SEMI-SUPERVISED 360° DEPTH ESTIMATION FROM MULTIPLE FISHEYE CAMERAS WITH PIXEL-LEVEL SELECTIVE LOSS

Jaewoo Lee, Daeul Park, Dongwook Lee, and Daehyun Ji

Samsung Advanced Institute of Technology (SAIT)

ABSTRACT

In this paper, we study a practical omnidirectional depth estimation with neural networks that enables effective learning on real world data obtained using wide-baseline multiple fisheye cameras. Most previous approaches only used synthetic data providing dense and accurate depth ground truth (GT). However, it is unrealistic to acquire such high quality GT data in real world due to limitations of the existing depth sensors. We first introduce two critical problems that can reduce the accuracy of depth estimation: depth GT sparsity and sensor calibration error. We then propose a novel semi-supervised learning method using pixel-level loss that selectively uses supervised loss and unsupervised re-projection loss according to existence of GT. Empirical results demonstrate that our method efficiently reduces the performance degradation in both simulation on synthetic data and real world data using sparse depth sensor.

Index Terms— depth estimation, fisheye camera, semi-supervised, sparse ground truth, calibration error

1. INTRODUCTION

Fisheye or omnidirectional cameras are used to sense the surroundings owing to their wide field of view [1, 2, 3] in many applications. In particular, in the automotive industry, the fisheye cameras are already being widely used in mass-produced vehicles for the surround view monitoring (SVM) function to assist parking by showing bird-eye view image. However, recently, there is a growing demand to expand the fisheye cameras to the higher level functions for autonomous driving, such as object detection, lane detection, and 3D SVM system. Surround depth estimation from multiple fisheye cameras is one of the most important techniques for implementing the next-level features.

To estimate the depth from multiple cameras, plane-sweeping stereo has been studied for numerous years [4, 5, 6, 7]. In plane-sweeping stereo, multi-view images are projected onto virtual planes at several depths from the reference image plane to generate a cost volume, and depth maps are estimated using this cost volume. Many recent approaches have been combined the convolution neural network and the plane-sweeping stereo for perspective images [8, 9, 10, 11].

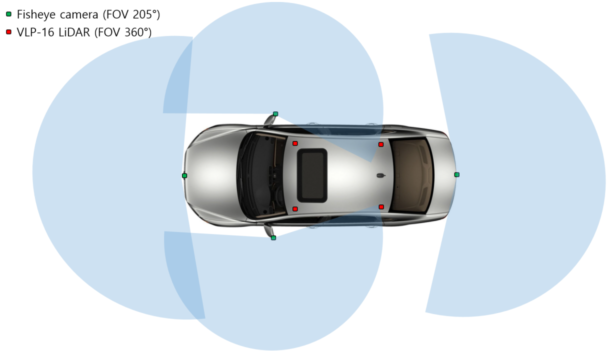

Fig. 1: Sensor configuration of fisheye cameras and LiDARs in our vehicle to gather real data.

Unlike to the perspective images, efforts to apply the convolution neural network to 360° surround depth estimation with multiple fisheye cameras has only begun more recently. Won et al, proposed a novel method for estimating the all-around depth from four fisheye images [12]. They converted fisheye images into equi-rectangular images by spherical sweeping to make the cost volume in 3D space, and then use 3D convolutional neural network to classify the depth index. For training the network, they use supervised L1 loss for every pixel. After that, there have been some extended works, such as use of icosahedron-based representation for robust features in highly distorted images [13], polar angle distortion compensation layer [14], and fast spherical sweeping for real time processing [15]. However, all of them verified the performance only in synthetic data which provides dense and completely accurate ground truth, although real depth sensors such as LiDAR generate sparse and noisy depth measurements due to its physical limitation and interference from the environments. Therefore, to use the depth estimation algorithm in real application, it is essential to consider the limitation and property of the depth ground truth from the real sensors.

In this study, we focus on analyzing how large the domain gap between synthetic and real data is, and solving problem to calculate an accurate all-around depth in even real data. We first investigate two major limitations of the real data and their effects on the performance of the conventional algorithms. Then we proposed a novel semi-supervised learning method with pixel-level loss that selectively uses supervised loss and unsupervised re-projection loss according to existence of GT. To show the effectiveness of the proposed algorithm in real data, we built real vehicle with 4 fisheye cameras and four Velodyne VLP-16 LiDARs on the roof, which is illustrated in Fig. 1. Our proposed method is shown to be effective by both the simulation with the synthetic data and experiments with real world data.

2. DOMAIN GAP BETWEEN SYNTHETIC AND REAL DATA

Usually, synthetic dataset for depth estimation is gathered by rendering software such as Blender, so that we can get not only dense but also accurate depth GT. In practice, however, that is impossible situation due to sparseness of the existing depth sensor and the errors of calibration parameters between the cameras. We show how those limitations are critical in the real world data, with properly designed simulations on the synthetic data.

2.1. Sparse GT

In autonomous driving application which requires relatively far range depth estimation, LiDAR sensor is mostly used but it is inherently sparse so that it is impossible to get dense depth GT for every ray from the camera. It means that only partial pixels have its ground truth. Then the previous works using the supervised loss such as [3, 12, 16] cannot utilize the whole pixels to train the network in the real data. To clarify the effect on the performance of the supervised method, we design sparse GT environment on the Omnithings synthetic data introduced in [12]. We randomly sample the pixels based on several ratios to the whole pixels and train the network of [12] by using partial ground truth on sampled pixels. Details about the dataset and experimental environment are presented in Section 4. As shown in Table 1, the performance of the conventional algorithm is severely degraded as the GT sparsity increases.

2.2. Sensor calibration error

Unlike the synthetic data where exact calibration parameters can be known, we need to perform calibration work in the real world and some calibration errors are inevitable. Intrinsic and extrinsic parameters of all cameras are needed for two objectives. First, they are used in projecting LiDAR data into each image plane for depth GT generation. If there are calibration noise, LiDAR points are projected wrong pixel location so that inaccurate depth GT is generated. Second, they are required in our network (Section 3) to warp all 2D feature maps into 3D spheres to make cost volume. To show the effect of calibration error on the performance, we generate random calibration noise with several variances and train the network.

Table 1: Depth accuracy of [12] in according to GT sparsity level and calibration noise in synthetic dataset.

GT Sparsity	Avg. MAE	Avg. RAE (%)
Full	2.22	16.20
0.01	2.43	18.33
0.0001	3.96	26.12

Noise Variance	Avg. MAE	Avg. RAE (%)
0	2.23	16.12
0.1	4.96	32.01
0.2	6.32	42.12

As shown in Table 1, as calibration error variance increases, depth accuracy severely deteriorates.

3. SEMI-SUPERVISED LEARNING WITH PIXEL-LEVEL SELECTIVE LOSS

3.1. Network structure

We basically use a network structure that described as in [12], but add new head network structure to calculate re-projection loss without ground truth values. The input to the network is a set of fisheye images. Each image is first encoded into a deep feature map, and then each feature map is aligned to the cost volume which will be used to measure the geometric compatibility at different depth values. We define a set of ordered 3D spheres $\mathcal{P}$, each inverse radius are linearly spaced between user parameter $1/d_{max}$ and $1/d_{min}$. Each feature map from ith camera is warped to the equi-rectangular map S_i in spherical coordinate $\langle \phi, \theta \rangle$ by using each of the hypothesised alternative depth d as

$$S_i(\phi, \theta, n, c) = U_c\left(\Pi_i\left(\bar{p}(\phi, \theta) \cdot d_n\right)\right) \tag{1}$$

where c is channel index of the feature map, $\bar{p}$ is a unit lay, d_n is a depth value of depth index n, and Π_i is ith camera projection function. We stack all warped equi-rectangular maps along the depth channel to make 4D cost volume of the size $H \times W \times N \times C$. This cost volume have the benefit of allowing the network to leverage inputs from multiple viewing angles.

Encoder-decoder architecture with 3D convolution and deconvolution layers is used to regularize the 4D cost volume and classify the pixel-wise inverse depth index as $\hat{n}(\phi, \theta)$ in form of 360° equi-rectangular map. In the equi-rectangular depth map, the value of each pixel means depth of a 3D ray in a referenced spherical coordinate. Finally, we calculate the depth estimate in metric scale by following decoding equation:

$$\frac{1}{\hat{d}(\phi, \theta)} = \frac{1}{d_{max}} + \frac{\hat{n}(\phi, \theta)}{N} \cdot \left(\frac{1}{d_{min}} - \frac{1}{d_{max}}\right) \tag{2}$$

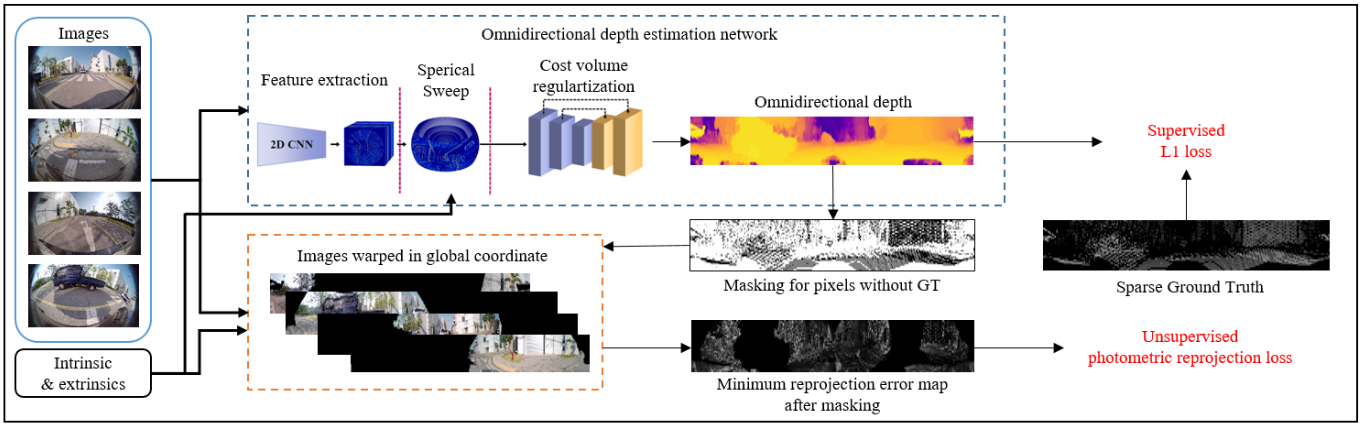
Images
Intrinsic & extrinsics
Omnidirectional depth estimation network
Feature extraction
2D CNN
Sperical Sweep
Cost volume regulartization
Omnidirectional depth
Supervised L1 loss
Images warped in global coordinate
Masking for pixels without GT
Sparse Ground Truth
Minimum reprojection error map after masking
Unsupervised photometric reprojection loss

Fig. 2: Overview of the proposed method.

3.2. Pixel-level selective loss

Based on existence of ground truth, we propose a pixel-level selective loss: smoothed L1 loss is calculated on the pixels with ground truth in supervised manner and re-projection loss is calculated on pixels without ground truth in unsupervised manner.

To calculate re-projection loss, all images are warped to the equi-rectangular map in the global coordinate based on depth inference result and sensor calibration parameters as follows:

$$S_i^I(\phi,\theta) = I_i\left(\Pi_i\left(\bar{p}(\phi,\theta)\cdot\hat{d}(\phi,\theta)\right)\right). \tag{3}$$

Each warped image has its own blind region at the outside of its field-of-view. For any two fisheye cameras i and j which have overlap region, we calculate re-projection loss for the pixels in the overlap region as

$$L_{\text{re-projection}}(i,j) = \sum_R pe(S_i^I, S_j^I) \tag{4}$$

where R is the set of pixels in overlap region of camera i and j, and we define the projection error with photometric L1 loss and Structural Similarity Index (SSIM) as follows:

$$pe(I_a,I_b) = \frac{\alpha}{2}(1-SSIM(I_a,I_b))+(1-\alpha)\left\|I_a - I_b\right\|. \tag{5}$$

We also use edge-aware smoothness loss and auxiliary loss to make better depth regression loss by following [17]. Final re-projection loss $L_{\text{re-projection}}$ is calculated by adding the loss in (4) for all combination of two different cameras:

$$L_{\text{re-projection}} = \sum_{\forall(i,j),i\neq j} L_{\text{re-projection}}(i,j) \tag{6}$$

By adding L1 loss and re-projection loss with a weight λ, we can have the total loss as follows:

$$L_{\text{total}} = L_{\text{L1}} + \lambda \cdot L_{\text{re-projection}} \tag{7}$$

All process of training is illustrated in Fig. 2.

4. EXPERIMENT RESULTS

4.1. Datasets

We used both the synthetic and the real data to present the performance of the proposed algorithm. For synthetic data, we used Omnithings and Sunny dataset from [12] with 9916 training set and 1324 validation set. We gathered 24192 training and 3712 validation real dataset by using our vehicle with configuration in Fig. 1 of four fisheye images and four LiDAR pointcloud data. Multi-cameras and multi-LIDARs calibration to get intrinsic and extrinsic parameters between the cameras are performed by using [18]. We gathered 24192 sets for training and 3712 sets for evaluation.

4.2. Experimental setup

We set the range of depth to estimate as $d_{min} = 2$m and $d_{max} = 50$m, and the vertical field-of-view to 36°. For synthetic data, we simulated sparse GT environment with ratio 0.01 and 0.0001 and calibration noise environment with variance 0.2. For real data, input image size is 1920×1080, and we set the size of resulting equi-rectangular depth map to 640×64. Learning rate is initially set to $3e^{-3}$ for the first 20 epochs and reduces 10 times in every 10 epochs. Global coordinate system for real data is defined by following same way of [12]; the y-axis is perpendicular to the plane closest to all camera centers and an the origin is the center of the all camera centers. In the all experiments, we use two kinds of performance measure which are mean absoulte error (MAE) for inverse depth index [12] and relative absolute error (RAE) for depth value as follow:

$$MAE(\phi,\theta) = \frac{|\hat{n}(\phi,\theta) - n^*(\phi,\theta)|}{N} \times 100 \tag{8}$$

$$RAE(\phi,\theta) = \frac{\left|\hat{d}(\phi,\theta) - d^*(\phi,\theta)\right|}{d^*(\phi,\theta)} \times 100. \tag{9}$$

Table 2: Depth accuracy of [12] and the proposed algorithm in synthetic dataset with sparse GT and calibration noise

GT sparsity	Algorithm	RAE (%)						MAE
		10m	20m	30m	40m	50m	Avg.	Avg.
0.0001	OmniMVS [12]	12.11	17.63	22.95	34.27	43.64	26.12	3.96
	Proposed (w/ scratch)	12.26	17.63	23.79	32.90	38.82	25.08	3.99
	Proposed (w/ pretrained)	12.56	17.18	22.50	29.40	37.54	**23.83**	**3.93**
0.01	OmniMVS [12]	7.53	12.37	17.66	22.09	32.00	18.33	2.43
	Proposed (w/ scratch)	7.91	12.43	17.52	19.57	25.64	**16.61**	2.46
	Proposed (w/ pretrained)	7.25	11.62	16.93	19.71	28.25	16.75	**2.30**

Noise variance	Algorithm	RAE (%)						MAE
		10m	20m	30m	40m	50m	Avg.	Avg.
0.2	OmniMVS [12]	19.20	24.27	42.44	57.85	66.83	42.12	6.32
	Proposed (w/ scratch)	19.16	24.60	36.46	52.82	62.25	39.05	6.30
	Proposed (w/ pretrained)	17.51	24.65	26.30	41.05	52.27	**32.35**	**5.49**

Table 3: Depth accuracy of the conventional algorithm [12] and the proposed algorithm in real dataset

Algorithm	RAE (%)						MAE
	10m	20m	30m	40m	50m	Avg.	Avg.
OmniMVS [12]	7.49	12.07	13.67	19.56	30.68	16.69	1.89
Proposed (w/ pretrained)	6.90	11.96	11.88	13.90	22.47	**13.42**	**1.65**

To compare the performance in aspect of the supervised loss, the MAE measure is useful to show the average error for the inverse depth index. However, in real autonomous driving applications, the relative error of the actual depth value, not the index, is commonly used as a performance measure. For that reason, we added a second RAE measure to compare the performance of the algorithms.

4.3. Results

Table 2 shows that the proposed semi-supervised training provides the better performance than training with only supervised loss in synthetic data in existence of sparse GT or sensor calibration error. For all conditions, the proposed method shows the lower error level in both performance measures. Even in real dataset, the proposed algorithm is confirmed to improve depth accuracy efficiently as in Table 3. We also find that the better performance is achieved by step-wise training: first train with only supervised loss until it converges and then train with the proposed pixel-level selective loss from pre-trained weight. It means that the proposed unsupervised re-projection loss gives better guidance to network when the training starts at better initial weight. Fig. 3 shows qualitative comparison between the OmniMVS [12] and the proposed method. In case of sparse GT, quality of depth map from [12] is degraded, especially at the edges and backgrounds. By the proposed semi-supervised learning, the depth map quality is improved at the both edges and background regions at far range.

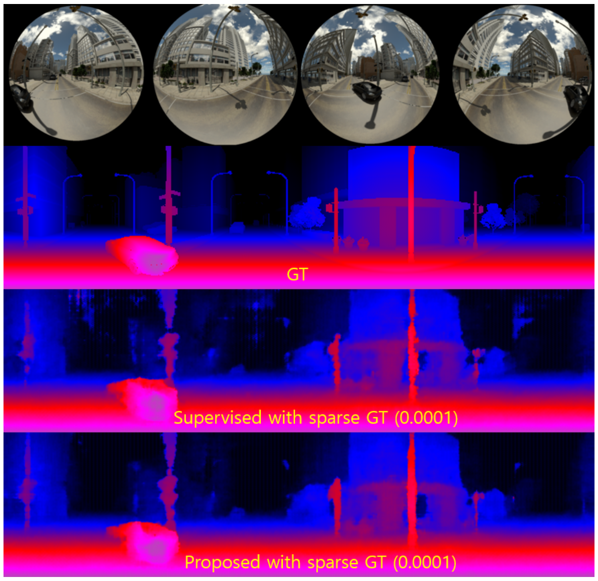
GT
Supervised with sparse GT (0.0001)
Proposed with sparse GT (0.0001)

Fig. 3: Qualitative comparison on synthetic data. From top; input images, GT, inverse depth from [12] and the proposed

5. CONCLUSION

We analyze how practical problems in real world lead to performance degradation and then propose semi-supervised training algorithm with pixel-level selective loss which efficiently improve the performance. The proposed algorithm shows the better performance in both synthetic and real data.

References

[1] Lionel Heng, Benjamin Choi, Zhaopeng Cui, Marcel Geppert, Sixing Hu, Benson Kuan, Peidong Liu, Rang Nguyen, Ye Chuan Yeo, Andreas Geiger, Gim Hee Lee, Marc Pollefeys, and Torsten Sattler, "Project autovision: Localization and 3d scene perception for an autonomous vehicle with a multi-camera system," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 4695–4702.

[2] Zhaopeng Cui, Lionel Heng, Ye Chuan Yeo, Andreas Geiger, Marc Pollefeys, and Torsten Sattler, "Real-time dense mapping for self-driving vehicles using fisheye cameras," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 6087–6093.

[3] Changhee Won, Jongbin Ryu, and Jongwoo Lim, "Omnimvs: End-to-end learning for omnidirectional stereo matching," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 8986–8995.

[4] D Marr and T Poggio, "Cooperative computation of stereo disparity," *Science*, vol. 194, no. 4262, pp. 283–287, 1976.

[5] M. Okutomi and T. Kanade, "A multiple-baseline stereo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 4, pp. 353–363, 1993.

[6] R.T. Collins, "A space-sweep approach to true multi-image matching," in *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1996, pp. 358–363.

[7] David Gallup, Jan-Michael Frahm, Philippos Mordohai, Qingxiong Yang, and Marc Pollefeys, "Real-time plane-sweeping stereo with multiple sweeping directions," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[8] 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 *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 66–75.

[9] Jia-Ren Chang and Yong-Sheng Chen, "Pyramid stereo matching network," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5410–5418.

[10] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang, "Deepmvs: Learning multi-view stereopsis," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2821–2830.

[11] Ren Komatsu, Hiromitsu Fujii, Yusuke Tamura, Atsushi Yamashita, and Hajime Asama, "Octave deep plane-sweeping network: Reducing spatial redundancy for learning-based plane-sweeping stereo," *IEEE Access*, vol. 7, pp. 150306–150317, 2019.

[12] Changhee Won, Jongbin Ryu, and Jongwoo Lim, "End-to-end learning for omnidirectional stereo matching with uncertainty prior," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.

[13] Ren Komatsu, Hiromitsu Fujii, Yusuke Tamura, Atsushi Yamashita, and Hajime Asama, "360 depth estimation from multiple fisheye images with origami crown representation of icosahedron," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 10092–10099.

[14] Ziye Lai, Dan Chen, and Kaixiong Su, "Olanet: Self-supervised 360 depth estimation with effective distortion-aware view synthesis and l1 smooth regularization," in *2021 IEEE International Conference on Multimedia and Expo (ICME)*, 2021, pp. 1–6.

[15] Andreas Meuleman, Hyeonjoong Jang, Daniel S. Jeon, and Min H. Kim, "Real-time sphere sweeping stereo from multiview fisheye images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 11423–11432.

[16] Changhee Won, Jongbin Ryu, and Jongwoo Lim, "Sweepnet: Wide-baseline omnidirectional depth estimation," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 6073–6079.

[17] Clement Godard, Oisin Mac Aodha, Michael Firman, and Gabriel Brostow, "Digging into self-supervised monocular depth estimation," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 3827–3837.

[18] Wonmyung Lee, Changhee Won, and Jongwoo Lim, "Unified calibration for multi-camera multi-lidar systems using a single checkerboard," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 9033–9039.