IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 15, 2018, accepted July 16, 2018, date of publication July 19, 2018, date of current version August 15, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2857703

Wearable Depth Camera: Monocular Depth Estimation via Sparse Optimization Under Weak Supervision

LI HE[1], (Member, IEEE), CHUANGBIN CHEN[1], TAO ZHANG[1], HAIFEI ZHU[1], (Member, IEEE), AND SHAOHUA WAN[2], (Member, IEEE)

[1]School of Electromechanical Engineering, Guangdong University of Technology, Guangzhou 510006, China
[2]School of Information and Safety Engineering, Zhongnan University of Economics and Law, Wuhan 430073, China

Corresponding author: Shaohua Wan (shaohua.wan@ieee.org)

This work was supported in part by the National Natural Science Foundation of China under Grant 61673125 and Grant 61703115, in part by the Leading Talents of Guangdong Province Program, in part by the Frontier and Key Technology Innovation Special Funds of Guangdong Province under Grant 2016B090910003, and in part by the Program of Foshan Innovation Team of Science and Technology under Grant 2015IT100072.

ABSTRACT Depth estimation is essential for many human-object interaction tasks. Despite its advantages, traditional depth sensors, including Kinect or depth camera, are always not wearable-friendly due to several critical drawbacks, such as over-size or over-weight. Monocular camera, on the other hand, provides a promising solution with limited burden to users and attracts more and more attentions in the literature. In this paper, we propose a depth estimation method with monocular camera. Our main idea lies in the weak-supervised learning model of monocular depth estimation based on left and right consistency. To learn an accurate depth estimation, on our training step, we employ LiDAR data, which are generated by laser radar with very high depth accuracy, to semi-supervise the learning scheme. We train our network on ResNet and propose a new penalty function, which takes into account the LiDAR depth loss in training. Compared with several state-of-the-art monocular camera depth estimators, our proposed method obtains the highest depth accuracy.

INDEX TERMS Wearable devices, depth estimation, deep learning, weak supervision, sparse optimization.

I. INTRODUCTION

In the last decade, we have witnessed the bloom of wearable devices in our daily life. Many researches have been deployed on wearable devices with topics ranging from communications [1]–[3] to computer vision [4]. In wearable-device-orientated computer vision, accurate depth has been proved to improve the performance of many applications, e.g., semantic segmentation [5], pose estimation and body posture recognition [6], with respect to its RGB-only counterpart.

Traditional methods of estimating depth from monocular image enforce optical geometric constraints or some environmental assumptions, such as Structure from Motion (SFM), focus or variations in illumination. In absence of such constraints or assumptions, however to develop a computer vision system capable of accurately monocular cues is a task. There are estimating depth maps by exploiting challenging two difficulties in this task. One is that a common computer vision system can extract information used for inferring 3D structure from monocular image like human brain. The other is that the task is a technically ill-posed problem: a 2D image to an infinite number of real world scenes.

In order to finally obtain reliable 3D structure information, SFM algorithm [7] is necessary to make the longer baseline between the two cameras. Research of binocular or multi-view method has been well studied in general. However, there are still some difficulties, such as weak texture region matching in this topic. In addition, due to its efficiency, many researchers focus on the restoration of scene depth information from a single image captured from a monocular camera. Monocular camera depth estimation is low-cost, convenient and flexible in application and, as a result, suitable for light-weight devices in particular the wearable devices.

In the absence of optical geometric constraints or related environmental assumptions, estimating the scene depth from a monocular image is a morbid issue, that is, a two-dimensional image can be generated from infinite number of real 3D scenes. This intrinsic uncertainty in mapping a single image to a depth map determines that in principle it

is impossible for the visual model to estimate an exact depth value from a single image. However, it is also well-known that human is able to perceive fairly reliable 3D structure from one single eye. This shows that it is feasible to estimate the depth map with certain reliability from the monocular image. The difficulty of monocular camera depth estimation is how to design a computer vision system to estimate a relatively reliable depth map like the human monocular vision.

In the field of computer vision, many works have been done in the early stage to study the task of image depth estimation from monocular images. In 2005, Saxena *et al*. [8] proposed make3D. Make3D first runs superpixel segmentation of the image. Taking each super-pixel as a plane, make3D estimate the correlations of these planes by Markov random field and finally estimates the depth of each plane. Subsequently, Liu *et al*. [9] applied the continuous conditional random field (CCRF) to the depth estimation model and, similar to make3D, taken into account the correlation of the planes corresponding to superpixels. This kind of methods are used in the modeling process to extract the manual features from the image, and these features do not represent the 3D structure of the scene properly. Therefore, the performance of these methods is not very satisfactory.

Recently, breakthroughs have been made in the field of deep learning. CNN has achieved great success in many other computer vision tasks mainly due to its powerful regression and self-taught feature expressions abilities. The multi-scale CNN proposed by Eigen *et al*. [10] is one of the first methods to apply CNN to monocular image depth estimation. Multi-scale CNN is divided into two parts when dealing with depth estimation tasks. First, the global structure of one scene is estimated by the coarse-scale part of the network. Then, the fine-scale part of the network uses the local information of the underlying features of CNN to optimize the global structure. Multi-scale CNN also proposes a loss function for supervised learning of depth estimation problem. The loss function consists of the absolute depth of the scene and the relative 3D structure learning. In addition, Eigen and Fergus [11] in their subsequent works extended their researches.

Following the study by Eigen *et al*., a large number of works on the application of CNN-based depth estimation from monocular images have been proposed. The DCNF-FCSP model proposed by Liu *et al*. [12] unifies CNN and CRF in a deep learning framework. CNN, in this work, extracts relevant features from the image and CRF provides the final prediction result which is smooth and edge-preserving. Because of the superpixel segmentation, the number of nodes in CRF is drastically reduced, making the accurate inference process of the maximum posterior probability (MAP) of the CRF computationally feasible.

Taking into account that the depth monitoring information provided by sensors are not ideal in general, the performance of deep learning models, thus, may be depressed due to the uncertainty. Garg *et al*. [13] proposed a reconstruction error using stereoscopic image pairs as the unsupervised learning method. Their works enable the CNN train and predict depth maps without in-depth monitoring information.

Many related work have demonstrated that the number of CNN stack layers is very important for the performance of CNN. Thanks to the birth of deeper CNN structure, many computer vision tasks, including depth estimation, have achieved better results. However, in depth estimation tasks, the network structure is not easy to be too deep because the potential and fatal problem of gradient disappearance, which makes CNN difficult to train. In order to increase the number of layers to obtain better performance and, meanwhile, to keep CNN easy to train, Cao *et al*. [14] applied the deep residual network proposed by He *et al*. [15] to depth estimation. Cao *et al*. quantified the scene depth and treated the depth estimation problem as a pixel-level classification problem.

Recently, Godard *et al*. [17] proposed an unsupervised deep neural network with left-right consistency which shows promising prediction accuracy. The main idea in [17] is taking the right camera image as the supervisor to train the left one and vice versa. The model in [17] is able to build a 512×256 depth map within 35ms, indicating real-time solution to many wearable applications. In this paper, we use a similar fashion as to [17] although we additionally employ the sparse LiDAR depth data as the weak supervision signal to further optimize Godard's model.

The rest of this paper is organized as follows. Section II gives the reviews of related works in depth estimation from left-right consistency. Our network, in particular the construction of LiDAR loss term, is described in Section III. Section IV shows the experimental results on benchmark datasets. We conclude our works in Section V.

II. MONOCULAR DEPTH ESTIMATION WITH LEFT-RIGHT CONSISTENCY

Godard *et al.* [17] use an unsupervised learning method to estimate the depth value of the monocular RGB image. The basic idea is to match the pixels of left and right views to get the disparity map. The depth map is calculated from Eq. (1) based on the obtained disparity, camera baseline b and focal length f.

$$d = b \times f / disparity, \tag{1}$$

The architecture proposed by Godard *et al.* [17] is inspired by Mayer's DispNet [28] which uses image reconstruction loss to minimize photometric errors. In order to improve estimation, Godard uses Left-Right Consistency to optimize the model. Works in [17] first takes the left view as input and use the right view as the supervision. The main idea in their works is the assumption that with a perfect depth estimation in hand, we are then able to perfectly reconstruct the right view from the left one and vice versa. As shown in [17], minimizing the joint loss of these two processes can get a better prediction depth accuracy. Works in [17] also use four different scales as inputs to improve the output resolution of the neural network.

In [17], the loss function follows

$$C_s = \alpha_{ap}(C^l_{ap} + C^r_{ap}) + \alpha_{ds}(C^l_{ds} + C^r_{ds}) + \alpha_{lr}(C^l_{lr} + C^r_{lr}). \tag{2}$$

The first part of the loss function, C^l_{ap} and C^r_{ap}, are photometric image reconstruction cost of the left and right camera, respectively. C^l_{ap} of the left camera follows

$$C^l_{ap} = \frac{1}{N}\sum_{ij} \alpha \frac{1 - SSIM(I^l_{ij}, \hat{I}^l_{ij})}{2} + (1-\alpha)\left\| I^l_{ij} - \hat{I}^l_{ij} \right\|, \tag{3}$$

where the superscript l indicates the left camera in general, *SSIM* is the structural similarity index and α is the weight.

In training, the network learns to generate images by sampling pixels from the opposite stereo image. The image formation model uses the image sampler of the Spatial Transformation Network (STN) [22] plus the disparity map to sample the input image. STN uses bilinear sampling where the output pixel is a weighted sum of four input pixels. Compared to other methods [13], [23], the bilinear sampler is differentiable and can be seamlessly integrated into fully convolution architecture, a promising property indicating no additional designs on cost function simplification or approximation.

The second part of the loss function, C_{ds}, is the parallax smoothness loss. This part encourages local smoothness of the parallax and the $L1$ loss function is used in the disparity gradient. Since depth discontinuities typically occur at image gradients, similar to [24], we use image gradients to weight the cost function using edge-aware terms.

$$C^l_{ds} = \frac{1}{N}\sum_{ij} \left|\partial_x d^l_{ij}\right| e^{-\left\|\partial_x I^l_{ij}\right\|} + \left|\partial_y d^l_{ij}\right| e^{-\left\|\partial_y I^l_{ij}\right\|}, \tag{4}$$

where $\partial_x d^l_{ij}$ is the disparity gradients on the x-axis of the left image and $\partial_x I^l_{ij}$ stands for the image gradients on the x-axis direction.

The third part, C_{lr}, is the disparity left and right consistency check. In order to produce a more accurate disparity map, Godard trains the network to predict left and right image differences and takes only the left view as input to the network. In order to ensure consistency, Godard introduces the L_1 differential consistency as part of the model. The cost function attempts to equalize the left parallax view to the projected right parallax view.

$$C^l_{lr} = \frac{1}{N}\sum_{ij} \left| d^l_{ij} - d^r_{ij+d^l_{ij}} \right|, \tag{5}$$

where d^l_{ij} is the left camera disparity value at position (i, j) and $d^r_{ij+d^l_{ij}}$ is the projected disparity value of the right camera.

To improve depth accuracy, Ma and Karaman [18] propose a different new model for ordinary supervised learning [19] with sparse depth samples and RGB images. Works in [18] use the residual sampling module of Laina *et al.* [19] to learn more features and, in return, improve the prediction accuracy and output resolution.

III. SPARSE OPTIMIZATION UNDER WEAK SUPERVISION

A. MODEL OPTIMIZATION

The left-right consistency provides a promising solution to unsupervised depth learning. Despite its advantages, one critical drawback of this method is the relatively low accuracy in depth estimation. The employment of accurate depth image, even a sparse one, is shown in [18] to be able to improve the accuracy. Motivated by the left-right consistency and training by sparse depth image, in this paper, we propose to use sparse LiDAR data to learn an accurate depth estimation network. We show the framework of our proposed model in Fig. 1.

Of each input raw color image, we first read its corresponding LiDAR data and transfer the very accurate depth values from the world coordinate to the image coordinate, or equivalently, we obtain the ground truth sparse depth image. We then train the left/right image along with their sparse depth image in our network and use the trained network to predict the depth values of any new monocular image. We show in Fig. 2 the input and output of our method where we take the raw RGB image and the corresponding sparse LiDAR data as input for training, and the trained network outputs the depth estimation in return.

Given a sparse LiDAR image, we need to employ those sparse depth samples in training, or equivalently to carefully design the loss function to fit depth samples. Since the LiDAR point cloud data is sparse, as shown in Fig. 2 (b), in this paper, only the pixels with valid depth values are used to calculate the loss function. We propose a new depth loss term, as shown in Eq. (6).

$$C_s = \alpha_{ap}(C^l_{ap} + C^r_{ap}) + \alpha_{ds}(C^l_{ds} + C^r_{ds}) + \alpha_{lr}(C^l_{lr} + C^r_{lr}) + \alpha_{dp}C_{dp}, \tag{6}$$

Compared with previous works, the most significant change of our loss function in Eq. (6) is the last term C_{dp}, the LiDAR loss, which follows

$$C_{dp} = \sum_{i=1}^{n} L(y^*_i, y_i), \tag{7}$$

where y^* is the ground truth depth value of the i-th pixel, y is the estimated depth and $L(y^*_i, y_i)$ is a loss function describing the dissimilarity between y^* and y. We will discuss on details of $L(\cdot, \cdot)$ in Sec. III-B. Please notice that in Eq. (7), we only sum up over pixels with valid LiDAR data, i.e., symbol n in Eq. (7) is the number of pixels with LiDAR data, which is much smaller than the total size of an image due to the sparse LiDAR image we used in our method.

To employ LiDAR ground truth in our training, we need first to map points generated by laser sensors (thus coordinated by the world coordinate system) to the image coordinate. It is well studied that given a certain point with world coordinates $[x, y, z]^T$, the corresponding image

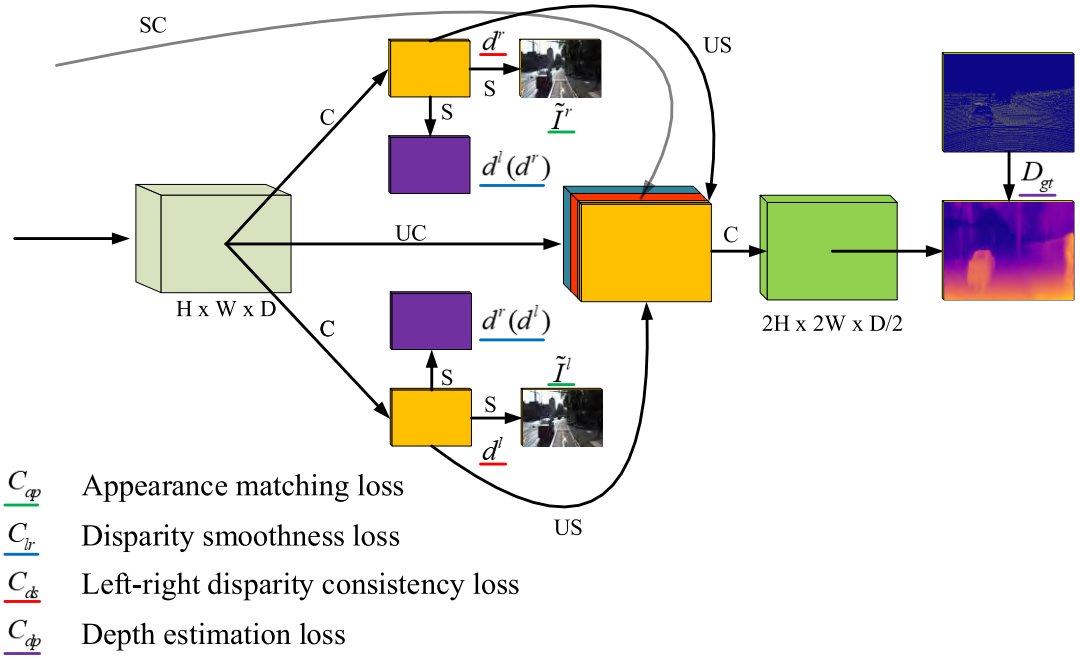

FIGURE 1. Framework of the proposed monocular depth estimation with Left-Right consistency and LiDAR weak supervision. C: Convolution, UC: Up-Convolution, S: Bilinear Sampling, US: Up-Sampling and SC: Skip Connection.

coordinates $[u, v]^T$ follow

$$y^* \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{8}$$

where K is the camera internal matrix.

B. LOSS FUNCTION FOR LIDAR DATA

The loss function is used to measure the degree of inconsistency between the predicted value y and the real value y^*. It is a non-negative real-valued function and to indicate the robustness of a model where a small value always indicates an accurate estimation.

There are alternative common loss functions that perform as promising candidates: the square loss function and the absolute loss function. The absolute loss function is defined by

$$L(y^*, y) = |y^* - y| \tag{9}$$

and the square loss function is

$$L(y^*, y) = (y^* - y)^2 \tag{10}$$

Other than the choice of the loss function, another essential parameter of our method is the LiDAR weight α_{dp}. In our paper, we test the alternative loss functions, i.e., the square loss function and the absolute loss function on one benchmark dataset and experimentally determine both the loss function fashion and its weight (see Tab. 1 and Tab. 2 in Sec. IV-C).

IV. EXPERIMENTAL RESULTS AND ANALYSIS

The network of this article is implemented using TensorFlow [20] and contains 31 million training parameters. Using two Titan X GPUs, it takes about 12 hours to train on

TABLE 1. Comparison of absolute loss function and square loss function.

	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Godard [17]	0.1245	1.4236	6.166	0.217	0.843	0.936	0.974
Absolute Loss Function	0.1335	1.4784	7.369	0.272	0.797	0.903	0.952
Square Loss Function	**0.1216**	**1.3511**	**6.064**	**0.214**	**0.844**	**0.938**	**0.975**

TABLE 2. Comparison of experimental results with different depths estimation loss term weights.

	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
$\alpha_{dp} = 0.1$	**0.1189**	**1.2714**	**6.105**	**0.213**	**0.846**	**0.940**	**0.975**
$\alpha_{dp} = 1$	0.1231	1.3208	6.195	0.219	0.838	0.935	0.973
$\alpha_{dp} = 10$	0.1249	1.4255	6.112	0.218	0.844	0.937	0.974

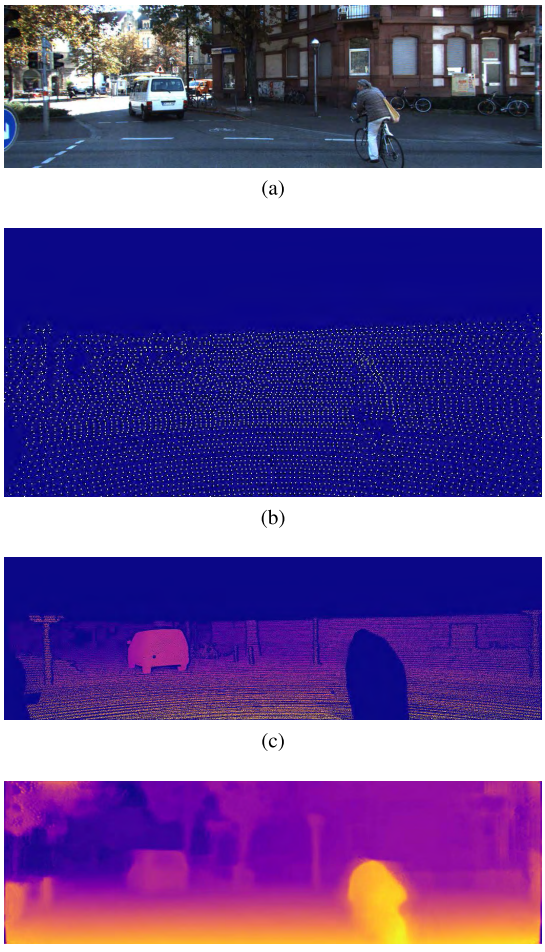

FIGURE 2. Demonstration of our input and output. (a) Raw image, (b) Sparse LiDAR, (c) Ground truth depth and (d) Our depth estimation. (a) Raw image. (b) Sparse LiDAR. (c) Ground truth depth. (d) Our depth estimation.

a 300,000 image data set (KITTI), which lasts for 50 epochs. For a 512 × 256 image, the proposed neural network completes depth estimation in less than 35 milliseconds per frame or 28 frames per second.

A. EXPERIMENT SETTINGS

In the optimization process, the weights of the first and second term of the loss function are set to $\alpha_{ap} = 1$ and $\alpha_{lr} = 1$, respectively. The disparity map output by the neural network is limited to be between 0 and d_{max} where d_{max} is 0.3× the size of the given output image width. Since the employment of multi-scale output in our model, the disparity between adjacent pixels will differ by a factor of 2 between each scale pair because of the upsampling. In order to remove errors introduced by upsampling, the disparity smoothing terms α_{ds} and scale r for each scale are scaled, in the purpose to get smooth consistency among all levels. Thus, $\alpha_{ds} = 0.1/r$. For the depth-estimated loss item α_{dp}, we set $\alpha_{dp} = 0.1$ according to our experimental results which are not shown in this paper.

For the choice of nonlinear part, we use exponential linear units [21], instead of the commonly used modified linear units (ReLU) [16], in our network. We have found through experiments that ReLUs tend to prematurely fix the intermediate-scale prediction parallax to a single value, leading to a narrow potential gap for further improvement by other fine-tuning technologies. We replace the usual deconvolution with the nearest neighbor sampling sum convolution [25].

As to training settings, we adopt Adam [26] to train the neural network from scratch and cycle through 50 epochs, where the size of each batch is 8, with Adam parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\varepsilon = 10^{-8}$. The initial learning rate is $\lambda = 10^{-4}$ and is kept constant in the first 30 epochs. Then, the learning rate becomes half of its previous value every 10 epochs until the end.

We first test a gradual update schedule, as described by Mayer *et al.* [28], in which the lower resolution image scale is optimized. However, we find that optimizing four scales, not the solo scale, can achieve more stable convergence. Similar to [28], we use the same weights to measure the loss of each scale because it is shown in many works that the unequal weights may lead to a unstable convergence. In our work, we test on batch standardization [27] but fail to find any significant improvements introduced by the batch standardization. Data enhancements are performed at the same time when the data are read.

B. EVALUATION METRICS

In this paper, we adopt four common evaluation metrics, absolute relative difference, square relative difference, root mean squared error and accuracy with a certain threshold, to verify the depth accuracy of each competing methods. The definition of each metric are listed in the followings. In our metric definition, y stands for an estimated depth value of one pixel and y^* is the ground truth value, T is the number of all pixels.

Absolute relative difference (**Abs Rel**) is defined by

$$M_{AbsRel} = \frac{1}{|T|}\sum_{y\in T}\frac{|y-y^*|}{y^*} \tag{11}$$

Square relative difference (**Sq Rel**) is

$$M_{SqRel} = \frac{1}{|T|}\sum_{y\in T}\frac{|y-y^*|^2}{y^*} \tag{12}$$

Root mean squared error (**RMSE**) is

$$M_{RMSE} = \sqrt{\frac{1}{|T|}\sum_{y\in T}|y-y^*|^2} \tag{13}$$

As to Accuracy, we first define δ as depth error ratio of one pixel, $\delta = \max(\frac{y^*}{y}, \frac{y}{y^*})$. Then, Accuracy is the number of pixels with corresponding δ less than a threshold.

C. PARAMETER FINE-TUNING

In this section, we fine tune parameters of our method on KITTI dataset. We test on three critical parameters in this

TABLE 3. Input image resolution for each scale (units: pixels).

Scale	Image Resolution
1	512×256
2	256×128
3	128×64
4	64×32

section, including the loss function (absolute vs. square), the loss term weight α_{dp} and the input image downsampling scale factor.

In the first test, we compare the absolute loss function with the square loss function, as defined in Sec. III-B. Experimental results are shown in Tab. 1. In Tab. 1, we compare the alternative loss functions on KITTI dataset and, for a better comprehension, we also list the performance of the network in [17], which is considered as the benchmark method to compare with.

As can be seen from Tab. 1, the model using the square loss function obtains better performance in all metrics. Compared with the benchmark method, the absolute value loss function has a negative effect on the performance of the original model. The square loss function is more sensitive to outliers. When there are many outliers, the square loss function will increase the penalty value and, as a result, accelerate the convergence to the minimum value. In order to speed up the convergence of the model and improve the prediction accuracy of the model, in our following experiments, the square loss function is used as the loss function of the depth estimation loss term C_{dp}.

The LiDAR loss weight α_{dp} is another critical parameter in our method. The depth estimation loss term introduces the LiDAR point cloud depth value as the ground truth into the loss function, adding a weak supervision signal to the original model, and guiding the predicted depth value to approach ground truth. A proper α_{dp} will benefit both convergence speed and convergence direction. In order to verify the influence of this weight on the convergence of the model, in the second experiment, we test on several common choices of α_{dp} to experimentally set a fixed value of α_{dp}. In this experiment, we test with $\alpha_{dp} = 10^{-1,0,1}$, as shown in Tab. 2.

In Tab. 2 we can see that the optimal α_{dp} occurs at $\alpha_{dp} = 0.1$. Although the LiDAR ground truth provides a weak supervision to learning, the overall loss function still prefers to use the traditional loss terms with larger weight and the new LiDAR term with relatively low value.

Typically, there are two parts in our network: an encoder and an decoder. The decoder uses a skip connection from the encoder activation block to enable it to parse higher resolution image details. There are four different scales (disp4 to disp1) in [17] to output the parallax prediction, and the spatial resolution on each subsequent scale is doubled, that is, the size of the image input to each scale is the scale of the previous scale. In this paper, we verify the use of multiple scales, as shown in Tab. 3, for depth estimation. The corresponding experimental results are shown in Tab. 4. In Tab. 4, the term 'scale *i*' means we employ the leading *i* scales in the network. Thus, the first row in Tab. 4 refers to the simplest network in this test with only one scale as input to our network and, in contrast, the last row means the raw image is downsampled in all four scales and we push all four downsampled (except the very first raw input) to our network.

As shown in Tab. 4, there is not an overwhelming winner in terms all seven metrics. Promising candidates come from scale 1 and scale 4 which claims three and four champions respectively in our test. Considering the slight superiority of scale 4, or the employment of all four downsampled images, over its counterpart, in this paper, we use scale 4 in our method.

D. COMPARISON WITH STATE-OF-THE-ARTS

In this section, we compare our network with several state-of-the-art methods to verify the superiority of our method. The competing methods consist of works in [10] and [29] and [17]. In [29], there are two kinds of networks, entitled coarse and fine respectively. As to [17], we implement [17] on both VGG and ResNet. Thus, in summary there are in a total of five competing methods in this test, as shown in Tab. 5. We also show in Fig. 3 several demonstrations of depth estimation by all competing methods.

According to Tab. 5 we can observe the followings:

1) Depth estimation by left-right consistency performs slightly better than the traditional estimation methods, such as works in [10] and [29]. Left-right-consistency-based methods, i.e., works in [17] and ours, show relatively better results compared with the non-consistency methods. In Tab. 5, our method shows a significant improvement over the coarse network in [10], e.g., 48.5% lower in terms of absolute relative difference.

2) ResNet model can improve the performance compared with VGG. Since the number of ResNet layers is higher than VGG, the network is able to handle complicated hidden structure of depth mapping, local information and image features. In Tab. 5, the replacement of VGG with ResNet offers an 8.8% increase in terms of RM SE of our method and 4.9% of Godard *et al.* [17].

TABLE 4. Comparison of experimental results with different depths estimation loss term scale.

	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
scale 1	0.1214	1.3417	**6.028**	0.212	0.845	**0.940**	**0.976**
scale 2	0.1205	1.3098	6.156	0.215	0.843	0.937	0.974
scale 3	0.1217	1.3375	6.106	0.214	**0.846**	0.938	0.974
scale 4	**0.1189**	**1.2714**	6.105	0.213	**0.846**	**0.940**	0.975

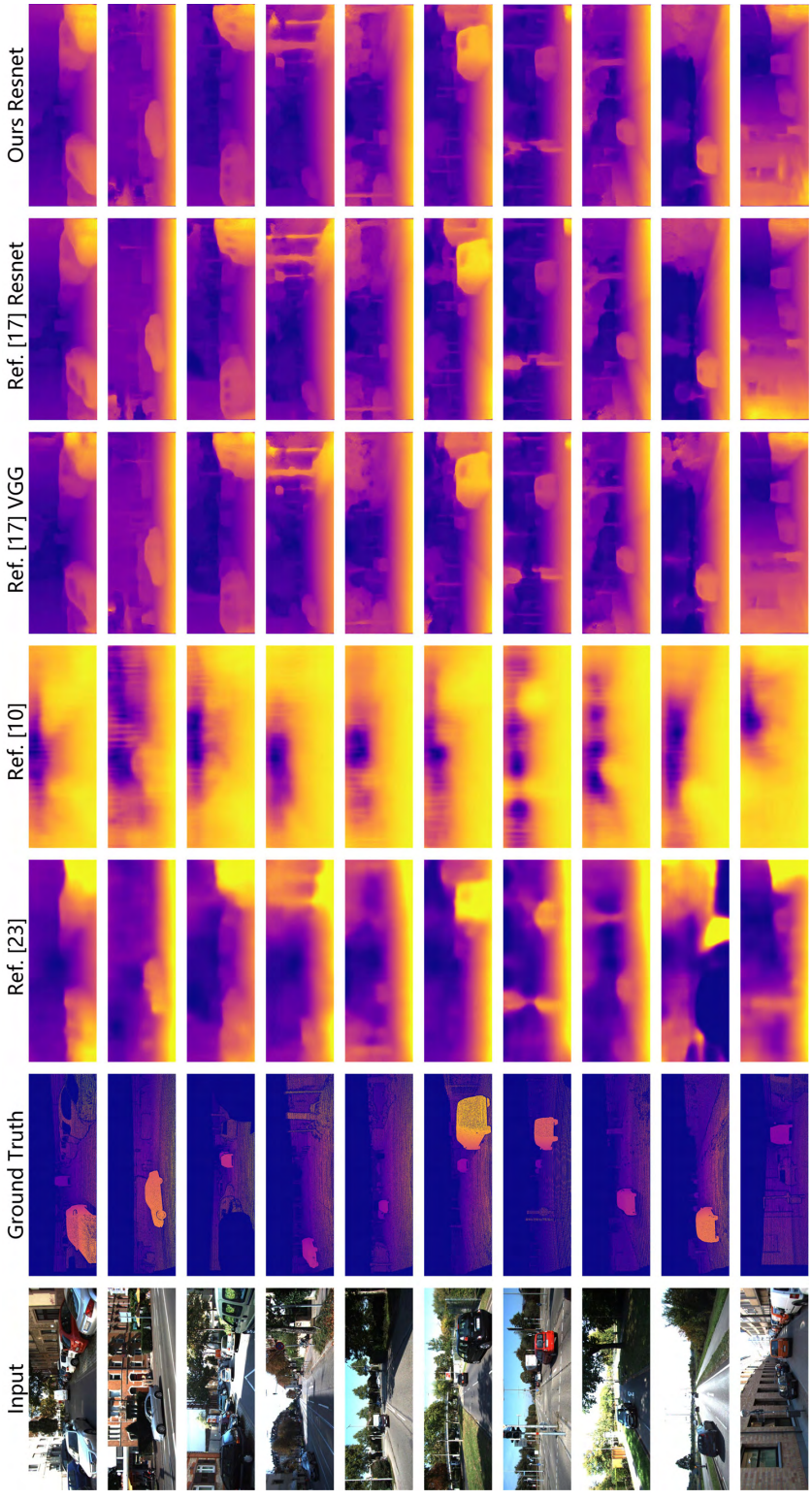

FIGURE 3. Our monocular depth estimation results.

TABLE 5. Comparison of Experimental Results of Two Neural Network Structures

	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Ref. [10] Coarse	0.214	1.605	6.563	0.292	0.673	0.884	0.957
Ref. [10] Fine	0.203	1.548	6.307	0.282	0.702	0.89	0.958
Ref. [29]	0.208	1.768	6.856	0.283	0.678	0.885	0.957
Ref. [17] VGG	0.1245	1.4236	6.166	0.217	0.843	0.936	0.974
Ours VGG	0.1225	1.3164	6.169	0.216	0.841	0.937	0.974
Ref. [17] ResNet	0.1131	1.1883	5.862	0.205	0.850	0.944	0.978
Ours ResNet	**0.1103**	**1.0853**	**5.628**	**0.199**	**0.855**	**0.949**	**0.981**

3) Using LiDAR as weak supervision improves depth estimation. Compared with its rivals, the proposed LiDAR-weak-supervision on ResNet reaches the best performance in terms of all metrics. The involvement of LiDAR data, even in a sparse fashion, is able to improve depth estimation. E.g., in terms of square relative difference, the use of LiDAR data obtains an increase of 8.7% compared with its non-LiDAR counterpart.

V. CONCLUSION

In this paper, we propose a monocular depth estimation method which is suitable for wearable devices. We train on ResNet with sparse ground truth values coming from LiDAR data and improve the depth estimation accuracy. In our training scheme, we construct our expected outputs in two fashions. First, we adopt the left-right consistency model and take the right camera images as the expected outputs under the assumption that with an accurate depth estimation, we should be able to re-construct images of the right camera from that of the left one. Second, we use laser radar to directly obtain the real depth of several individual points and take the LiDAR depth as the ground truth values to compare with. We propose a new cost function in our network to combine both camera-consistency and LiDAR ground truth in training. We compare the proposed method with several state-of-the-art depth estimation methods and verify the superiority of our method. Our proposed network can be implemented on a light-weight device which requires very limited additional burden to users and, as a result, is wearable-friendly in general.

REFERENCES

[1] M. Ur-Rehman, Q. H. Abbasi, M. Akram, and C. Parini, "Design of band-notched ultra wideband antenna for indoor and wearable wireless communications," *IET Microw., Antennas Propag.*, vol. 9, no. 3, pp. 243–251, 2015.

[2] S. Wan, Y. Zhang, and J. Chen, "On the construction of data aggregation tree with maximizing lifetime in large-scale wireless sensor networks," *IEEE Sensors J.*, vol. 16, no. 20, pp. 7433–7440, Oct. 2016.

[3] B. Mi, D. Huang, S. Wan, L. Mi, and J. Cao, "Oblivious transfer based on NTRUEncrypt," *IEEE Access*, vol. 6, pp. 35283–35291, 2018, doi: 10.1109/ACCESS.2018.2846798.

[4] N. Dawar and N. Kehtarnavaz, "Real-time continuous detection and recognition of subject-specific smart TV gestures via fusion of depth and inertial sensing," *IEEE Access*, vol. 6, pp. 7019–7028, 2018.

[5] L. He and H. Zhang, "Iterative ensemble normalized cuts," *Pattern Recognit.*, vol. 52, pp. 274–286, Apr. 2016.

[6] X. Yang *et al.*, "Reverse recognition of body postures using on-body radio channel characteristics," *IET Microw., Antennas Propag.*, vol. 11, no. 9, pp. 1212–1217, 2017.

[7] J. J. Koenderink and A. J. van Doorn, "Affine structure from motion," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 8, no. 2, pp. 377–385, 1991.

[8] A. Saxena, M. Sun, and A. Y. Ng, "Make3D: Learning 3D scene structure from a single still image," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 824–840, May 2009.

[9] M. Liu, M. Salzmann, and X. He, "Discrete-continuous depth estimation from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 716–723.

[10] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2366–2374.

[11] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 2650–2658.

[12] F. Liu, C. Shen, G. Lin, and I. D. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2024–2039, Oct. 2016.

[13] R. Garg, B. G. Vijay Kumar, G. Carneiro, and I. Reid, "Unsupervised CNN for single view depth estimation: Geometry to the rescue," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 740–756.

[14] Y. Cao, Z. Wu, and C. Shen, "Estimating depth from monocular images as classification using deep fully convolutional residual networks," *IEEE Trans. Circuits Syst. Video Technol.*, to be published.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 770–778.

[16] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 807–814.

[17] C. Godard, O. M. Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6602–6611.

[18] F. Ma and S. Karaman. (2017). "Sparse-to-dense: Depth prediction from sparse depth samples and a single image." [Online]. Available: https://arxiv.org/abs/1709.07492

[19] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *Proc. 4th Int. Conf. 3D Vis. (3DV)*, 2016, pp. 239–248.

[20] M. Abadi *et al.* (2016). "TensorFlow: Large-scale machine learning on heterogeneous distributed systems." [Online]. Available: https://arxiv.org/abs/1603.04467

[21] D.-A. Clevert, T. Unterthiner, and S. Hochreiter. (2015). "Fast and accurate deep network learning by exponential linear units (ELUs)." [Online]. Available: https://arxiv.org/abs/1511.07289

[22] M. Jaderberg, K. Simonyan, and A. Zisserman, "Spatial transformer networks," in *Proc. NIPS*, 2015, pp. 2017–2025.

[23] J. Xie, R. Girshick, and A. Farhadi. (2016). "Deep3D: Fully automatic 2D-to-3D video conversion with deep convolutional neural networks." [Online]. Available: https://arxiv.org/abs/1604.03650

[24] P. Heise, S. Klose, B. Jensen, and A. Knoll, "PM-Huber: PatchMatch with huber regularization for stereo matching," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 2360–2367.

[25] A. Odena, V. Dumoulin, and C. Olah, "Deconvolution and checkerboard artifacts," *Distill*, vol. 1, no. 10, p. e3, 2016.

[26] D. P. Kingma and J. Ba. (2014). "Adam: A method for stochastic optimization." [Online]. Available: https://arxiv.org/abs/1412.6980

[27] S. Ioffe and C. Szegedy. (2015). "Batch normalization: Accelerating deep network training by reducing internal covariate shift." [Online]. Available: https://arxiv.org/abs/1502.03167

[28] N. Mayer *et al.*, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4040–4048.
[29] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6612–6619.

LI HE (M'16) received the B.Sc., M.Sc., and Ph.D. degrees from the Department of Automation, Northwestern Polytechnical University, Xi'an, China, in 2006, 2009, and 2014, respectively. He was a Visiting Student from 2010 to 2011 and then served as a Post-Doctoral Fellow with the Department of Computing Science, University of Alberta, from 2014 to 2017. He is currently an Assistant Professor with the School of Electromechanical Engineering, Guangdong University of Technology, Guangzhou, China. He has more than 20 peer-reviewed publications on venues, such as TCYB, TIP, PR, and IROS. His current research interests include machine learning, robotics, and computer vision.

CHUANGBIN CHEN received the B.Sc. degree (Hons.) from the School of Electromechanical Engineering, Guangdong University of Technology, Guangzhou, China, where he is currently pursuing the master's degree with the School of Electromechanical Engineering. His research interests include computer vision and deep learning.

TAO ZHANG received the B.Eng. degree in aircraft manufacturing engineering from Beihang University in 2008 and the Ph.D. degree in mechanical engineering from the Inbotics Institute, Beihang University, in 2017. He has been an Assistant Professor with the School of Electromechanical Engineering, Guangdong University of Technology, since 2017. His interests include robotics and electromechanical systems.

HAIFEI ZHU received the bachelor's degree in mechanical engineering from the Wuhan University of Technology, Wuhan, China, in 2008, and the Ph.D. degree in mechanical engineering from the South China University of Technology, Guangzhou, China, in 2013. He was a Research Fellow with the School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore. He is currently an Assistant Professor with the School of Electromechanical Engineering, Guangdong University of Technology, Guangzhou. His research interests include climbing robots, robotic manipulation, and motion/path planning.

SHAOHUA WAN received the joint Ph.D. degree from the School of Computer, Wuhan University, and the Department of Electrical Engineering and Computer Science, Northwestern University, USA, in 2010. Since 2015, he has been holding a post-doctoral position at the State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology. From 2016 to 2017, he was a Visiting Scholar with the Department of Electrical and Computer Engineering, Technical University of Munich, Germany. He is currently an Associate Professor and a Master Advisor with the School of Information and Safety Engineering, Zhongnan University of Economics and Law. His main research interests include massive data computing for Internet of Things and edge computing.