Toward Hierarchical Self-Supervised Monocular Absolute Depth Estimation for Autonomous Driving Applications

Feng Xue[1], Guirong Zhuo[1,*], Ziyuan Huang[2], Wufei Fu[1], Zhuoyue Wu[1] and Marcelo H. Ang Jr[2]

Abstract— In recent years, self-supervised methods for monocular depth estimation has rapidly become an significant branch of depth estimation task, especially for autonomous driving applications. Despite the high overall precision achieved, current methods still suffer from a) imprecise object-level depth inference and b) uncertain scale factor. The former problem would cause texture copy or provide inaccurate object boundary, and the latter would require current methods to have an additional sensor like LiDAR to provide depth ground-truth or stereo camera as additional training inputs, which makes them difficult to implement. In this work, we propose to address these two problems together by introducing DNet. Our contributions are twofold: a) a novel dense connected prediction (DCP) layer is proposed to provide better object-level depth estimation and b) specifically for autonomous driving scenarios, dense geometrical constrains (DGC) is introduced so that precise scale factor can be recovered without additional cost for autonomous vehicles. Extensive experiments have been conducted and, both DCP layer and DGC module are proved to be effectively solving the aforementioned problems respectively. Thanks to DCP layer, object boundary can now be better distinguished in the depth map and the depth is more continues on object level. It is also demonstrated that the performance of using DGC to perform scale recovery is comparable to that using ground-truth information, when the camera height is given and the ground point takes up more than 1.03% of the pixels. Code is available at https://github.com/TJ-IPLab/DNet.

I. INTRODUCTION

Estimating an accurate depth map from single RGB image is of great significance in 3D scene understanding as well as in many real-world applications such as augmented reality and autonomous driving. Compared to traditional hand-crafted feature-based methods [1], supervised [2], [3], [4], [5], [6], [7] and stereo self-supervised [8], [9], [10], [11] learning has been proved to be able to achieve better performance on this task. Unfortunately, these methods either require a large amount of high-quality annotated ground-truth, which is difficult to obtain, or need complex stereo calibration. Therefore, monocular self-supervised learning methods became the focus of research. Some recent works [12], [13], [14], [15] revealed its great potential to tackle monocular depth estimation task.

Despite its potential to reach satisfying performances, current methods have two shortcomings. One of them is that they are only able to estimate relative depth rather than the absolute one. For evaluation, scale factor is calculated by

[1]Feng Xue, Guirong Zhuo, Wufei Fu and Zhuoyue Wu are with the School of Automotive Studies, Tongji University, 201804 Shanghai, China zhuoguirong@tongji.edu.cn

[2]Ziyuan Huang and Marcelo H. Ang Jr are with Department of Mechanical Engineering, National University of Singapore, Singapore mpeangh@nus.edu.sg
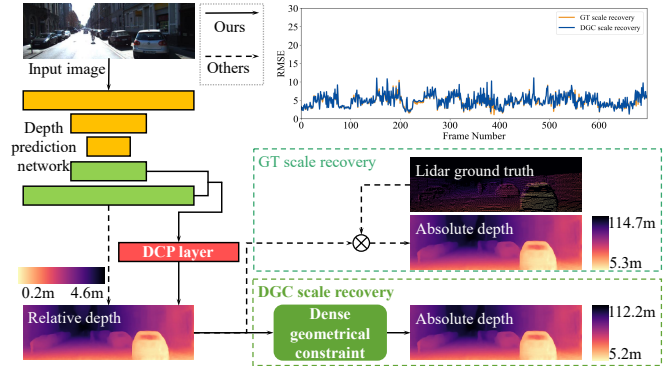
Fig. 1. Structure difference of DNet with other self-supervised monocular depth estimation methods. Solid lines indicates our work flow and dotted lines are that of other methods. Dense connected prediction (DCP) layer is introduced to generate hierarchical features for better object-level depth inference, and dense geometrical constraint (DGC) is introduced to directly estimate absolute depth from monocular images. Performance comparison of *DGC* and ground-truth based scale recovery is indicated in the top-right plot.

ratio between the medians of ground-truth (given by LiDAR) and predicted depth [12], [13], [14], [15], as can be seen from Fig. 1. Theoretically, it is a decent solution. However, for practical uses, obtaining ground-truth in real applications using other sensors not only raises the cost, it also complexes the system, leading to complicated joint calibration processes and synchronization problems.

Another problem is that because the decoder of current methods predict depth in different resolutions separately, some details on object-level is omitted. For example, object boundary can be blurred and the depth of texture on the object may be predicted differently than the object itself.

In this paper, we propose DNet, a novel self-supervised monocular depth estimation pipeline that exploits densely connected hierarchical features to obtain more precise object-level depth inference, and uses dense geometrical constraint to eliminate the dependence on additional sensors or depth ground-truth to perform scale recovery, so that it is easier to be brought into practical use.

Our contributions are listed as follows:

- We improve the former multi-scale estimation strategy by proposing a novel dense connected prediction (DCP) layer. Instead of predicting depth and computing reconstruction loss separately under different scales, the proposed DCP layer exploits hierarchical feature so that object-level depth inference can be made based on multi-scale prediction features, refining object boundary

and reducing visual artifacts.

- A novel dense geometrical constraints (DGC) module is introduced to perform high-quality scale recovery for autonomous driving. Based on relative depth estimation, DGC module can finish per-pixel ground segmentation and estimate a camera height from every ground point. Statistical method is applied to determine the camera height so that outliers of ground point extraction can be robustly suppressed. Scale factor can be determined through comparison between the given and estimated camera height.

- DNet is extensively evaluated on KITTI[16] Eigen Split [2], where the results not only showed the capability of DCP layer to improve the performance of object-level depth inference, but also proved that DNet using DGC module has competitive performance against those methods using depth ground-truth to determine scale factor. Ablation studies demonstrated module effectiveness as well as sensitivity of DNet to ground points ratio.

II. RELATED WORKS

A. Self-supervised monocular depth estimation

Monocular depth estimation has always been an important aspect of scene understanding. Some works apply supervised [2], [3], [4], [5], [6], [7] or stereo self-supervised [8], [9], [10], [11] methods to tackle the problem. However, due to the difficulty of obtaining large amount of labeled data or complex stereo calibration to train the depth estimation network, monocular self-supervised method was proposed instead [12], [13], [14], [15].

Proposed by the pioneering work [15], the basic idea is to use photometric reconstruction loss calculated by comparing the target image with the target view reconstructed from nearby source views. However, it assumes that the scene is static and that no occlusion is present between different consecutive frames. [17], [18], [19] explicitly established different motion models to resolve the moving scene problem. [20] introduced 3D surface normal by constructing two additional layers for better depth estimation. [14] replaced the original photometric reconstruction error with per-pixel minimum reprojection error, which partially enabled it to tackle occlusion. It also used up-sampling and proposed auto-masking of stationary pixels to avoid 'holes' of infinite depth generated by low-texture and moving objects respectively.

However, all aforementioned works predict only relative depth, which means there still exists a scale gap between the prediction and true depth. For evaluation purpose, ratio between medians of ground-truth and current prediction is employed to acquire absolute depth. Unfortunately, in real application scenarios, ground-truth is either too difficult or financially expensive to obtain. Therefore, a scale recovering approach which is free of depth ground truth is called for.

B. Monocular scale recovery

Scale uncertainty has always been a problem for 3D vision for monocular camera. To recover scale factor and achieve absolute depth estimation, [21] utilizes pose information and [22] uses stereo data to pretrain network, both introducing additional sensor information but the results were no as satisfactory. Besides depth estimation, a typical example of this is monocular visual SLAM. In order to mitigate this, [23], [24] integrate object detection algorithms into monocular visual SLAM system and take advantage of object size prior to recover scale. However, in addition to the significant increase of computation complexity, these methods show limited robustness under scenes without known object classes.

Handling the geometrical relationship between camera and ground is also an effective approach to tackle this problem. This geometrical constrain is broadly used in autonomous driving tasks, for ground is commonly seen in images captured by on-board cameras. The main task of these methods is to estimate a relative camera height using camera-ground geometrical constrains, and thus infer scale with absolute camera height prior. [25] extracted ground using trained classifier, but it doesn't possess an excellent generalization power. [26] extracted the ground points densely in region of interest similar to [27], [28], but it requires dense stereo to be added to the system, which can potentially raise cost and increase complexity. In [29], the most similar work to this one, used surface normal to extract ground points and thus calculate camera height. However, due to the sparsity originated in its key-point-based strategy, data association through consecutive frames are needed, makes this method hardly integrated into monocular depth estimation tasks which use only single image as input. In additon, this method regards the ground as a whole, flat panel with single surface normal, which is a strong assumption for autonomous driving scenarios. In contrast, our method is free of data association, which means it can be integrated into both monocular depth estimation and visual SLAM tasks. Furthermore, our method achieve per-pixel surface normal calculation and ground segmentation, makes the algorithm robust to different road conditions for autonomous driving.

III. METHOD

In this section, a novel pipeline called DNet specifically designed for monocular absolute depth estimation in autonomous driving applications is proposed. The pipeline can be divided into two parts, respectively relative depth estimation, with dense connected prediction (DCP) layer to improve object-level depth inference, and scale recovery based on dense geometrical constraint, without needing any additional sensor signals or depth ground-truth. The overview of DNet can be seen in Fig. 2.

A. Relative depth estimation

The proposed DNet is based on Monodepth2 [14]. As all self-supervised depth estimation methods, its object-level inference can still have texture copy and imprecise object boundaries. In this section, we will first introduce Monodepth2 and then resolve this issue by introducing DCP layer to replace full resolution module used in Monodepth2.
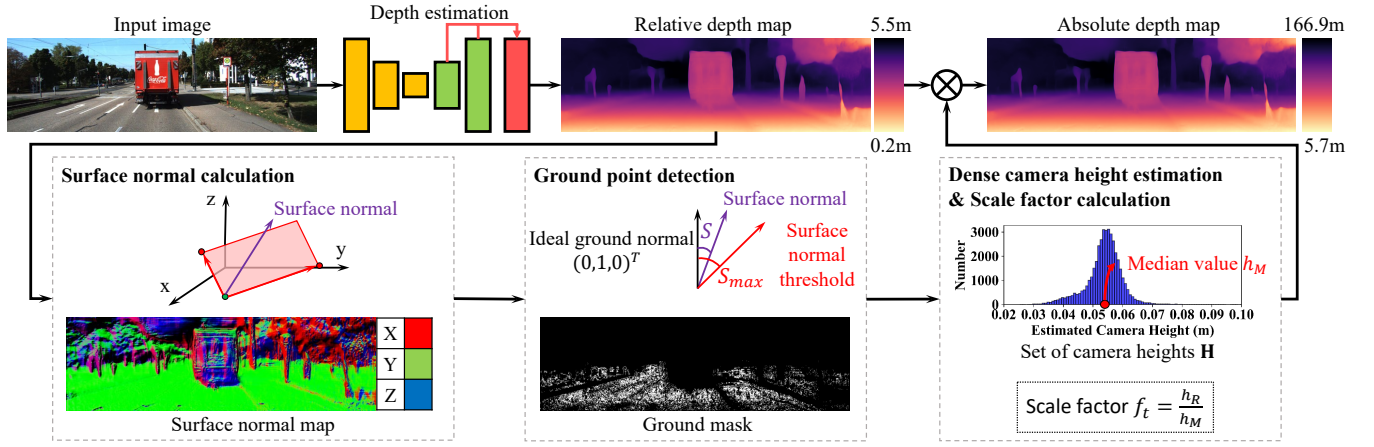
Fig. 2. Overall structure of proposed DNet pipeline. The algorithm first estimates relative depth using the depth estimation network with our proposed DCP layer (pink layer in the figure) to generate and exploit hierarchical features. After that, scale recovery module pops in. It utilizes geometrical relationship between the ground and the camera, using extracted ground points to densely calculate camera heights point by point. Median value of all camera heights is then selected to be the final estimated value and used to obtain scale factor. Combined with relative depth map generated in the first step, scale recovery module outputs absolute depth of the given monocular image.

1) Baseline: Monodepth2 w/o full resolution: **Architecture:** Two networks are required in monocular self-supervision architecture, respectively a depth network and a pose network. Single image I_t of the t-th frame is taken as the input of the depth network. Depth network outputs a dense relative depth map $\mathbf{D}_t^{rel}$. Pose network takes $\{\mathbf{I}_{t-1}, \mathbf{I}_t\}$ and $\{\mathbf{I}_t, \mathbf{I}_{t+1}\}$ sequentially as inputs and then outputs camera poses of the t-th image relative to that of the $(t-1)$-th and $(t+1)$-th images, i.e., $\{\mathbf{T}_{t \to t-1}^{rel}, \mathbf{T}_{t \to t+1}^{rel}\}$.

Self-supervision loss: Two parts constitute the overall loss, respectively per-pixel minimum reconstruction loss L_p and inverse depth smoothness loss L_s. Reconstruction loss is calculated by firstly inverse warping source images $\{\mathbf{I}_{t-1}, \mathbf{I}_{t+1}\}$ to rebuild two target images $\{\mathbf{I}_{t-1 \to t}, \mathbf{I}_{t+1 \to t}\}$. After that, photometric error (PE) between reconstructed image and target image is calculated combining structural similarity index (SSIM) [30] and L1 norm between two images $\mathbf{I}_a, \mathbf{I}_b$ as follows:

$$\text{PE}(\mathbf{I}_a, \mathbf{I}_b) = \alpha \frac{1 - \text{SSIM}(\mathbf{I}_a, \mathbf{I}_b)}{2} + (1-\alpha)\|\mathbf{I}_a - \mathbf{I}_b\|_1 , \quad (1)$$

where α is used for weight adjustment.

Per-pixel minimum loss $\mathbf{L}_p$ is then calculated as follows:

$$\mathbf{L}_p = \min_{\mathbf{I}'} \left(PE(\mathbf{I}', \mathbf{I}_t) \right) , \quad (2)$$

where $\mathbf{I}' \in \{\mathbf{I}_{t-1 \to t}, \mathbf{I}_{t+1 \to t}, \mathbf{I}_{t-1}, \mathbf{I}_{t+1}\}$.

Combined with edge-aware smoothness loss $\mathbf{L}_s$:

$$\mathbf{L}_s = |\partial_x d_t^*| e^{-|\partial_x \mathbf{I}_t|} + |\partial_y d_t^*| e^{-|\partial_y \mathbf{I}_t|} , \quad (3)$$

where $d_t^* = d_t / \bar{d}_t$ is the mean-normalized inverse depth, overall loss can be constructed with two hyper-parameters μ and λ as:

$$\mathbf{L}_i = \sum_i (\mu \mathbf{L}_{p,i} + \lambda w_i \mathbf{L}_{s,i}) , \quad (4)$$

where subscript i denotes different resolution layers of the decoder. w_i is determined according to the resolution.

2) DNet & Dense connected prediction layer: **Overall loss:** Because the photometric error of low resolution depth prediction can be the result of wrong network prediction or the aliasing of down-sampling, using the same weight in loss for low-res and high-res results can mislead the network to converge in non-optimal values. Additionally, in consideration that features with lower resolution are reused for multiple times, the weight of error in lower resolution depth prediction is reduced as follows:

$$\mathbf{L}_i = \sum_i (\mu v_i \mathbf{L}_{p,i} + \lambda w_i \mathbf{L}_{s,i}) . \quad (5)$$

where $v_i < 1$ is introduced as weight adjustment parameter.

DCP layer: In order to handle local gradient caused by bilinear sampling [31] and local minima, current works [12], [13], [14], [15] including our baseline Monodepth2 use multi-scale depth prediction strategy. This strategy implicitly uses low-res features to predict depth by repeated upsampling layers, which has the tendency of depth artifacts (Fig. 9). Motivated by reducing the depth artifacts and acquiring more reasonable object-level depth inference, we propose a novel DCP layer that explicitly combines features in different scales hierarchically. The intuition is based on the observation that low-res layers of decoder network can provide more reliable object-level depth inference and high-res layers focus more on local depth details.

Formally, the numbers of feature channels in different scales are reduced to eight using a convolutional layer in the DCP layer, so that the number of channels are uniformed and calculations afterwards can be simplified. Features in low-res layers are then up-sampled and concatenated to higher-res layer features. By doing this, we introduce more precise object-level inference into higher resolution depth predictions that originally care less about object-level depth. The final depth estimation is performed based on the hierarchical features provided by densely connected feature layers. Detailed structure can be seen in Fig. 3.
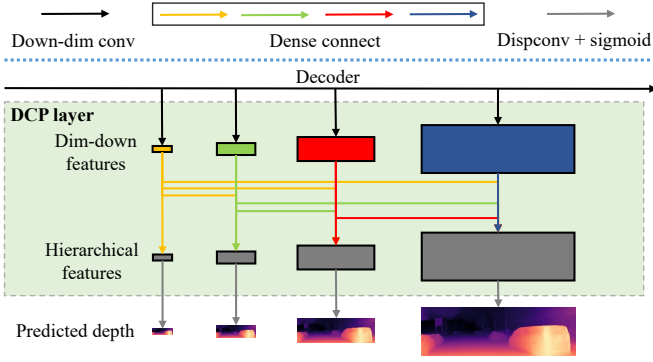
Fig. 3. Structure of proposed DCP layer. Different from baseline multi-scale prediction strategy (directly uses feature in different resolutions independently), features are densely connected from low to high resolution to form hierarchical features in DCP layer.

B. Scale recovery

Scale recovery is performed after relative depth is predicted so that absolute depth map can be generated solely relying on monocular image. Dense geometrical constraint (DGC) is thus introduced. DGC is specifically designed for autonomous driving applications. It works under the assumption that there are enough ground points in the monocular image, which is usually the case for autonomous driving. Unlike the scale recovery employed by feature-based visual odometry, ground points are densely extracted by DGC from the monocular images to form a dense ground point map. Each point in the map is used to estimate one camera height, as can be seen in Fig. 5. A large number of camera heights can thus be obtained. By applying statistical methods for overall camera height estimation, outliers can barely harm the estimation result of the scale factor.

1) Surface normal calculation: The first step is to determine a surface normal for each pixel in the input image. All the pixel points need to be projected to 3D space according to the following equation:

$$\mathbf{D}_t^{rel}(\mathbf{p}_{i,j})\mathbf{p}_{i,j} = \mathbf{K}\mathbf{P}_{i,j} \ , \tag{6}$$

where $\mathbf{p}_{i,j} = [i,j,1]^\top$ refers to the pixel on the i-th row and the j-th column in 2D space with one homogeneous coordinate, and $\mathbf{P}_{i,j} = [X,Y,Z]^\top$ is the corresponding 3D point, $\mathbf{D}_t^{rel}(\mathbf{p}_{i,j})$ is the depth of that specific point, and $\mathbf{K}$ is the camera intrinsic matrix.

Similar to [20], for each pixel point , 8-neighbor convention is used to determine several planes around it, as in Fig. 4. All 8 neighbors of $\mathbf{p}_{i,j}$ are grouped into 4 pairs. Two vectors of $\mathbf{p}_{i,j}$ connected respectively to two points in one pair form a 90-degree angle, i.e., $G(\mathbf{P}_{i,j}) = \{[\mathbf{P}_{i+1,j}, \mathbf{P}_{i,j-1}], [\mathbf{P}_{i+1,j-1}, \mathbf{P}_{i-1,j-1}]...\}$. Four pairs of vector constitutes 4 surfaces, thus generating 4 surface normals, which can be calculated by:

$$\mathbf{n}_g = \overrightarrow{\mathbf{P}_{i,j}G_{g,1}} \times \overrightarrow{\mathbf{P}_{i,j}G_{g,2}} \ , \tag{7}$$

where $G_{a,b}$ denotes the b-th element of the a-th pair in $G(\mathbf{P}_{i,j})$ and $g = 1,2,3,4$.
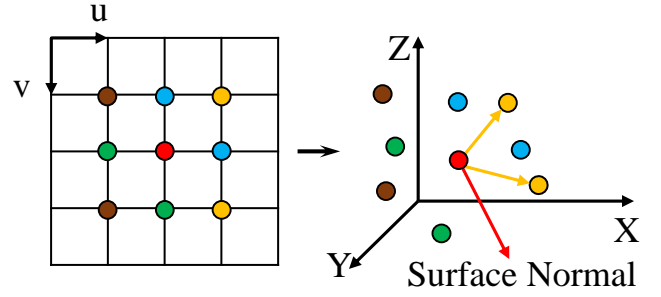

Fig. 4. 2D to 3D projection and pairing of 8-neighbors in surface normal calculation. Points with the same color is paired to form two vectors respectively with the center point. Four surface normals can be calculated from four vector pairs and used to form one surface normal.

The final normalized surface normal of point $\mathbf{P}_{i,j}$ is given by normalizing and averaging four estimated normals:

$$\mathbf{N}(\mathbf{P}_{i,j}) = \frac{\sum_g \mathbf{n}/\|\mathbf{n}_g\|_2}{4} \ . \tag{8}$$

2) Ground point detection: Ground points usually refers to the points that has a normalized normal close to ideal ground normal, i.e., $\tilde{\mathbf{n}} = (0,1,0)^\top$. With this ideal target normal and the calculated normalized surface normal, we propose a similarity function $s(\mathbf{P}_{i,j})$ based on absolute value of cosine function. The calculated similarity S can be used as a simple criteria to determine whether $\mathbf{P}_{i,j}$ is a ground point or not.
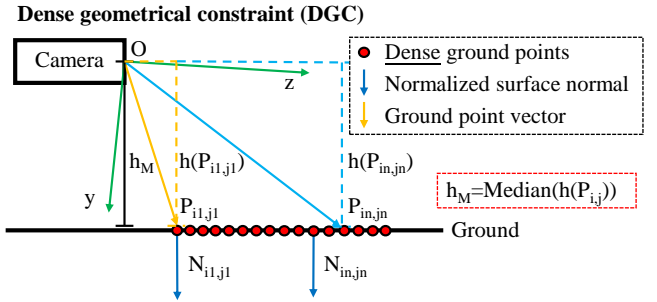

Fig. 5. Schematic for DGC. Different from common geometrical constraint, which outputs only one surface normal for all ground points, for each ground point in DGC, a surface normal vector is calculated. Each surface normal vector is used to estimate one camera height. The overall camera height is estimated through calculating the median of all estimated camera heights.

$$S = s(\mathbf{P}_{i,j}) = |\angle(\tilde{\mathbf{n}}, \mathbf{N}(\mathbf{P}_{i,j}))| = |arccos\frac{\tilde{\mathbf{n}} \cdot \mathbf{P}_{i,j}}{\|\tilde{\mathbf{n}}\|\|\mathbf{P}_{i,j}\|}| \ , \tag{9}$$

where operator $\cdot$ denotes the inner product operation.

Considering the uncertainty produced by estimating the surface normal and the y-axis of camera coordinate system is not strictly perpendicular to the ground as in Fig. 5, a threshold S_{max} is set. For $S < S_{max}$, the pixel point is considered as ground points. After determination for ground points has finished for all pixel points, a set of ground points $\mathbf{GP} = \{\mathbf{P}_{i,j}|s(\mathbf{P}_{i,j}) < S_{max}, y(\mathbf{P}_{i,j}) > 0\}$ is detected,

Method	Scale Factor	Lower is better				Higher is better		
		Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Zhou et al. [15]CVPR'17	GT	0.183	1.595	6.709	0.270	0.734	0.902	0.959
Yang et al. [20]AAAI'18	GT	0.182	1.481	6.501	0.267	0.725	0.906	0.963
Mahjourian et al. [32]CVPR'18	GT	0.163	1.240	6.220	0.250	0.762	0.916	0.968
LEGO [13]CVPR'18	GT	0.162	1.352	6.276	0.252	-	-	-
DDVO [33]CVPR'18	GT	0.151	1.257	5.583	0.228	0.810	0.936	0.974
DF-Net [34]ECCV'18	GT	0.150	1.124	5.507	0.223	0.806	0.933	0.973
GeoNet [18]CVPR'18	GT	0.149	1.060	5.567	0.226	0.796	0.935	0.975
EPC++ [17]TPAMI'18	GT	0.141	1.029	5.350	0.216	0.816	0.941	0.976
Struct2Depth [12]AAAI'19	GT	0.141	1.026	5.291	0.215	0.816	0.945	<u>0.979</u>
CC [35]CVPR'19	GT	0.139	1.032	5.199	0.213	0.827	0.943	0.977
Bian et al. [36]NIPS'19	GT	0.128	1.047	5.234	0.208	<u>0.846</u>	0.947	0.976
Monodepth2 [14]ICCV'19	GT	<u>0.115</u>	<u>0.903</u>	<u>4.863</u>	<u>0.193</u>	**0.877**	<u>0.959</u>	**0.981**
DNet (Ours)	GT	**0.113**	**0.864**	**4.812**	**0.191**	**0.877**	**0.960**	**0.981**
Pinard et al. [21]ECCV'18	P	0.271	4.495	7.312	0.345	0.678	0.856	0.924
Roussel et al. [22]IROS'19	S	<u>0.175</u>	<u>1.585</u>	<u>6.901</u>	<u>0.281</u>	<u>0.751</u>	<u>0.905</u>	<u>0.959</u>
DNet (Ours)	DGC	**0.118**	**0.925**	**4.918**	**0.199**	**0.862**	**0.953**	**0.979**

where $y(\mathbf{P}_{i,j})$ denotes the y-axis value of $\mathbf{P}_{i,j}$. A ground mask is thereafter generated.

3) Camera height estimation: When all the ground points have been densely identified from the image, the geometrical relationship between ground points and camera itself is ready to be exploited. As can be seen from Fig. 5, camera height is the projection of vector $\overrightarrow{\mathbf{OP}_{i,j}}$ in the direction of surface normal of point $\mathbf{P}_{i,j}$, i.e., $\mathbf{N}(\mathbf{P}_{i,j})$. Therefore, camera height of $\mathbf{P}_{i,j}$ can be calculated as follows:

$$h(\mathbf{P}_{i,j}) = \mathbf{N}(\mathbf{P}_{i,j})^\top \cdot \overrightarrow{\mathbf{OP}_{i,j}} , \qquad (10)$$

where $\overrightarrow{\mathbf{OP}_{i,j}} = \mathbf{P}_{i,j} = [X, Y, Z]^\top$. This operation is done for all $\mathbf{P}_{i,j} \in \mathbf{GP}$.

Now a set of camera heights $\mathbf{H} = \{h(\mathbf{P}_{i,j})|\mathbf{P}_{i,j} \in \mathbf{GP}\}$ with element number equal to that of ground points is obtained. But for overall scale factor, one single camera height should be estimated for the relative depth map. After careful experiments, median of all estimated camera heights $h_M = Median(\mathbf{H})$ is selected as the final camera height.

4) Scale factor calculation: Given the camera height estimated for current relative depth map for $\mathbf{I}_t$, in order to calculate the scale factor, all that is still needed is the real height of the camera h_R. The scale factor for the current relative depth estimation is simply determined as follows:

$$f_t = \frac{h_R}{h_M} . \qquad (11)$$

C. Absolute depth estimation

After successfully estimated the scale factor for current relative depth map $\mathbf{D}_t^{rel}$, absolute depth can be thus pixel-wise calculated:

$$\mathbf{D}_t^{abs} = f_t \mathbf{D}_t^{rel}. \qquad (12)$$

where $\mathbf{D}_t^{abs}$ denotes the absolute depth estimated for current image $\mathbf{I}_t$.

IV. EXPERIMENT

Thorough experiments are presented here for evaluation of DNet pipeline. Quantitative results show our proposed DNet is able to achieve competitive performance on both relative depth estimation and scale recovery. Also, ablation study is performed to prove the effectiveness of our proposed DCP layer. And due to the dependency of enough visible ground, experiments under different ground point ratio show the robustness of DGC scale recovery module.

A. Implementation details

The same training parameters and method as Monodepth2 are used. Specifically, we set $\mu = 1, \lambda = 0.001$, and α for SSIM is equal to 0.85. Only monocular image sequence is used during training. For scale recovery, angle threshold $S_{max} = 5$. Low values are assigned to v_i and w_i for low-res predictions, i.e., $\mathbf{v} = \mathbf{w} = \{1/8, 1/4, 1/2, 1\}$.

The experiments are run on a computer with Intel Xeon 8163 CPU (2.5GHz) and NVIDIA RTX 2080 Ti.

B. Evaluation dataset

All experiments for evaluation of DNet are conducted on the Eigen split [2] of KITTI[16] 2015 containing 697 test images. For evaluation of depth estimation results, it contains ground truth projected from LiDAR 3D point clouds to 2D depth maps. However, there is no ground truth for scale factors to transfer relative depth maps to absolute depth maps. Usually used method is to use the ratio between medians of LiDAR detected depth values and estimated ones as ground truth of scale factor.

C. Quantitative evaluation

Thorough quantitative evaluation is presented to show the overall performance of DNet pipeline on both relative

and absolute depth estimation. Commonly used metrics are adopted for evaluation.

Table I demonstrates the overall depth estimation performance of DNet, both using ground-truth (GT) and DGC scale recovery, in comparison with 14 self-supervised monocular depth estimators. DNet with GT scale recovery is first evaluated to demonstrate its relative depth estimation performance. As can be seen from the table, DNet with GT scale recovery has achieved a satisfactory result. It has improved compared to Monodepth2 on former four metrics by respectively 1.74%, 4.32%, 1.05% and 1.04%.

In terms of absolute depth estimation, DGC performs almost as well as GT scale recovery. Compared to Roussel et al.[22], DNet achieves improvement on former four metrics by respectively 32.57%, 41.64%, 28.73% and 29.18%. The performance of DGC module can even outperform most early depth estimator using GT scale recovery. These indicate that DGC scale recovery method, in spite of its simplicity, can carry out a satisfactory scale recovery.

D. Ablation study

In order to better show the benefit of our proposed modules and the robustness against ground point ratio, comprehensive ablation study is conducted.

TABLE II

ABLATION STUDY. COMPARISON ON THE PERFORMANCE BETWEEN BASELINE AND OUR DNET WITH PROPOSED DCP LAYER. SCALE FACTOR IS DETERMINED USING LIDAR GROUND TRUTH.

Method	Lower is better				Higher is better		
	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Baseline	0.117	0.894	4.899	0.195	0.871	0.958	**0.981**
Ours	**0.113**	**0.864**	**4.812**	**0.191**	**0.877**	**0.960**	**0.981**

TABLE III

ABLATION STUDY. COMPARISON ON THE OBJECT LEVEL PREDICTION PERFORMANCE BETWEEN BASELINE AND OUR DNET WITH PROPOSED DCP LAYER. SCALE FACTOR IS DETERMINED USING LIDAR GROUND TRUTH.

Method	Lower is better				Higher is better		
	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Baseline	0.227	3.680	8.430	0.327	0.690	0.857	0.924
Ours	**0.202**	**2.817**	**7.941**	**0.310**	**0.725**	**0.875**	**0.932**

1) Benefit of DCP layer: In order to show the effectiveness of hierarchical feature generated by DCP layer, comparisons are made between baseline and DNet as can be seen in Table II. It can be seen that, our proposed DCP layer can boost the performance on the former four metrics by respectively 3.42%, 3.36%, 1.78%, 2.05%.

2) Benefit of DCP layer on object-level prediction: Depth estimation on objects can be challenging for the irrgular boundary and texture copy effects. To show the improvement of DCP layer on object-level prediction, Mask-RCNN[37] is used to generate object masks as shown in Fig.6 on test files and error metrics are calculated only within the masked

Fig. 6. **Object masks** extracted by Mask-RCNN[37], object level performance is calculated only on pixels in the mask.

areas. Table III compares performance between baseline and DNet on the object-level depth prediction. Our proposed DCP layer improves the object-level prediction performance on the former four metrics by respectively 11.01%, 23.45%, 5.80%, 2.14%.

E. Robustness of DGC scale recovery against visible ground:

Since DGC scale recovery largely depends on the ground points extraction, the relationship of its performance and the proportion of ground points in a single frame should be carefully evaluated. We evaluate 697 test images in Eigen split and plot ground points ratio and corresponding scale error of each frame. Result is shown in Fig. 7, where the x-axis is ground point ratio and y-axis is $\frac{\text{DGC}-\text{GT}}{\text{GT}}$. It can be seen that when the ground point ratio is larger than 1.03%, the proposed DGC module can perform uniformly and robustly comparable to GT scale recovery. But with extreme low ground points ratio, scale may be incorrectly estimated.
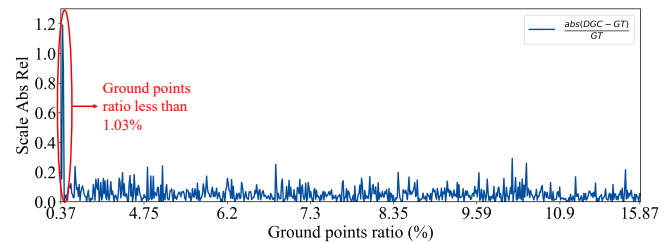

Fig. 7. **Robustness evaluation of DGC scale recovery module** under different ground point ratios. The result shows that when detected ground points take up more than 1.03% of all pixels, our proposed DGC module can perform comparable to GT scale recovery.

TABLE IV

SPEED PERFORMANCE OF DNET.

Stage	Time consumption
Inference	50.0ms
DGC scale recovery	4.1ms

F. Qualitative evaluation

Qualitative results are demonstrated in Fig. 8 and Fig. 9. Fig. 8 shows the overall absolute depth estimation results as well as intermediate results such as surface normal and ground point mask. Fig. 9 demonstrates intuitively the improvement brought by introducing DCP in comparison with baseline. It can be seen that object boundary is more precise and depth artifacts are to some extent eliminated.
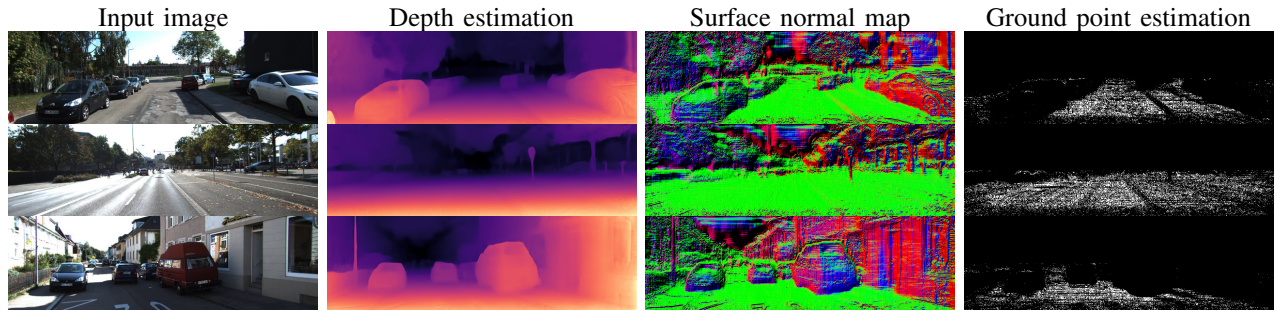
Fig. 8. **Qualitative results** of DNet absolute depth estimation result as well as components in DGC scale recovery module on KITTI 2015 Eigen Split.
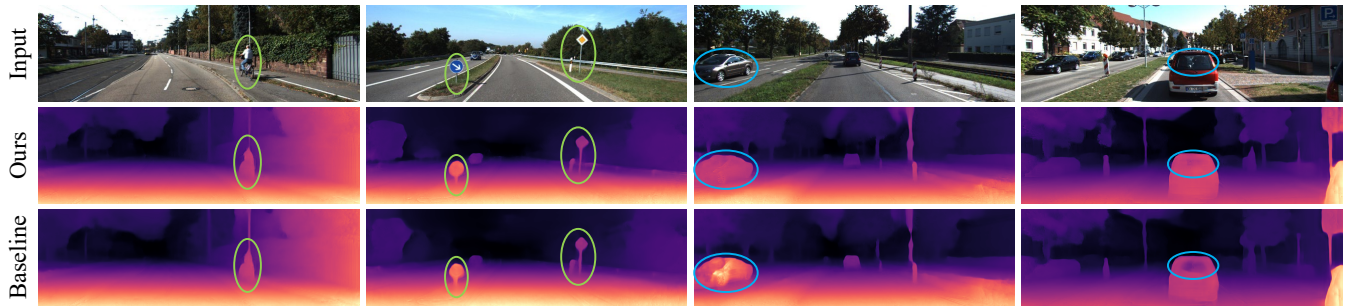

Fig. 9. **Qualitative results** of our proposed DCP layer on KITTI 2015 Eigen Split. Compared to baseline, DNet with DCP layer is able to present more precise object boundary (green) and significantly reduce depth artifacts (blue).

TABLE V

QUANTITATIVE RESULT IN SCENES LISTED IN FIG. 10. WHEN GROUND POINT RATIO IS RELATIVELY HIGH, DGC USUALLY PERFORMS BETTER THAN GT SCALE RECOVERY IN AT LEAST ON METRIC.

Frame	Scale Factor	Lower is better			
		Abs Rel	Sq Rel	RMSE	RMSE log
#106	GT	0.195	1.443	6.416	0.320
#106	DGC	0.110	1.105	5.799	0.319
#183	GT	0.194	1.175	5.888	0.231
#183	DGC	0.127	0.995	5.745	0.229
#330	GT	0.211	1.100	4.138	0.270
#330	DGC	0.144	0.844	4.174	0.271
#395	GT	0.353	2.181	5.837	0.418
#395	DGC	0.273	1.754	5.834	0.470

TABLE VI

RATIO OF THE FRAMES WHERE DGC SCALE RECOVERY PERFORMS BETTER THAN GT IN TERMS OF DIFFERENT METRICS. IT CAN BE SEEN THAT ESPECIALLY IN ABSOLUTE RELATIVE ERRORS, DGC PERFORMS BETTER IN MANY FRAMES.

Evaluation metrics			
Abs Rel	Sq Rel	RMSE	RMSE log
45.2%	38.5%	39.3%	31.7%

Fig. 10. **Scenes where DGC scale recovery performs better than GT scale recovery.** It can be intuitively seen that ground point ratio are all relatively high in those scenes.

G. Additional DGC and GT comprarisons

There are also results showing that in some cases, DGC scale recovery works even better than GT scale recovery, especially in those scenes, where ground point ratio is relatively large. Some example of those scenes can be seen in Fig. 10. The performance in those frames can be seen in Table V. Surprisingly, in at least 31.7% and at most 45.2% of the frames, DGC scale recovery module performs better in terms of four metrics. Detailed result of the ratio of frames where DGC performs favorably against GT scale recovery can be seen in Table VI.

V. CONCLUSIONS

In this work, a novel pipeline for self-supervised monocular absolute depth estimation is presented. DCP layer is proposed to generate hierarchical features for high resolution depth inferences, so that object boundary can be more accurate and depth artifacts can be better addressed. In order for the self-supervised monocular depth estimation to be more easily adapted to and used in autonomous driving applications, DGC module is introduced to perform absolute depth prediction without additional sensors and depth ground truth. Extensive experiments were conducted to demonstrate the effectiveness and robustness of the proposed DNet pipeline as well as DCP and DGC module. In future, this work provides intuition for better use of hierarchical features and can serve as the basis for further explorations of scale recovery methods.

REFERENCES

[1] K. Karsch, C. Liu, and S. Kang, "Depth extraction from video using non-parametric sampling-supplemental material," in *European conference on Computer Vision*. Citeseer, 2012.

[2] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Advances in neural information processing systems*, 2014, pp. 2366–2374.

[3] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2002–2011.

[4] F. Liu, C. Shen, G. Lin, and I. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 10, pp. 2024–2039, 2015.

[5] J. H. Lee, M.-K. Han, D. W. Ko, and I. H. Suh, "From big to small: Multi-scale local planar guidance for monocular depth estimation," *arXiv preprint arXiv:1907.10326*, 2019.

[6] W. Yin, Y. Liu, C. Shen, and Y. Yan, "Enforcing geometric constraints of virtual normal for depth prediction," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 5684–5693.

[7] I. Alhashim and P. Wonka, "High quality monocular depth estimation via transfer learning," *arXiv preprint arXiv:1812.11941*, 2018.

[8] M. Goldman, T. Hassner, and S. Avidan, "Learn stereo, infer mono: Siamese networks for self-supervised, monocular, depth estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[9] S. Pillai, R. Ambruș, and A. Gaidon, "Superdepth: Self-supervised, super-resolved monocular depth estimation," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 9250–9256.

[10] M. Poggi, F. Tosi, and S. Mattoccia, "Learning monocular depth estimation with unsupervised trinocular assumptions," in *2018 International Conference on 3D Vision (3DV)*. IEEE, 2018, pp. 324–333.

[11] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 270–279.

[12] V. Casser, S. Pirk, R. Mahjourian, and A. Angelova, "Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 8001–8008.

[13] Z. Yang, P. Wang, Y. Wang, W. Xu, and R. Nevatia, "Lego: Learning edge with geometry all at once by watching videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 225–234.

[14] C. Godard, O. M. Aodha, M. Firman, and G. J. Brostow, "Digging into self-supervised monocular depth estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 3828–3838.

[15] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1851–1858.

[16] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[17] C. Luo, Z. Yang, P. Wang, Y. Wang, W. Xu, R. Nevatia, and A. Yuille, "Every pixel counts++: Joint learning of geometry and motion with 3d holistic understanding," *arXiv preprint arXiv:1810.06125*, 2018.

[18] Z. Yin and J. Shi, "Geonet: Unsupervised learning of dense depth, optical flow and camera pose," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1983–1992.

[19] S. Vijayanarasimhan, S. Ricco, C. Schmid, R. Sukthankar, and K. Fragkiadaki, "Sfm-net: Learning of structure and motion from video," *arXiv preprint arXiv:1704.07804*, 2017.

[20] Z. Yang, P. Wang, W. Xu, L. Zhao, and R. Nevatia, "Unsupervised learning of geometry with edge-aware depth-normal consistency," *arXiv preprint arXiv:1711.03665*, 2017.

[21] C. Pinard, L. Chevalley, A. Manzanera, and D. Filliat, "Learning structure-from-motion from motion," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 0–0.

[22] T. Roussel, L. Van Eycken, and T. Tuytelaars, "Monocular depth estimation in new environments with absolute scale," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 1735–1741.

[23] D. P. Frost, O. Kähler, and D. W. Murray, "Object-aware bundle adjustment for correcting monocular scale drift," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 4770–4776.

[24] E. Sucar and J.-B. Hayet, "Probabilistic global scale estimation for monoslam based on generic object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 48–56.

[25] S. Choi, J. H. Joung, W. Yu, and J.-I. Cho, "What does ground tell us? monocular visual odometry under planar motion constraint," in *2011 11th International Conference on Control, Automation and Systems*. IEEE, 2011, pp. 1480–1485.

[26] S. Song, M. Chandraker, and C. C. Guest, "High accuracy monocular sfm and scale correction for autonomous driving," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 4, pp. 730–743, 2015.

[27] B. M. Kitt, J. Rehder, A. D. Chambers, M. Schonbein, H. Lategahn, and S. Singh, "Monocular visual odometry using a planar road model to solve scale ambiguity," 2011.

[28] D. Zhou, Y. Dai, and H. Li, "Ground-plane-based absolute scale estimation for monocular visual odometry," *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[29] X. Wang, H. Zhang, X. Yin, M. Du, and Q. Chen, "Monocular visual odometry scale recovery using geometrical constraint," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 988–995.

[30] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, *et al.*, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[31] M. Jaderberg, K. Simonyan, A. Zisserman, *et al.*, "Spatial transformer networks," in *Advances in neural information processing systems*, 2015, pp. 2017–2025.

[32] R. Mahjourian, M. Wicke, and A. Angelova, "Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5667–5675.

[33] C. Wang, J. Miguel Buenaposada, R. Zhu, and S. Lucey, "Learning depth from monocular videos using direct methods," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2022–2030.

[34] Y. Zou, Z. Luo, and J.-B. Huang, "Df-net: Unsupervised joint learning of depth and flow using cross-task consistency," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 36–53.

[35] A. Ranjan, V. Jampani, L. Balles, K. Kim, D. Sun, J. Wulff, and M. J. Black, "Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 240–12 249.

[36] J.-W. Bian, Z. Li, N. Wang, H. Zhan, C. Shen, M.-M. Cheng, and I. Reid, "Unsupervised scale-consistent depth and ego-motion learning from monocular video," *arXiv preprint arXiv:1908.10553*, 2019.

[37] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.