate comprehensive performance across different ranges and classes.

4.2. Distance-aware Metrics

To capture the depth information in a farther range and prevent the disturbance from sparse LiDAR-transformed label, the distance-aware metrics are defined on the whole distances. Based on the focal length and the stereo baseline, the ground-truth disparity map D_g is transformed back to depth map D_g^{-1} . We sample multiple points K with a certain of interval on depth axis. For each sampled depth point k , we indicate its measurement range R_k as $[k - r, k + r]$. The pixels of D_g^{-1} within the range R_k are accumulated. Then we compute the absolute relative difference (ARD) between the disparity estimation d_p and the ground-truth d_g in such range R_k :

$$ARD_k = \frac{1}{N_{R_k}} \sum_{d_g \in R_k} \frac{|d_p - d_g|}{d_g}, \quad (4)$$

where N_{R_k} is the number of valid pixels in R_k . We can draw the ARD curve by linking the ARD_k . Moreover, the global difference (GD) as the summation of single ARDs:

$$GD = \frac{1}{K} \sum_{k \in K} ARD_k. \quad (5)$$

From the ARD curve, we observe the change of predicted error with the increase of distance. Compared to EPE and D1 error, the GD is able to overcome the imbalance among different distance. In Sec. 5.3, we will draw the ARD curves of deep stereo methods on the KITTI dataset [21] and exploit the properties of various methods.

4.3. Semantic-aware Metrics

The semantic-aware metrics are presented to evaluate the stereo matching performance on different objects in driving scenes, especially for vehicle and pedestrian. We merge the classes from the Cityscapes semantic segmentation [7] into 6 categories, including ground, nature, construction, vehicle, human, and others. Inspired from [9], we define the matching rate (MR) for each category k :

$$MR_k : \% \text{ of } d_p \text{ s.t. } \max\left(\frac{d_p}{d_g}, \frac{d_g}{d_p}\right) < \theta, \quad (6)$$

where θ is a threshold to identify whether the disparity prediction d_p matches d_g . We draw the radar map of matching rate for each category in Sec. 5.3. In addition to background regions, we can observe the accuracy of learning disparity on foreground objects, which facilitates the analysis of current stereo methods.

5. Experiments

In this section, we first provide an overview of our dataset and compare it with existing stereo datasets. We then introduce the implementations of our GuideNet and validate the filtering strategy on KITTI Stereo dataset [21]. Finally, we provide baseline results of several stereo methods on our dataset and evaluate these methods using the proposed metrics. Comparative results illustrate the capacity of our dataset and metrics.

5.1. Dataset Overview

We first specify the key parameters of collecting stereo data. When we make fusion of multi-frame point clouds, the forward-searching frames n and the backward-searching frames m are set to 20 and 5, respectively. The stereo images are well rectified, and the resolution is cropped to 1762×800 . For semantic labels, we produce initial segmentation results using PSP-Net [35], then use a common 2D detector for refinement on vehicles and humans. The examples of semantic segmentation are shown in supplementary material.

Table 1. Comparison of available stereo datasets.

Dataset	Scenario	Frames		Resolution	Labels
		Training	Testing		
Virtual KITTI [10]	Virtual	21,260	—	1242×375	✓
FlyingThings3D [20]	Virtual	21,818	4,248	960×540	✓
MPI Sintel [3]	Virtual	1,064	564	1024×436	✓
Middlebury [25]	Indoor	15	15	720×480	✓
ETH 3D [27]	Outdoor	27	20	940×490	✓
Cityscapes [7]	Driving	2,975	500	2048×1024	✗
RobotCar [19]	Driving	—	—	1280×960	✗
Raw KITTI [11]	Driving	—	—	1226×370	✗
KITTI 2012 [11]	Driving	194	195	1226×370	✓
KITTI 2015 [21]	Driving	200	200	1242×375	✓
DrivingStereo(Ours)	Driving	174,437	7,751	1762×800	✓

In Tab. 1, we list the parameters of previous stereo datasets. Our dataset has the following advantages: 1) **Real-world scenes**. Unlike synthetic datasets [10, 20, 3], we focus on real-world driving scenarios and build the acquisition platform to collect data. 2) **High diversity**. As shown in Fig. 1, our dataset provides diverse scenarios, which is possible to cover most conditions in autonomous driving. 3) **Substantial size**. The total frames of our data exceed 180k that are much larger than other real-world datasets [25, 27, 11, 21], even more than those synthetic datasets [10, 20, 3]. 4) **High-quality labels**. Different from Cityscapes [7], the disparity labels are projected from LiDAR points and filtered by model-guided strategy as well. We also validate the effectiveness of guided filtering on KITTI in next sub-section.

5.2. Validation of Model-Guided Filtering

When conducting guided filtering, the GuideNet is successively trained on FlyingThings3D dataset [20] and fine-tuned on the primary DrivingStereo. We implement the GuideNet in PyTorch. In the phase of training, we use the “poly” learning rate policy. We set the base learning rate to 0.01, the power to 0.9, the momentum to 0.9 and the weight decay to 0.0001 respectively. The maximum iteration and batch size are set to 300K and 16. For spatial augmentation, we adopt random resize and crop. The random factor is between 0.5 to 2.0. The “crop size” is set to 513×321 . For color augmentation, we use the adjustments of color shift and contrast brightness. The maximum color shift along RGB axes is set to 20, and the maximum brightness shift is set to 5. The multiplier of contrast brightness is between 0.8 to 1.2. We adopt L1 loss to train the model. The threshold δ_d in Eq. 3 is set to 2 for guided filtering. We also provide the layer-by-layer definition of GuideNet and the ablation study on “poly” learning policy in supplementary material.

Table 2. Results of model-guided filtering on KITTI Stereo 2015 [21] and our dataset.

(a) KITTI Stereo 2015 [21]			
All	Valid Pixels	Correct Pixels	Accuracy
<i>single frame</i>	1,586,714	1,548,043	97.56%
<i>iter₀</i>	4,948,724	3,607,359	72.89%
<i>iter₁</i>	2,445,925	2,374,226	97.07%
<i>iter₂</i>	2,973,882	2,915,110	98.02%
Obj	Valid Pixels	Correct Pixels	Accuracy
<i>single frame</i>	127,579	102,728	80.54%
<i>iter₀</i>	1,005,084	283,011	28.13%
<i>iter₁</i>	219,319	200,559	91.45%
<i>iter₂</i>	221,828	210,912	95.07%

(b) Our DrivingStereo dataset				
Iteration	<i>single frame</i>	<i>iter₀</i>	<i>iter₁</i>	<i>iter₂</i>
Valid Pixels	8,250	216,025	61,555	105,800

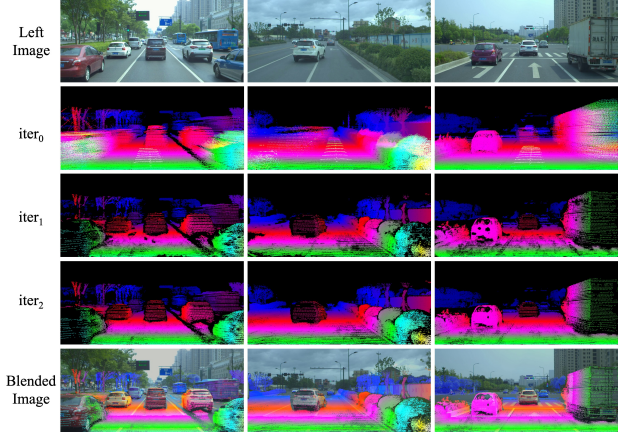


Figure 3. Model-guided filtering examples.

Since our dataset originally does not have ground-truths, we validate the guided filtering strategy on KITTI dataset [21]. The stereo 2015 dataset [21] releases 200 pairs of disparity labels along with their foreground masks, in which 142 images have origin index in raw sequences [11]. Here, we select these 142 images as a validation set and sample other 8,260 frames from the raw dataset as a training set. Similar to the preparation on DrivingStereo, we conduct cloud fusion, guided filtering as Sec. 3.2. In Tab. 2(a), we measure the valid pixels, correct pixels, and accuracy at different iterations. Compared with the disparity map from a single-frame cloud, the projected disparity map from fused clouds at *iter₀* has a large number of valid pixels, but its accuracy is only 72.89% on all areas and 28.13% on foreground regions. After the primary and secondary filtering, the accuracy is improved to 98.02% on all areas and 95.07% on foreground regions. In addition, the numbers of valid pixels and correct pixels also increase by secondary filtering. It is because the finetuned model is more adaptable to the road scene and predict better results. Thus, the validation results on KITTI illustrate the usability of the guided filtering for improving the labeling accuracy.

After the validation on KITTI [21] is done, we adopt the guided filtering to generate disparity labels for our dataset. In Tab. 2(b), the average amount of valid pixels is increased from 8,250 to 216,025 by multi-frame fusion, and from 61,555 to 105,800 by secondary filtering. Fig. 3 shows several examples of our guided filtering. It can be found that the artifacts from moving objects in the fused cloud and the holes located on roads in primary filtering are mostly optimized in secondary filtering.

5.3. Baseline Results

In order to demonstrate the capacity of our dataset, we use three dataset including FlyingThings3D [20], Cityscapes [7], and our DrivingStereo to train deep stereo models. Here, we employ SGM [12], DispNet [20], CRL [23], iResNet [17], PSMNet [4], SegStereo [32],

Table 3. Baseline results on KITTI Stereo 2015 and our DrivingStereo test set. The deep stereo methods are trained on FlyingThings3D [20], Cityscapes [7], and our DrivingStereo, respectively. Among the metrics, “GD” is the global difference of our distance-aware metric, and we use percentage mark to identify “GD”. For the running time, [G], [C], [P] denote the platforms of GPU, Caffe, and PyTorch.

(a) Evaluation on KITTI Stereo 2015 [21]

Methods	Running Time	Pretrained on FlyingThings3D [20]			Pretrained on Cityscapes [7]			Pretrained on DrivingStereo (Ours)		
		EPE	D1 Error	GD	EPE	D1 Error	GD	EPE	D1 Error	GD
SGM [12]	850ms [G]	1.82	8.28%	13.52%	1.82	8.28%	13.52%	1.82	8.28%	13.52%
DispNet [20]	40ms [C]	3.05	21.50%	14.59%	1.91	14.66%	17.77%	1.24	6.30%	14.04%
CRL [23]	152ms [C]	3.22	28.63%	14.57%	1.26	6.42%	13.45%	1.18	5.34%	10.69%
PSMNet [4]	456ms [P]	—	—	—	1.20	6.05%	10.18%	1.14	5.44%	11.26%
iResNet [17]	122ms [C]	2.98	21.64%	13.52%	1.38	7.51%	14.37%	1.20	5.52%	8.96%
SegStereo [32]	225ms [C]	2.07	12.43%	24.54%	1.43	7.93%	14.07%	1.13	5.31%	12.04%
ResNetCorr [32]	170ms [C]	2.42	15.14%	28.95%	1.46	8.24%	14.13%	1.28	6.50%	14.39%
EdgeStereo [28]	320ms [C]	2.41	12.91%	17.93%	1.27	6.99%	12.56%	1.12	5.35%	10.64%
GuideNet	12ms [P]	2.36	18.02%	22.52%	1.51	9.22%	14.82%	1.36	7.33%	18.44%

(b) Evaluation on DrivingStereo test set

Methods	Running Time	Pretrained on FlyingThings3D [20]			Pretrained on Cityscapes [7]			Pretrained on DrivingStereo (Ours)		
		EPE	D1 Error	GD	EPE	D1 Error	GD	EPE	D1 Error	GD
SGM [12]	850ms [G]	4.43	26.44%	8.56%	4.43	26.44%	8.56%	4.43	26.44%	8.56%
DispNet [20]	40ms [C]	4.91	49.47%	11.75%	3.59	34.15%	10.47%	1.93	16.82%	7.02%
CRL [23]	152ms [C]	6.49	51.05%	12.43%	2.25	13.70%	6.18%	1.36	6.02%	4.69%
PSMNet [4]	456ms [P]	—	—	—	2.40	19.09%	7.29%	1.64	10.03%	6.70%
iResNet [17]	122ms [C]	5.10	42.99%	11.14%	2.19	13.42%	6.70%	1.24	4.27%	4.23%
SegStereo [32]	225ms [C]	4.14	36.81%	10.55%	2.17	15.75%	7.40%	1.32	5.89%	4.78%
ResNetCorr [32]	170ms [C]	3.54	34.90%	10.94%	2.27	17.55%	7.74%	1.37	6.75%	4.97%
EdgeStereo [28]	320ms [C]	5.02	36.85%	10.12%	2.05	12.39%	6.12%	1.19	3.47%	4.17%
GuideNet	12ms [P]	4.92	42.02%	11.13%	2.43	19.86%	8.33%	1.51	8.89%	5.85%

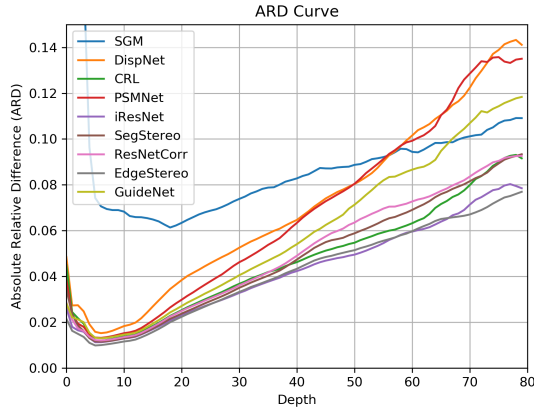
ResNetCorr [31], and EdgeStereo [28, 29]. Due to the different size of three training datasets, we keep the same epochs, so that the maximum training iterations for FlyingThings3D [20], Cityscapes [7], and our DrivingStereo are set to 62.5K, 62.5K, and 500K respectively. The other parameters are set as same as the training of our GuideNet described in Sec. 5.2. In addition to traditional EPE and D1 error, the proposed metrics are used to evaluate the stereo methods. For distance-aware metrics, the measuring range r for each depth point is set to $4m$. The sampling interval is set to $8m$ so that 10 depth values are accumulated to compute GD. When drawing the distance-aware curves, the sampling interval is set to $1m$ for better visualization. For semantic-aware metrics, the threshold θ is set to 1.10.

In Tab. 3, we provide the baseline results on the KITTI Stereo 2015 [21] and the test set of our DrivingStereo. Compared to the FlyingThings3D-trained and Cityscapes-trained models, the DrivingStereo-trained models achieve the best performance on both KITTI and our test set, which shows the capacity of DrivingStereo for deep stereo models. Among the models, EdgeStereo [28] gets the highest accuracy, followed by iResNet [17], SegStereo [32] and CRL [23]. On the other hand, the mean and the variance of these results evaluated on our DrivingStereo are larger than the results evaluated on KITTI. This reflects that our dataset is more challenging for current stereo methods due to the large amounts of frames and diverse scenarios.

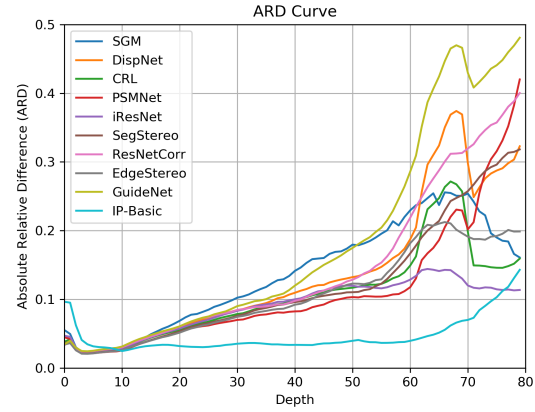
In order to further analyze the properties of our dataset and current methods, we utilize the distance-aware and semantic-aware metrics to perform evaluations. Fig. 4 depicts the distance-aware curves of stereo methods which are trained on DrivingStereo dataset and evaluated on KITTI [21] and our test set. Compared to EPE or D1 error, these metric curves intuitively exhibit the performance of stereo methods on different ranges. The general trend is that the average difference slightly decreases in the interval of $0 \sim 10m$ and then increases with distances. Benefited from the structure of cascade refinement, iResNet [17], CRL [23] and EdgeStereo [28] are better than other methods, especially on KITTI dataset. Besides, our GuideNet achieves similar accuracy on short and middle distances, which shows the feasibility of our design. From the radar maps of semantic-aware metrics in Fig. 5, we discover that most of deep learning methods behave better on the ground and vehicles, but the accuracy is slightly lower on the classes of human, construction, and others. The above results illustrate that the distance-aware and semantic-aware metrics facilitate the exploitation of stereo methods.

6. Discussion

Comparison with LiDAR. Most of perception systems adopt LiDAR to measure ranges. Here, we compare the stereo accuracy with LiDAR in terms of distance-aware and semantic-aware metrics. State-of-the-art depth completion

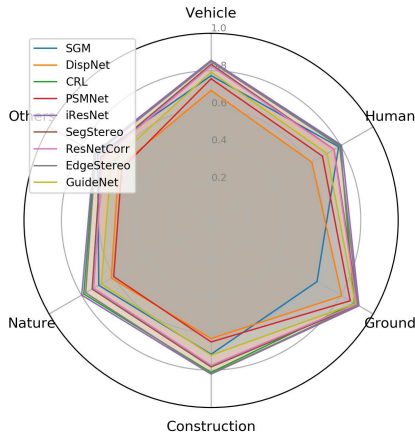


(a) Evaluated on our DrivingStereo

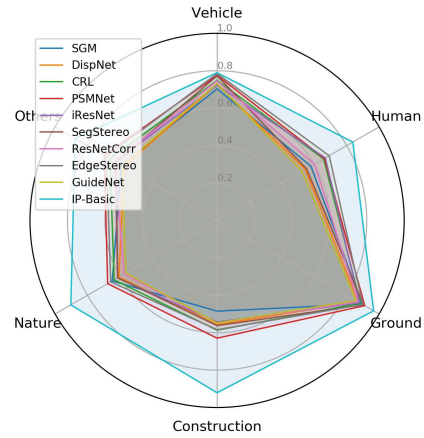


(b) Evaluated on KITTI Stereo 2015 [21]

Figure 4. The distance-aware ARD curves.



(a) Evaluated on our DrivingStereo [20]



(b) Evaluated on KITTI Stereo 2015 [21]

Figure 5. The semantic-aware MR radar maps.

method [16] is employed to produce dense LiDAR outputs and make a fair comparison on KITTI benchmark [21].

From the distance-aware curve in Fig. 4 and the semantic-aware map in Fig. 5, we find existing stereo models reach comparable accuracy with LiDAR-based methods in short ranges and the areas of ground and vehicle, while the gaps on long distances and complex objects remain. It is natural since disparity relies on focal length and baseline distance between stereo cameras.

To shorten this gap, more remote disparity labels are necessary for stereo training, and the baseline between cameras should be stretched so as to increase stereo identification. If possible, the telephoto-lens camera could be utilized. Meanwhile, the loss function needs to be more sensitive to long ranges and foreground objects. For quantitative evaluation, our DrivingStereo dataset and metrics is an appropriate platform to evaluate the performance of such methodologies.

7. Conclusion

In this work, we build a large-scale stereo dataset for driving scenarios and present new evaluation metrics. The baseline results demonstrate the capacity of our dataset and validate the effectiveness of our new metrics. With the progress to be made on stereo performance, the gap between the stereo cameras and the expensive sensors like LiDAR can be further decreased, facilitating an affordable stereo matching solution to autonomous driving vehicles.

Acknowledgment

This work was supported in part by the National Key R&D Program of China under Grant No. 2017YFB1302200, by TOYOTA TTAD-2019-08, by Joint Fund of NORINCO Group of China for Advanced Research under Grant No. 6141B010318, and by CUHK direct fund (No.4055098).

References

- [1] Hassan Abu Alhaija, Siva Karthik Mustikovela, Lars Mescheder, Andreas Geiger, and Carsten Rother. Augmented reality meets deep learning for car instance segmentation in urban scenes. In *BMVC*, 2017. 5
- [2] S. Baker, S Roth, D Scharstein, M. J Black, J. P Lewis, and R Szeliski. A database and evaluation methodology for optical flow. *IJCV*, 2011. 1
- [3] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012. 2, 5, 6
- [4] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *CVPR*, 2018. 2, 3, 6, 7
- [5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI*, 2016. 3
- [6] Xinjing Cheng, Peng Wang, and Ruigang Yang. Learning depth with convolutional spatial propagation network. *arXiv preprint arXiv:1810.02695*, 2018. 1, 3
- [7] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 2, 3, 5, 6, 7
- [8] 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 *ICCV*, 2015. 3, 4
- [9] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*, 2014. 5
- [10] A Gaidon, Q Wang, Y Cabon, and E Vig. Virtual worlds as proxy for multi-object tracking analysis. In *CVPR*, 2016. 1, 2, 5, 6
- [11] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 1, 2, 3, 5, 6
- [12] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *TPAMI*, 2008. 2, 3, 6, 7
- [13] Heiko Hirschmuller and Daniel Scharstein. Evaluation of stereo matching costs on images with radiometric differences. *TPAMI*, 2009. 1, 2, 3
- [14] Xinyu Huang, Peng Wang, Xinjing Cheng, Dingfu Zhou, Qichuan Geng, and Ruigang Yang. The apolloscape open dataset for autonomous driving and its application. *arXiv:1803.06184*, 2018. 2
- [15] 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 *ICCV*, 2017. 2, 3
- [16] Jason Ku, Ali Harakeh, and Steven L Waslander. In defense of classical image processing: Fast depth completion on the cpu. *arXiv preprint arXiv:1802.00036*, 2018. 2, 8
- [17] 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 *CVPR*, 2018. 2, 3, 6, 7
- [18] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 3, 4
- [19] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The oxford robotcar dataset. *The International Journal of Robotics Research*, 2017. 2, 5
- [20] 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 *CVPR*, 2016. 1, 2, 3, 4, 5, 6, 7, 8
- [21] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *CVPR*, 2015. 1, 2, 3, 4, 5, 6, 7, 8
- [22] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Buló, and Peter Kotschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *ICCV*, 2017. 2
- [23] Jiahao Pang, Wenxiu Sun, JS Ren, Chengxi Yang, and Qiong Yan. Cascade residual learning: A two-stage convolutional neural network for stereo matching. In *ICCV Workshop*, 2017. 2, 3, 6, 7
- [24] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *ICRA*, 2011. 3
- [25] Daniel Scharstein, Heiko Hirschmuller, York Kitajima, Greg Krathwohl, Nera Nei, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *GCPR*, 2014. 2, 5, 6
- [26] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 2002. 3
- [27] Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *CVPR*, 2017. 2, 5, 6
- [28] Xiao Song, Xu Zhao, Liangji Fang, and Hanwen Hu. Edgestereo: An effective multi-task learning network for stereo matching and edge detection. *arXiv preprint arXiv:1903.01700*, 2019. 2, 3, 7
- [29] Xiao Song, Xu Zhao, Hanwen Hu, and Liangji Fang. Edgestereo: A context integrated residual pyramid network for stereo matching. In *ACCV*, 2018. 3, 7
- [30] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *3DV*, 2017. 3
- [31] Guorun Yang, Zhidong Deng, Hongchao Lu, and Zeping Li. Src-disp: Synthetic-realistic collaborative disparity learning for stereo matching. In *ACCV*, 2018. 7
- [32] Guorun Yang, Hengshuang Zhao, Jianping Shi, Zhidong Deng, and Jiaya Jia. Segstereo: Exploiting semantic information for disparity estimation. In *ECCV*, 2018. 2, 3, 6, 7
- [33] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv:1805.04687*, 2018. 2

- [34] Jure Zbontar and Yann LeCun. Stereo matching by training a convolutional neural network to compare image patches. *JMLR*, 2016. 3
- [35] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 3, 5