

# Enforcing geometric constraints of virtual normal for depth prediction

Wei Yin<sup>1</sup> Yifan Liu<sup>1</sup> Chunhua Shen<sup>1\*</sup> Youliang Yan<sup>2</sup>

<sup>1</sup>The University of Adelaide, Australia

<sup>2</sup>Noah’s Ark Lab, Huawei Technologies

## Abstract

Monocular depth prediction plays a crucial role in understanding 3D scene geometry. Although recent methods have achieved impressive progress in evaluation metrics such as the pixel-wise relative error, most methods neglect the geometric constraints in the 3D space. In this work, we show the importance of the high-order 3D geometric constraints for depth prediction. By designing a loss term that enforces one simple type of geometric constraints, namely, virtual normal directions determined by randomly sampled three points in the reconstructed 3D space, we can considerably improve the depth prediction accuracy. Significantly, the byproduct of this predicted depth being sufficiently accurate is that we are now able to recover good 3D structures of the scene such as the point cloud and surface normal directly from the depth, eliminating the necessity of training new sub-models as was previously done. Experiments on two benchmarks: NYU Depth-V2 and KITTI demonstrate the effectiveness of our method and state-of-the-art performance. Code is available at:

<https://tinyurl.com/virtualnormal>

## 1. Introduction

Monocular depth prediction aims to predict distances between scene objects and the camera from a single monocular image. It is a critical task for understanding the 3D scene, such as recognizing a 3D object and parsing a 3D scene.

Although the monocular depth prediction is an ill-posed problem because many 3D scenes can be projected to the same 2D image, many deep convolutional neural networks (DCNN) based methods [7, 8, 12, 14, 24, 27, 35] have achieved impressive results by using a large amount of labelled data, thus taking advantage of prior knowledge in labelled data to solve the ambiguity.

These methods typically formulate the optimization problem as either point-wise regression or classification. That is, with the i.i.d. assumption, the overall loss is summing over all pixels. To improve the performance, some

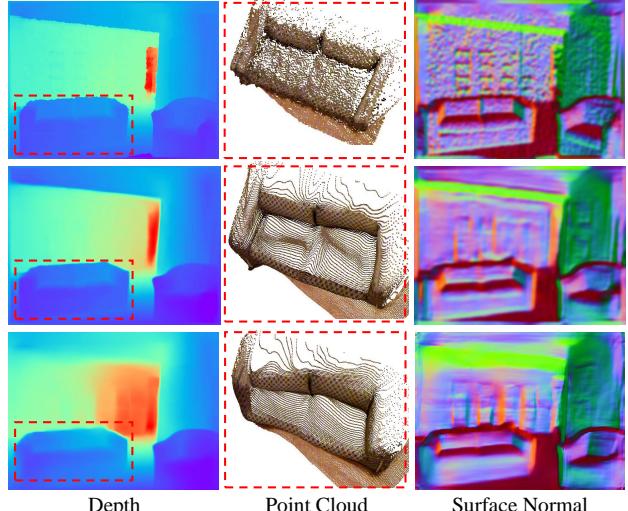


Figure 1. Example results of ground truth (the first row), our method (the second row) and Hu *et al.* [18] (the third row). By enforcing the geometric constraints of virtual normals, our reconstructed 3D point cloud can represent better shape of sofa (see the left part) and the recovered surface normal has much less errors (see green parts) even though the absolute relative error (rel) of our predicted depth is only slightly better than Hu *et al.* (0.108 vs. 0.115).

endeavours have been made to employ other constraints besides the pixel-wise term. For example, a continuous conditional random field (CRF) [28] is used for depth prediction, which takes pair-wise information into account. Other high-order geometric relations [9, 31] are also exploited, such as designing a gravity constraint for local regions [9] or incorporating the depth-to-surface-normal mutual transformation inside the optimization pipeline [31]. Note that, for the above methods, almost all the geometric constraints are ‘local’ in the sense that they are extracted from a small neighborhood in either 2D or 3D. Surface normal is ‘local’ by nature as it is defined by the local tangent plane. As the ground truth depth maps of most datasets are captured by consumer-level sensors, such as the Kinect, depth values can fluctuate considerably. Such noisy measurement would adversely affect the precision and subsequently the effectiveness of those local constraints inevitably. Moreover, local constraints calculated over a small neighborhood have

\*Corresponding author, email: chunhua.shen@adelaide.edu.au

not fully exploited the structure information of the scene geometry that may be possibly used to boost the performance.

To address these limitations, here we propose a more stable geometric constraint from a global perspective to take long-range relations into account for predicting depth, termed *virtual normal*. A few previous methods already made use of 3D geometric information in depth estimation, almost all of which focus on using surface normal. *We instead reconstruct the 3D point cloud from the estimated depth map explicitly.* In other words, we generate the 3D scene by lifting each RGB pixel in the 2D image to its corresponding 3D coordinate with the estimated depth map. This 3D point cloud serves as an intermediate representation. With the reconstructed point cloud, we can exploit many kinds of 3D geometry information, not limited to the surface normal. Here we consider the long-range dependency in the 3D space by randomly sampling three non-collinear points with the large distance to form a *virtual plane*, of which the normal vector is the proposed *virtual normal* (VN). The direction divergence between ground-truth and predicted VN can serve as a high-order 3D geometry loss. Owing to the long-range sampling of points, the adverse impact caused by noises in depth measurement is much alleviated compared to the computation of the surface normal, making VN significantly more accurate. Moreover, with randomly sampling we can obtain a large number of such constraints, encoding the global 3D geometric. Second, *by converting estimated depth maps from images to 3D point cloud representations it opens many possibilities of incorporating algorithms for 3D point cloud processing to 2D images and 2.5D depth processing.* Here we show one instance of such possibilities.

By combining the high-order geometric supervision and the pixel-wise depth supervision, our network can predict not only an accurate depth map but also the high-quality 3D point cloud, subsequently other geometry information such as the surface normal. It is worth noting that we do not use a new model or introduce network branches for estimating the surface normal. Instead it is computed directly from the reconstructed point cloud. The second row of Fig. 1 demonstrates an example of our results. By contrast, although the previously state-of-the-art method [18] predicts the depth with low errors, the reconstructed point cloud is far away from the original shape (see, e.g., left part of ‘sofa’). The surface normal also contains many errors. We are probably the first to achieve high-quality monocular depth and surface normal prediction with a single network.

Experimental results on NYUD-v2 [36] and KITTI [13] datasets demonstrate state-of-the-art performance of our method. Besides, when training with the lightweight backbone, MobileNetV2 [34], our framework provides a better trade-off between network parameters and accuracy. Our method outperforms other state-of-the-art real-time systems

by up to 29% with a comparable number of network parameters. Furthermore, from the reconstructed point cloud, we directly calculate the surface normal, with a precision being on par with that of specific DCNN based surface normal estimation methods.

In summary, our main contributions of this work are as follow.

- We demonstrate the effectiveness of enforcing a high-order geometric constraint in the 3D space for the depth prediction task. Such global geometry information is instantiated with a simple yet effective concept termed *virtual normal* (VN). By enforcing a loss defined on VNs, we demonstrate the importance of 3D geometry information in depth estimation, and design a simple loss to exploit it.
- Our method can reconstruct high-quality 3D scene point clouds, from which other 3D geometry features can be calculated, such as the surface normal. In essence, we show that for depth estimation, one should not consider the information represented by depth only. Instead, converting depth into 3D point clouds and exploiting 3D geometry is likely to improve many tasks including depth estimation.
- Experimental results on NYUD-V2 and KITTI illustrate that our method achieves state-of-the-art performance.

## 1.1. Related Work

**Monocular Depth Prediction.** Depth prediction from images is a long-standing problem. Previous work can be divided into active methods and passive methods. The former ones use the assistant optical information for prediction, such as coded patterns [41], while the latter ones completely focus on image contents. Monocular depth prediction [2, 7, 8, 28, 43] has been extensively studied recently. As limited geometric information can be directly extracted from the monocular image, it is essentially an ill-posed problem. Recently, owing to the structural features from very deep convolution neural network, such as ResNet [16], various DCNN-based methods learn to predict depth with deep CNN features. Fu *et al.* [12] proposed an encoder-decoder network, which extracts multi-scale features from the encoder and is trained in an end-to-end manner without iterative refinement. They achieved state-of-the-art performance on several datasets. Jiao *et al.* [19] proposed an attention-driven loss, which merges the semantic priors to improve the prediction precision on unbalanced distribution datasets.

Most previous methods only adopted the pixel-wise depth supervision to train a network. By contrast, Liu *et al.* [28] combined DCNN with the continuous conditional random field (CRF) to exploit consistency information of neighbouring pixels. CRF establishes a pair-wise constraint

for local regions. Furthermore, several high-order constraints are investigated. Chen *et al.* [5] applied the generative adversarial training to lead the network to learn a context-aware and patch-level loss automatically. Note that most of these methods directly work with the depth, instead of in the 3D space.

**Surface Normal.** Surface normal is an important geometry information for 3D scene understanding. Several data-driven methods [7, 8, 10, 11, 39, 42] have achieved promising results. Eigen *et al.* [7] proposed a CNN with different output channels to directly predict depth map, surface normal and semantic labels. Bansal *et al.* [1] proposed a two-stream network to predict the surface normal first, which is further joined with the input image to learn the pose. Note that most of these methods formulate surface normal prediction and depth prediction as multiple different tasks.

## 2. Our Method

Our approach resolves the monocular depth prediction and reconstructs the high-quality scene 3D point cloud from the predicted depth at the same time. The pipeline is illustrated in Fig. 2.

We take an RGB image  $I_{in}$  as the input of an encoder-decoder network and predict the depth map  $D_{pred}$ . From the  $D_{pred}$ , the 3D scene point cloud  $P_{pred}$  can be reconstructed. The ground truth point cloud  $P_{gt}$  is reconstructed from  $D_{gt}$ .

We enforce two types of supervision for training the network. We firstly follow standard monocular depth prediction methods to enforce pixel-wise depth supervision over  $D_{pred}$  with  $D_{gt}$ . With the reconstructed point clouds, we then align the spatial relationship between the  $P_{pred}$  and the  $P_{gt}$  using the proposed *virtual normal*.

When the network is well trained, we not only obtain accurate depth map but also high-quality point clouds. From the reconstructed point clouds, other 3D features can be directly calculated, such as the surface normal.

### 2.1. High-order Geometric Constraints

**Surface Normal.** The surface normal is an important ‘local’ feature for many point-cloud based applications such as registration [33] and object detection [17, 15]. It appears to be a promising 3D cue for improving depth prediction. One can apply the angular difference between ground-truth and calculated surface normal to be a geometric constraint. One major issue of this approach is, when computing surface normal from either a depth map or 3D point cloud, it is sensitive to noise. Moreover, surface normal only considers short-range local information.

We follow [21] to calculate the surface normal. It assumes that local 3D points locate in the same plane, of which the normal vector is the surface normal. In practice ground-truth depth maps are usually captured by

a consumer-level sensor with limited precision, so depth maps are contaminated by noise. The reconstructed point clouds in the local region can vary considerably due to noises as well as the size of local patch for sampling (Fig. 3(a)). We experiment on the NYUD-V2 dataset to test the robustness of the surface normal computation. Five different sampling sizes around the target pixel are employed to sample points, which are used to calculate its surface normal. The sample area is  $a = (2i+1) \cdot (2i+1)$ ,  $i = 1, \dots, 5$ . The Mean Difference Error (Mean) [7] between calculated surface normals is evaluated. From Fig. 3(b), we can learn that the surface normal varies significantly with different sampling sizes. For example, the Mean between  $3 \times 3$  and  $11 \times 11$  is  $22^\circ$ . Such unstable surface normal negatively affects its effectiveness for learning. Likewise, other 3D geometric constraints demonstrating the ‘local’ relative relations also encounter this problem.

**Virtual Normal.** In order to enforce robust high-order geometric supervision in the 3D space, we propose the virtual normal (VN) to establish 3D geometric connections between regions in a much larger range. The point cloud can be reconstructed from the depth based on the pinhole camera model. For each pixel  $p_i(u_i, v_i)$ , the 3D location  $P_i(x_i, y_i, z_i)$  in the world coordinate can be obtained by the prospective projection. We set the camera coordinate as the world coordinate. Then the 3D coordinate  $P_i$  is denoted as follows:

$$z_i = d_i, x_i = \frac{d_i \cdot (u_i - u_0)}{f_x}, y_i = \frac{d_i(v_i - v_0)}{f_y} \quad (1)$$

where  $d_i$  is the depth.  $f_x$  and  $f_y$  are the focal length along the  $x$  and  $y$  coordinate axis respectively.  $u_0$  and  $v_0$  are the 2D coordinate of the optical center.

We randomly sample  $N$  groups points from the depth map, with three points in each group. The corresponding 3D points are  $\mathcal{S} = \{(P_A, P_B, P_C)_i | i = 0 \dots N\}$ . Three points in a group are restricted to be non-collinear based on the restriction  $\mathcal{R}_1$ .  $\angle(\cdot)$  is the angle between two vectors.

$$\begin{aligned} \mathcal{R}_1 = & \{\alpha \geq \angle(\overrightarrow{P_A P_B}, \overrightarrow{P_A P_C}) \geq \beta, \\ & \alpha \geq \angle(\overrightarrow{P_B P_C}, \overrightarrow{P_B P_A}) \geq \beta | P \in \mathcal{S}\} \end{aligned} \quad (2)$$

where  $\alpha, \beta$  are hyper-parameters. In all experiments, we set  $\alpha = 120^\circ, \beta = 30^\circ$

In order to sample more long-range points, which have ambiguous relative locations in 3D space, we perform long-range restriction  $\mathcal{R}_2$  for each group in  $\mathcal{S}$ .

$$\mathcal{R}_2 = \{\|\overrightarrow{P_k P_m}\| > \theta | k, m \in [A, B, C], P \in \mathcal{S}\} \quad (3)$$

where  $\theta = 0.6m$  in our experiments.

Therefore, three 3D points in each group can establish a plane. We compute the normal vector of the plane to encode

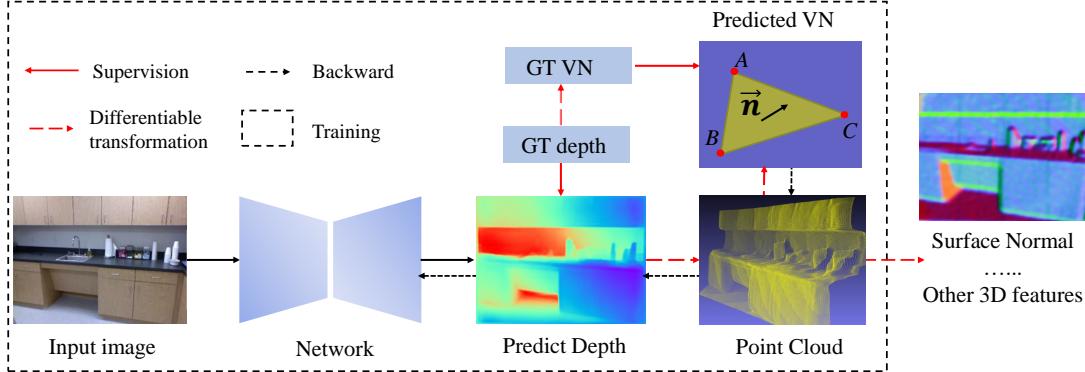


Figure 2. Illustration of the pipeline of our method. An encoder-decoder network is employed to predict the depth, from which the point cloud can be reconstructed. A pixel-wise depth supervision is firstly enforced on the predicted depth, while a geometric supervision, virtual normal constraint, is enforced in 3D space. With the well trained model, other 3D features, such as the surface normal, can be directly recovered from the reconstructed 3D point cloud in the inference.

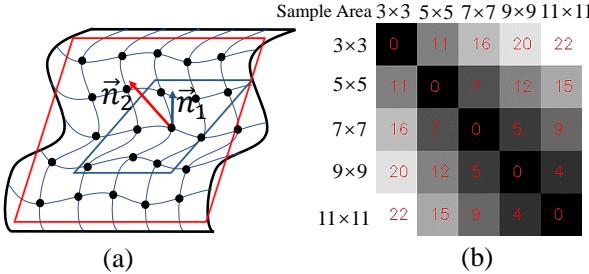


Figure 3. Illustration of fitting point clouds to obtain the local surface normal. The directions of the surface normals is fitted with different sampling sizes on a real point cloud (a). Because of noise, the surface normals vary significantly. (b) compares the angular difference between surface normals computed with different sample sizes in Mean Difference Error. The error can vary significantly.

geometric relations, which can be written as

$$\mathcal{N} = \{\mathbf{n}_i = \frac{\overrightarrow{P_A P_B} \times \overrightarrow{P_A P_C}}{\|\overrightarrow{P_A P_B} \times \overrightarrow{P_A P_C}\|} \mid (P_A, P_B, P_C)_i \in \mathcal{S}, i = 0 \dots N\} \quad (4)$$

where  $\mathbf{n}_i$  is the normal vector of the virtual plane  $i$ .

**Robustness to Depth Noise.** Compared with local surface normal, our virtual normal is more robust to noise. In Fig. 4, we sample three 3D points with large distance.  $P_A$  and  $P_B$  are assumed to locate on the  $XY$  plane,  $P_C$  is on the  $Z$  axis. When  $P_C$  varies to  $P_C'$ , the direction of the virtual normal changes from  $\mathbf{n}$  to  $\mathbf{n}'$ .  $P_C''$  is the intersection point between plane  $P_A P_B P_C'$  and  $Z$  axis. Because of restrictions  $\mathcal{R}_1$  and  $\mathcal{R}_2$ , the difference between  $\mathbf{n}$  and  $\mathbf{n}'$  is usually very small, which is simple to show:

$$\angle(\mathbf{n}, \mathbf{n}') = \angle(\overrightarrow{O P_C}, \overrightarrow{O P_C'}) = \arctan \frac{\|\overrightarrow{P_C P_C'}\|}{\|\overrightarrow{O P_C}\|} \approx 0, \\ \|\overrightarrow{P_C P_C'}\| \ll \|\overrightarrow{O P_C}\| \quad (5)$$

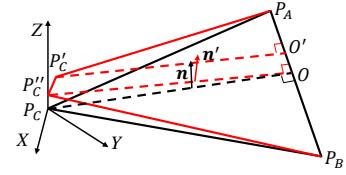


Figure 4. Robustness of VN to depth noise.

Furthermore, we conduct a simple experiment to verify the robustness of our proposed virtual normal against data noise. We create an unit sphere and then add gaussian noise to simulate the ideal noise-free data and the real noisy data (see Fig. 5a). We then sample 100K groups points from the noisy surface and the ideal one to compute the virtual normal respectively, while 100K points are sampled to compute the surface normal as well. For the gaussian noise, we use different deviations to simulate different noise levels by varying deviation  $\sigma = [0.0002, \dots, 0.01]$ , and the mean being  $\mu = 0$ . The experimental results are illustrated in Fig. 15c. We can learn that our proposed virtual normal is much more robust to the data noise than the surface normal. Other local constraints are also sensitive to data noise.

Most ‘local’ geometric constraints, such as the surface normal, actually enforcing the first-order smoothness of the surface but are less useful for helping the depth map prediction. In contrast, the proposed VN establishes long-range relations in the 3D space. Compared with pairwise CRFs, VN encodes triplet based relations, thus being of high order.

**Virtual Normal Loss.** We can sample a large number of triplets and compute corresponding VNs. With the sampled VNs, we compute the divergence as the Virtual Normal Loss (VNL):

$$\ell_{VN} = \frac{1}{N} \left( \sum_{i=0}^N \|\mathbf{n}_i^{pred} - \mathbf{n}_i^{gt}\|_1 \right) \quad (6)$$

where the  $N$  is the number of valid sampling groups satisfy-

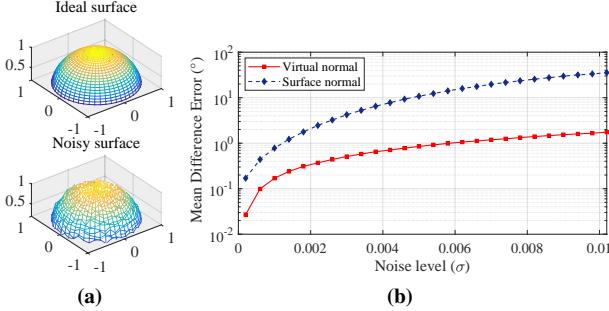


Figure 5. Robustness of virtual normal and surface normal against data noise. (a) The ideal surface and noisy surface. (b) The Mean Difference Error (Mean) is applied to evaluate the robustness of virtual normal and surface normal against different noise level. Our proposed virtual normal is more robust.

ing  $\mathcal{R}_1, \mathcal{R}_2$ . In experiments we have employed online hard example mining.

**Pixel-wise Depth Supervision.** We also use a standard pixel-wise depth map loss. We quantize the real-valued depth and formulate the depth prediction as a classification problem instead of regression, and employ the cross-entropy loss. In particular we follow [2] to use the weighted cross-entropy loss (WCEL), with the weight being the information gain. See [2] for details.

To obtain the accurate depth map and recover high-quality 3D information, we combine WCEL and VNL together to supervise the network output. The overall loss is:

$$\ell = \ell_{WCE} + \lambda \ell_{VN}, \quad (7)$$

where  $\lambda$  is a trade-off parameter, which is set to 5 in all experiments to make the two terms roughly of the same scale.

Note that the above overall loss function is differentiable. The gradient of the  $\ell_{VN}$  loss can be easily computed as Eq. (4) and Eq. (6) are both differentiable.

### 3. Experiments

In this section, we conduct several experiments to compare ours against state-of-the-art methods. We evaluate our methods on two datasets, NYUD-V2 and KITTI.

#### 3.1. Datasets

**NYUD-V2.** The NYUD-V2 dataset consists of 464 different indoor scenes, which are further divided into 249 scenes for training and 215 for testing. We randomly sample 29K images from the training set to form NYUD-Large. Note that DORN uses the whole training set, which is significantly larger than that what we use. Apart from the whole dataset, there are officially annotated 1449 images (NYUD-Small), in which 795 images are split for training and others are for testing. In the ablation study, we use the NYUD-Small data.

Table 1. Results on NYUD-V2. Our method outperforms other state-of-the-art methods over all evaluation metrics.

Method	rel	log10	rms	$\delta_1$	$\delta_2$	$\delta_3$
	Lower is better		Higher is better			
Saxena <i>et al.</i> [35]	0.349	-	1.214	0.447	0.745	0.897
Karsch <i>et al.</i> [20]	0.349	0.131	1.21	-	-	-
Liu <i>et al.</i> [29]	0.335	0.127	1.06	-	-	-
Ladicky <i>et al.</i> [23]	-	-	-	0.542	0.829	0.941
Li <i>et al.</i> [25]	0.232	0.094	0.821	0.621	0.886	0.968
Roy <i>et al.</i> [32]	0.187	0.078	0.744	-	-	-
Liu <i>et al.</i> [28]	0.213	0.087	0.759	0.650	0.906	0.974
Wang <i>et al.</i> [38]	0.220	0.094	0.745	0.605	0.890	0.970
Eigen <i>et al.</i> [7]	0.158	-	0.641	0.769	0.950	0.988
Chakrabarti [3]	0.149	-	0.620	0.806	0.958	0.987
Li <i>et al.</i> [26]	0.143	0.063	0.635	0.788	0.958	0.991
Laina <i>et al.</i> [24]	0.127	0.055	0.573	0.811	0.953	0.988
DORN [12]	0.115	0.051	0.509	0.828	0.965	0.992
Ours	<b>0.108</b>	<b>0.048</b>	<b>0.416</b>	<b>0.875</b>	<b>0.976</b>	<b>0.994</b>

**KITTI.** The KITTI dataset contains over 93K outdoor images and depth maps with the resolution around  $1240 \times 374$ . All images are captured on driving cars by stereo cameras and a Lidar. We test on 697 images from 29 scenes split by Eigen *et al.* [8], validate on 888 images, and train on about 23488 images from the remaining 32 scenes.

#### 3.2. Implementation Details

The pre-trained ResNeXt-101 [40] ( $32 \times 4d$ ) model on ImageNet [6] is used as our backbone model. A polynomial decaying method with the base learning rate 0.0001 and the power of 0.9 is applied for SGD. The weight decay and the momentum are set to 0.00005 and 0.9 respectively. Batch size is 8 in our experiments. The model is trained for 10 epochs on NYUD-Large and KITTI, and is trained for 40 epochs on NYUD-Small in the ablation study. We perform the data augmentation on the training samples by the following methods. For NYUD-V2, the RGB image and the depth map are randomly resized with ratio  $[1, 0.92, 0.86, 0.8, 0.75, 0.7, 0.67]$ , randomly flipped in the horizon, and finally randomly cropped with the size  $384 \times 384$  for NYUD-V2. The similar process is applied for KITTI but resizing with the ratio  $[1, 1.1, 1.2, 1.3, 1.4, 1.5]$  and cropping with  $384 \times 512$ . Note that the depth map should be scaled with the corresponding resizing ratio.

#### 3.3. Evaluation Metrics

We follow previous methods [24] to evaluate the performance of monocular depth prediction quantitatively based on following metrics: mean absolute relative error (rel), mean  $\log_{10}$  error (log<sub>10</sub>), root mean squared error (rms), root mean squared log error (rms(log)) and the accuracy under threshold ( $\delta_i < 1.25^i, i = 1, 2, 3$ ).

#### 3.4. Comparison with State-of-the-art

In this section, we detail the comparison of our methods with state-of-the-art methods.

**NYUD-V2.** In this experiment, we compare with other state-of-the-art methods on the NYUD-V2 dataset. Table 1 demonstrates that our proposed method outperforms other state-of-the-art methods across all evaluation metrics significantly. Compare to DORN, we have improved the accuracy from 0.2% to 18% over all evaluation metrics that they report.

In addition to the quantitative comparison, we demonstrate some visual results between our method and the state-of-the-art DORN in Fig. 6. Clearly, the predicted depth by the proposed method is much more accurate. The plane of ours is much smoother and has fewer errors (see the wall regions colored with red in the 1st, 2nd, and 3rd row). Furthermore, the last row in Fig. 6 manifests that our predicted depth is more accurate in the complicated scene. We have fewer errors in shelf and desk regions.

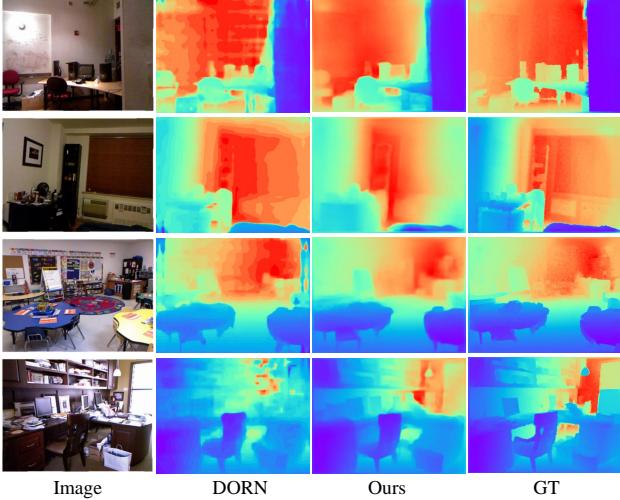


Figure 6. Examples of predicted depth maps by our method and the state-of-the-art DORN on NYUD-V2. Color indicates the depth (red is far, purple is close). Our predicted depth maps have fewer errors in planes (see walls) and have high-quality details in complicated scenes (see the desk and shelf in the last row)

**KITTI.** In order to demonstrate that our proposed method can still reach the state-of-the-art performance on outdoor scenes, we test our method on the KITTI dataset. Results in Table 2 show that our method has outperformed all other methods on all evaluation metrics except root mean square (rms) error. The rms error is only slightly behind that of DORN. Note that for outdoor scenes, the rms (log) error, instead of rms, is usually the metric of interest, in which ours is better.

### 3.5. Ablation Studies

In this section, we conduct several ablation studies to analyze the details of our approach.

**Effectiveness of VNL.** In this study, in order to prove the

Table 2. Results on KITTI. Our method outperforms other methods over all evaluation metrics except rms.

Method	$\delta_1$	$\delta_2$	$\delta_3$	rel	rms	rms (log)
	Higher is better			Lower is better		
Make3D [35]	0.601	0.820	0.926	0.280	8.734	0.361
Eigen <i>et al.</i> [8]	0.692	0.899	0.967	0.190	7.156	0.270
Liu <i>et al.</i> [28]	0.647	0.882	0.961	0.114	4.935	0.206
Semi. [22]	0.862	0.960	0.986	0.113	4.621	0.189
Guo <i>et al.</i> [14]	0.902	0.969	0.986	0.090	3.258	0.168
DORN [12]	0.932	0.984	0.994	0.072	<b>2.727</b>	0.120
Ours	<b>0.938</b>	<b>0.990</b>	<b>0.998</b>	<b>0.072</b>	3.258	<b>0.117</b>

Table 3. Illustration of the effectiveness of VNL.

Metrics	rel	log10	rms	$\delta_1$	$\delta_2$	$\delta_3$
<hr/> Pixel-wise Depth Supervision <hr/>						
CEL	0.1456	0.061	0.617	0.8087	0.9559	0.9862
WCEL	0.1427	0.060	0.511	0.8117	0.9611	0.9895
WCEL+L1	0.1429	0.061	0.626	0.8098	0.9539	0.9858
<hr/> Pixel-wise Depth Supervision + Geometric Supervision <hr/>						
WCEL+PL <sup>‡</sup>	0.1380	0.059	0.504	0.8212	0.9643	0.9913
WCEL+PL+VNL	0.1341	0.056	0.485	<b>0.8336</b>	<b>0.9671</b>	0.9913
WCEL+SNL <sup>†</sup>	0.1406	0.059	0.599	0.8209	0.9602	0.9886
WCEL+VNL <sup>‡</sup> (Ours)	<b>0.1337</b>	<b>0.056</b>	<b>0.480</b>	0.8323	0.9669	<b>0.9920</b>

<sup>†</sup> ‘Local’ geometric supervision in 3D.

<sup>‡</sup> ‘Global’ geometric supervision in 3D.

effectiveness of the proposed VNL we compare it with two types of pixel-wise depth map supervision, a pair-wise geometric supervision, and a high-order geometric supervision: 1) the ordinary cross-entropy loss (CEL); 2) the  $L_1$  loss ( $L_1$ ); 3) the surface normal loss (SNL); 4) the pair-wise geometric loss (PL). We reconstruct the point cloud from the depth map and further recover the surface normal from the point cloud. The angular discrepancy between the ground truth and recovered surface normal is defined as the surface normal loss, which is a high-order geometric supervision in 3D space. The pair-wise loss is the direction difference of two vectors in 3D, which are established by randomly sampling paired points in ground-truth and predicted point cloud. The loss function of PL is as follow,

$$\ell_{PL} = \frac{1}{N} \sum_{i=0}^N \left( 1 - \frac{\overrightarrow{P_{Ai}^* P_{Bi}^*} \cdot \overrightarrow{P_{Ai} P_{Bi}}}{\|\overrightarrow{P_{Ai}^* P_{Bi}^*}\| \cdot \|\overrightarrow{P_{Ai} P_{Bi}}\|} \right) \quad (8)$$

where  $(P_A^*, P_B^*)_i$  and  $(P_A, P_B)_i$  are paired points sampled from the ground truth and the predicted point cloud respectively.  $N$  is the total number of pairs.

We also employ the long-range restriction  $\mathcal{R}_2$  for the paired points. Therefore, similar to VNL, PL can also be seen as a global geometric supervision in 3D space. The experimental results are reported in Table. 3. WCEL is the baseline for all following experiments.

Firstly, we analyze the effect of pixel-wise depth supervision for prediction performance. As WCE employs an weight in the CE loss, its performance is slightly better than that of CEL. However, when we enforce two pixel-wise supervision (WCEL+L1) on the depth map, the performance

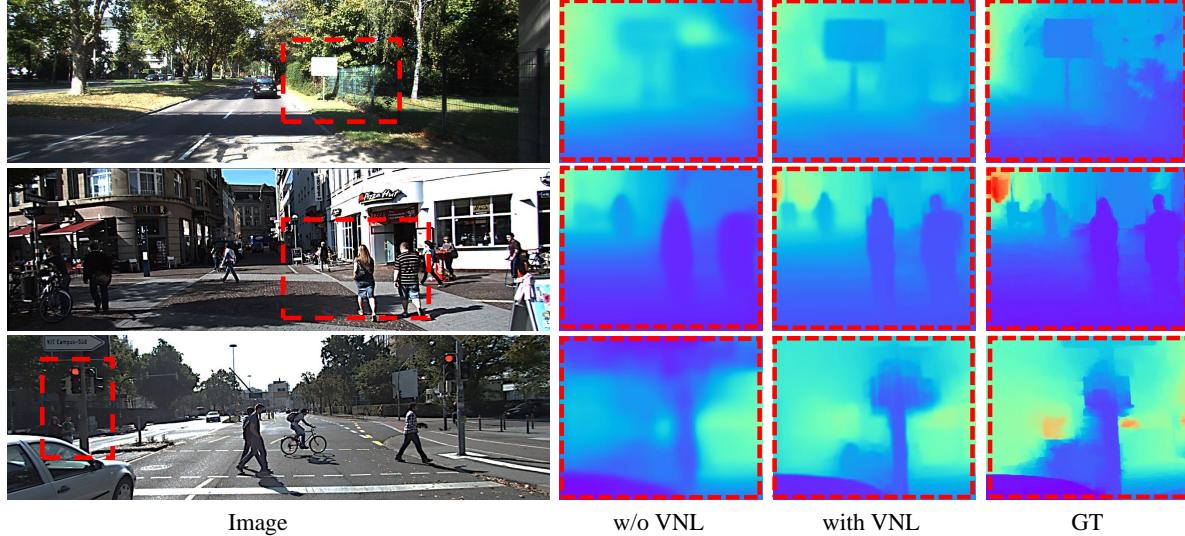


Figure 7. Depth maps in the red dashed boxes with sign, pedestrian and traffic lights are zoomed in. One can see that with the help of virtual normal, predicted depth maps in these ambiguous regions are considerably more accurate.

cannot improve any more. Thus using two pixel-wise loss terms does not help.

Secondly, we analyze the effectiveness of the supplementary 3D geometric constraint (PL, SNL, VNL). Compared with the baseline (WCEL), three supplementary 3D geometric constraints can promote the network performance with varying degrees. Our proposed VNL combining with WCEL has the best performance, which has improved the baseline performance by up to 8%.

Thirdly, we analyze the difference of three geometric constraints. As SNL can only exploit geometric relations of homogeneous local regions, its performance is the lowest among the three constraints over all evaluation metrics. Compared with SNL, since PL constrains the global geometric relations, its performance is clearly better. However, the performance of WCEL+PL is not as good as our proposed WCEL+VNL. When we further add our VNL on top of WCEL+PL, the precision can further be slightly improved and is comparable to WCEL+VNL. Therefore, although PL is a global geometric constraint in 3D, the pairwise constraint cannot encode as strong geometry information as our proposed VNL.

At last, in order to further demonstrate the effectiveness of VNL, we analyze the results of network trained with and without VNL supervision on the KITTI dataset. The visual comparison is shown in Fig. 7. One can see that VNL can improve the performance of the network in ambiguous regions. For example, the sign (1st row), the distant pedestrian (2nd row), and traffic light in the last row of the figure can demonstrate the effectiveness of the proposed VNL.

In conclusion, the geometric constraints in the 3D space can significantly boost the network performance. Moreover, the global and high-order constraints can enforce stronger

supervision than the ‘local’ and pair-wise ones in 3D space.

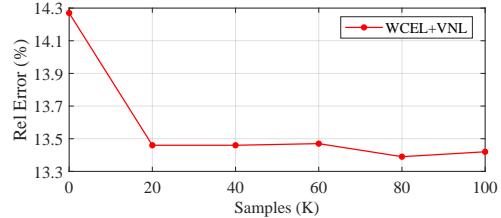


Figure 8. Illustration of the impact of the samples size. The more samples will promote the performance.

**Impact of the Amount of Samples.** Previously, we have proved the effectiveness of VNL. Here the impact of the size of samples for VNL is discussed. We sample six different sizes of point groups, 0K, 20K, 40K, 60K, and 80K and 100K, to establish VNL. ‘0K’ means that the model is trained without VNL supervision. The rel error is reported for evaluation. Fig. 8 demonstrates that ‘rel’ slumps by 5.6% with 20K point groups to establish VNL. However, it only drops slightly when the samples for VNL increase from 20K to 100K. Therefore, the performance saturates with more samples, when samples reach a certain number in that the diversity of samples is enough to construct the global geometric constraint.

**Lightweight Backbone Network.** We train the network with the MobileNetV2 backbone to evaluate the effectiveness of the proposed geometric constraint on the light network. We train it on the NYUD-Large for 10 epochs. Results in Table 4 show that the proposed VNL can improve the performance by 1% - 8%. Comparing with previous state-of-the-art methods, we have improved the accuracy by around 29% over all evaluation metrics and achieved a bet-

ter trade-off between parameters and the accuracy.

Table 4. Performance on NYUD-V2 with MobileNetV2 backbone.  
<sup>†</sup>Trained without VN. <sup>‡</sup>Trained with VN.

Metrics	CReAM [37]	RF-LW[30]	Ours-B <sup>†</sup>	Ours-VN <sup>‡</sup>
$\delta_1$	0.704	0.790	0.814	<b>0.829</b>
$\delta_2$	0.917	0.955	0.947	<b>0.956</b>
$\delta_3$	0.977	<b>0.990</b>	0.972	0.980
rel	0.190	0.149	0.144	<b>0.134</b>
rms	0.687	0.565	0.502	<b>0.485</b>
rms (log)	0.251	0.205	0.201	<b>0.185</b>
params	<b>1.5M</b>	3.0M	2.7M	2.7M

### 3.6. Recovering 3D Features from Estimated Depth

We have argued that, with geometric constraints in the 3D space, the network can achieve more accurate depth and also obtain higher-quality 3D information. Here we show the recovered 3D point cloud and the surface normal to support this.

**3D Point Cloud.** Firstly, we compare the reconstructed 3D point cloud from our predicted depth and that of DORN. Fig. 9 demonstrate that the overall quality of ours outperforms theirs significantly. Although our predicted depth is only slightly better than theirs on evaluation metrics, the reconstructed wall (see the 2nd row in 9) of ours is much flatter and has fewer errors. The shape of the bed is more similar to the ground truth. From the bird view, it is hard to recognize the bed shape of their results. The point cloud in Fig. 1 also leads to a similar conclusion.

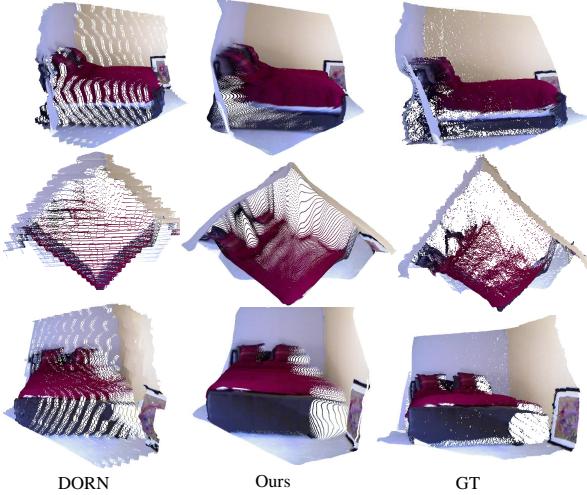


Figure 9. Comparison of reconstructed point clouds from estimated depth maps between DORN [12] and ours. We can see that our point cloud results contain less noise and are closer to ground-truth than that of DORN.

**Surface Normal.** Lastly, we compare the calculated surface normal with previous state-of-the-art methods and demonstrate the quantitative results in Table 5. The ground truth

is obtained as described in [7]. We first compare our geometrically calculated results with DCNN-based optimization methods. Although we do not optimize a sub-model to achieve the surface normal, our results can outperform most of the previous methods and even are the best on  $30^\circ$  metric.

Table 5. Evaluation of the surface normal on NYUD-V2.

Method	Mean	Median	$11.2^\circ$	$22.5^\circ$	$30^\circ$
	Lower is better	Higher is better			
Predicted Surface Normal from the Network					
3DP [10]	33.0	28.3	18.8	40.7	52.4
Ladicky <i>et al.</i> [42]	35.5	25.5	24.0	45.6	55.9
Fouhey <i>et al.</i> [11]	35.2	17.9	40.5	54.1	58.9
Wang <i>et al.</i> [39]	28.8	17.9	35.2	57.1	65.5
Eigen <i>et al.</i> [7]	<b>23.7</b>	<b>15.5</b>	<b>39.2</b>	<b>62.0</b>	71.1
Calculated Surface Normal from the Point cloud					
GT-GeoNet <sup>†</sup> [31]	36.8	32.1	15.0	34.5	46.7
DORN <sup>‡</sup> [12]	36.6	31.1	15.7	36.5	49.4
Ours	24.6	17.9	34.1	60.7	<b>71.7</b>

<sup>†</sup> Cited from the original paper.

<sup>‡</sup> Using authors' released models.

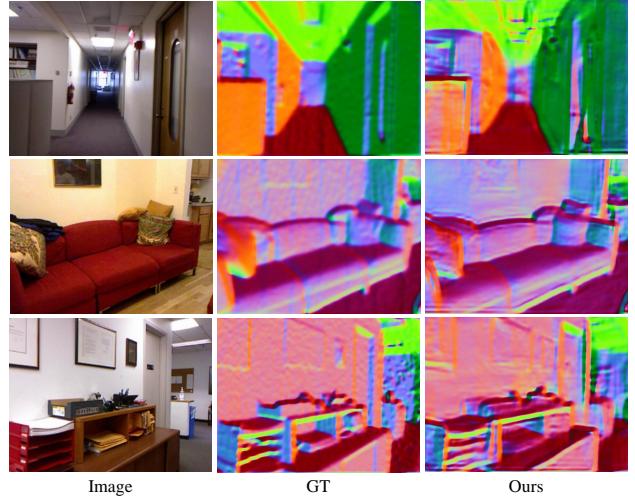


Figure 10. Recovered surface normal from 3D point cloud. According to the visual effect, the surface normal is in high-quality in planes (1st row) and the complicated curved surface (2nd and last row).

Furthermore, we compare the surface normals directly computed from the reconstructed point cloud with that of DORN [12] and GeoNet [31]. Note that we run the released code and model of DORN to obtain depth maps and then calculate surface normals from the depth, while the evaluation of GeoNet is cited from the original paper. In Table 5, we can see that, with high-order geometric supervision, our method outperforms DORN and GeoNet by a large margin, and even is close to Eigen method which trains to output normals. It suggests that our method can lead the model to learn the shape from images.

Apart from the quantitative comparison, the visual effect is shown in Fig. 10, demonstrating that our directly calculated surface normals are not only accurate in planes (the 1st row), but also are of higher quality in regions with sophisticated curved surface (the 2nd and last row).

## 4. Conclusion

In this paper, we have proposed to construct a *long-range* geometric constraint (VNL) in the 3D space for monocular depth prediction. In contrast to previous methods with only pixel-wise depth supervision in 2D space, our method can not only obtain the accurate depth maps but also recover high-quality 3D features, such as the point cloud and the surface normal, eliminating necessities to optimize a new sub-model. Compared with other 3D constraints, our proposed VNL is more robust to noise and can encode strong global constraints. Experimental results on NYUD-V2 and KITTI have proved the effectiveness of our method and the state-of-the-art performance.

In particular, to demonstrate that our method is able to produce sensible local shapes, the normals directly derived from the estimated depth of our method outperform many other recent depth estimation methods and are close to that of those trained to output normals. We hope that our method provides a useful tool and stimulates insight into predicting not only depth but also shape from monocular images.

## 5. Appendix

### 5.1. Model

An overview architecture of our model is illustrated in Fig.11. The network is mainly composed of two parts, an encoder to establish features in different levels from  $I_{in}$ , and a decoder to reconstruct the depth map. Inspired by [27], the decoder is composed of several adaptive merging blocks (AMB) to fuse features from different levels and dilated residual blocks (DRB) to transform features. In order to improve the receptive field of the decoder, we set the dilation rates of all  $3 \times 3$  convolutions in DRB to 2 and insert an Astrous Spatial Pyramid Pooling (ASPP) module (dilation rate: 2, 4, 8) [4] between the encoder and the decoder. Furthermore, we establish 4 flip connections from different levels of encoder blocks to the decoder to merge more low-level features. The AMB will learn a merging parameter for adaptive merging. Apart from features from the highest level with 512 channels, other flips' features dimension are 256. At last, a prediction module, a  $3 \times 3$  convolution and a softmax, is applied to transfer the features dimensions from 256 channels to 150 depth bins.

In the lightweight backbone network experiment, the backbone is replaced with MobileNetV2. In order to further reduce parameters, the dimensions of four flip connections are reduced to (128, 64, 64, 64). In the prediction module,

the features are transferred from 64 channels to 60 depth bins.

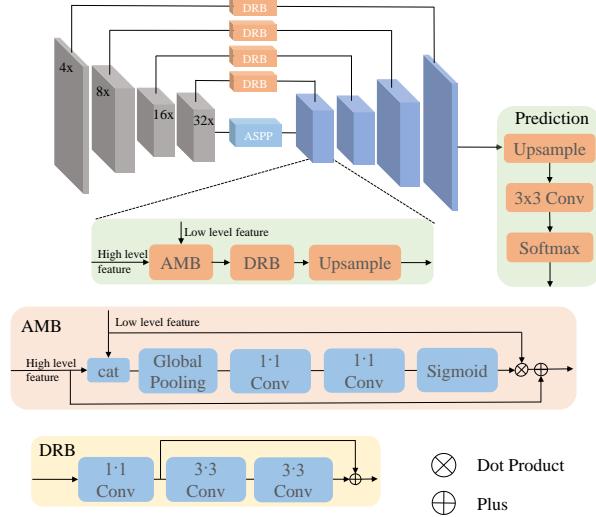


Figure 11. Model architecture. The encoder-decoder network has four flip connections to merge low-level features.

### 5.2. Predicted Depth and Surface Normal

We provide more predicted depth maps and recovered surface normals on KITTI and NYUD-V2 dataset. Depth maps are illustrated in Fig. 12, and Fig. 14, the recovered surface normals are demonstrated in Fig. 13.

### 5.3. 3D point cloud

In order to further show the quality of reconstructed point cloud from the predicted depth, we randomly select 3 scenes from the testing part of NYUD-V2 and KITTI. 3 views are randomly selected to display the reconstructed point cloud. The results are shown in Fig. 15 and Fig. 16.

## Acknowledgments

We would like to thank Huawei Technologies for the donation of GPU cloud computing resources. We are particularly grateful to one of the reviewers who sees the value of our work and has provided constructive comments.

## References

- [1] A. Bansal, B. Russell, and A. Gupta. Marr revisited: 2d-3d alignment via surface normal prediction. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 5965–5974, 2016.
- [2] 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.*, 2017.
- [3] A. Chakrabarti, J. Shao, and G. Shakhnarovich. Depth from a single image by harmonizing overcomplete local network

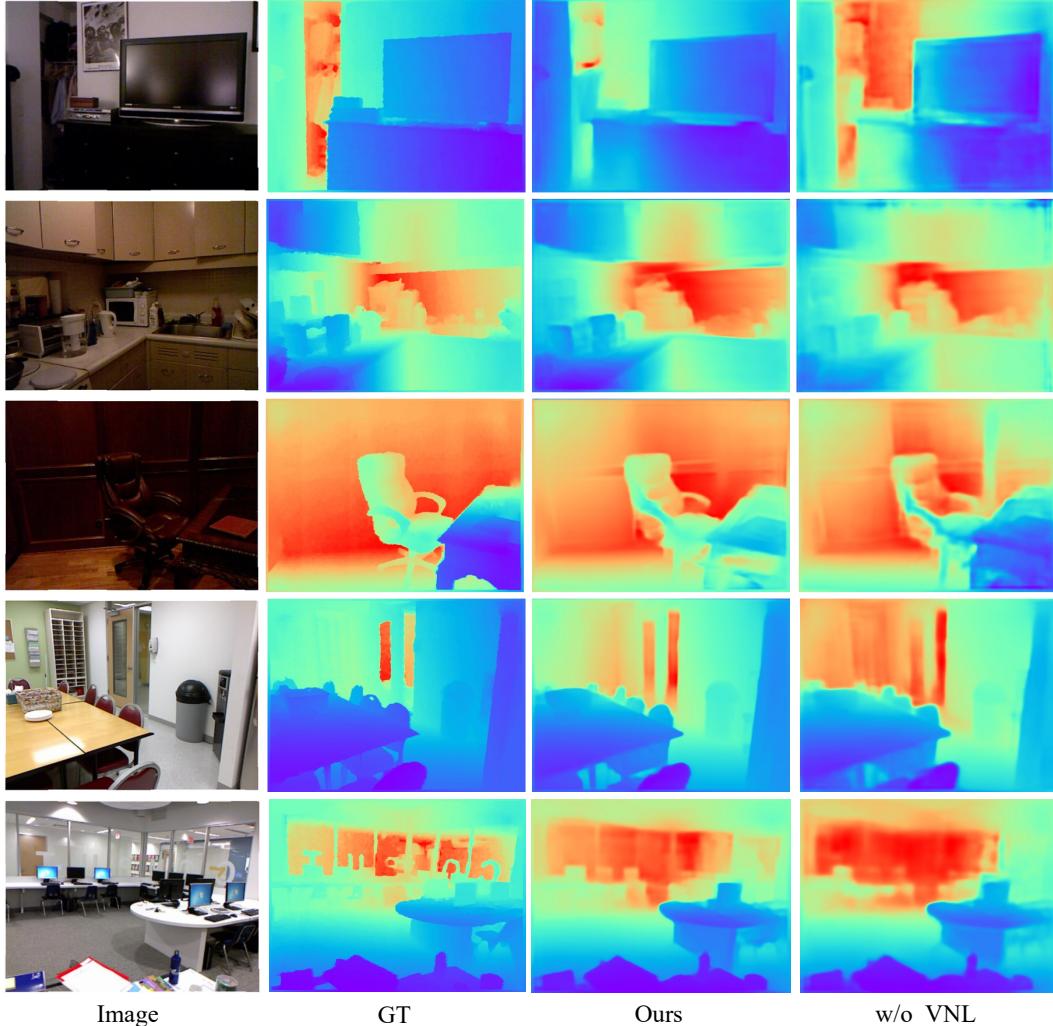


Figure 12. Samples of the predicted depth on NYUD-V2. By adding the virtual normal constraints, our proposed model can produce more accurate and smooth depth map.

- predictions. In *Proc. Advances in Neural Inf. Process. Syst.*, pages 2658–2666, 2016.
- [4] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *arXiv: Comp. Res. Repository*, volume abs/1802.02611, 2018.
  - [5] R. Chen, F. Mahmood, A. Yuille, and N. J. Durr. Rethinking monocular depth estimation with adversarial training. In *arXiv: Comp. Res. Repository*, volume abs/1808.07528, 2018.
  - [6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 248–255. Ieee, 2009.
  - [7] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 2650–2658, 2015.

- [8] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Proc. Advances in Neural Inf. Process. Syst.*, pages 2366–2374, 2014.
- [9] X. Fei, A. Wang, and S. Soatto. Geo-supervised visual depth prediction. In *arXiv: Comp. Res. Repository*, volume abs/1807.11130, 2018.
- [10] D. F. Fouhey, A. Gupta, and M. Hebert. Data-driven 3d primitives for single image understanding. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 3392–3399, 2013.
- [11] D. F. Fouhey, A. Gupta, and M. Hebert. Unfolding an indoor origami world. In *Proc. Eur. Conf. Comp. Vis.*, pages 687–702. Springer, 2014.
- [12] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao. Deep ordinal regression network for monocular depth estimation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 2002–2011, 2018.
- [13] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *SAGE Int. J. Robotics Research*,

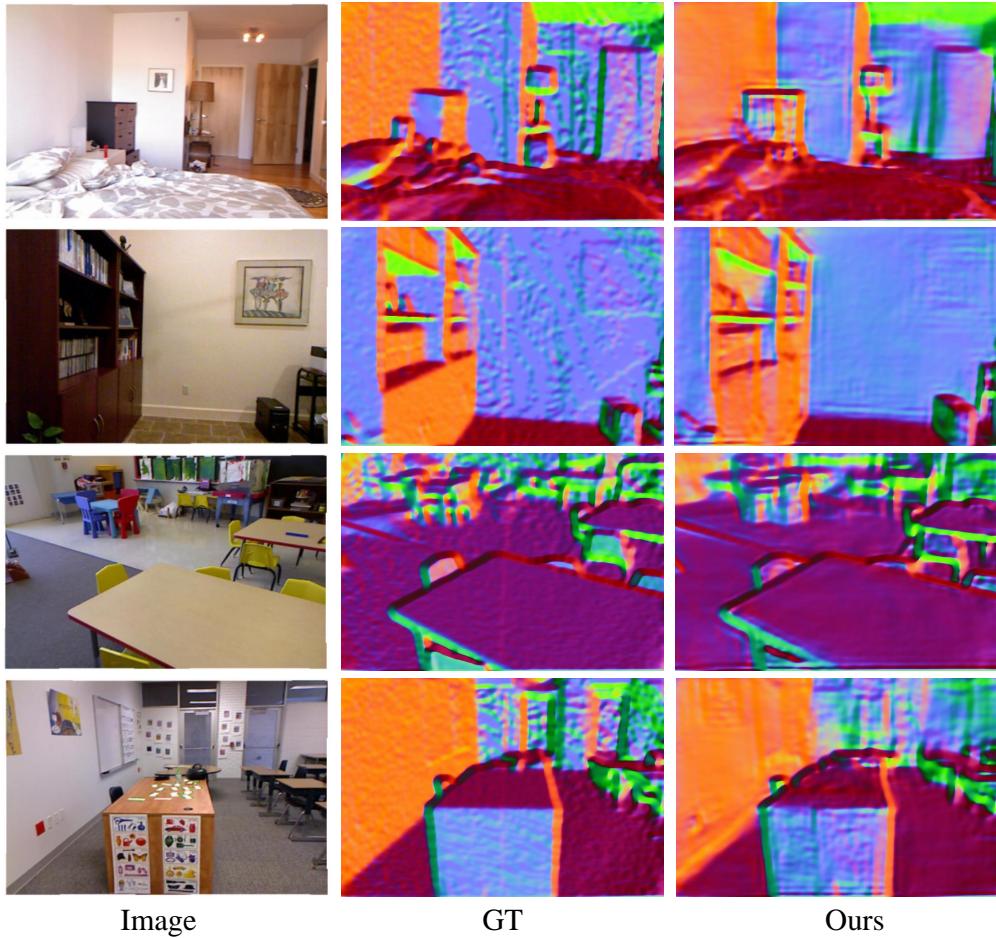


Figure 13. Samples of the recovered surface normals on NYUD-V2. One can see that we can recover surface normals from the reconstructed point cloud with high quality .

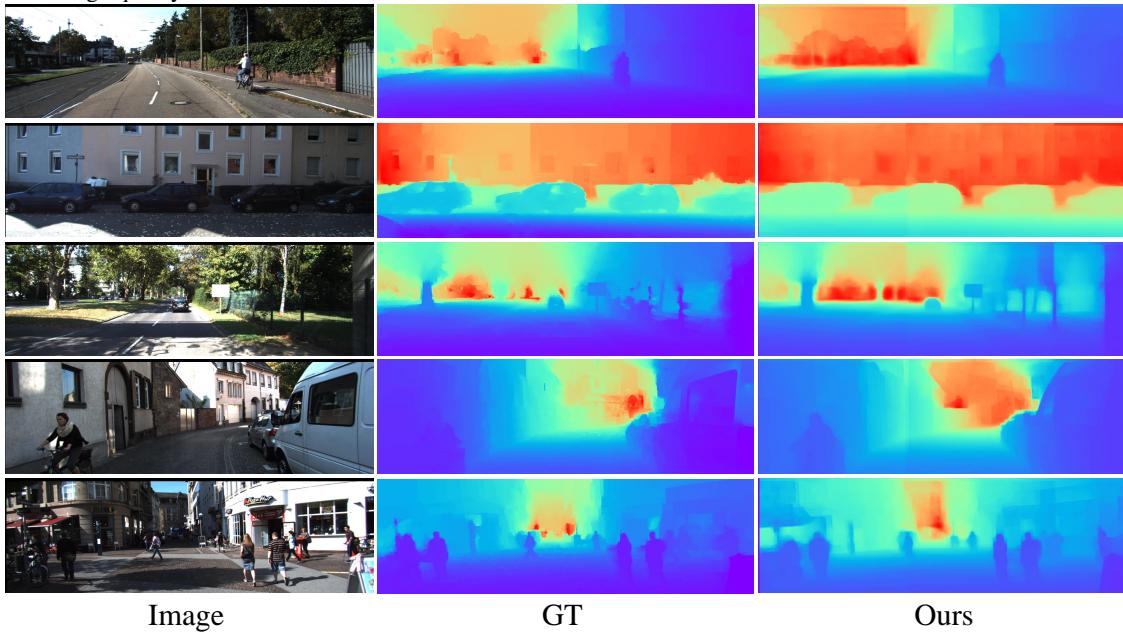


Figure 14. Samples of the predicted depth on KITTI. The high quality results show the effectiveness of our methods.

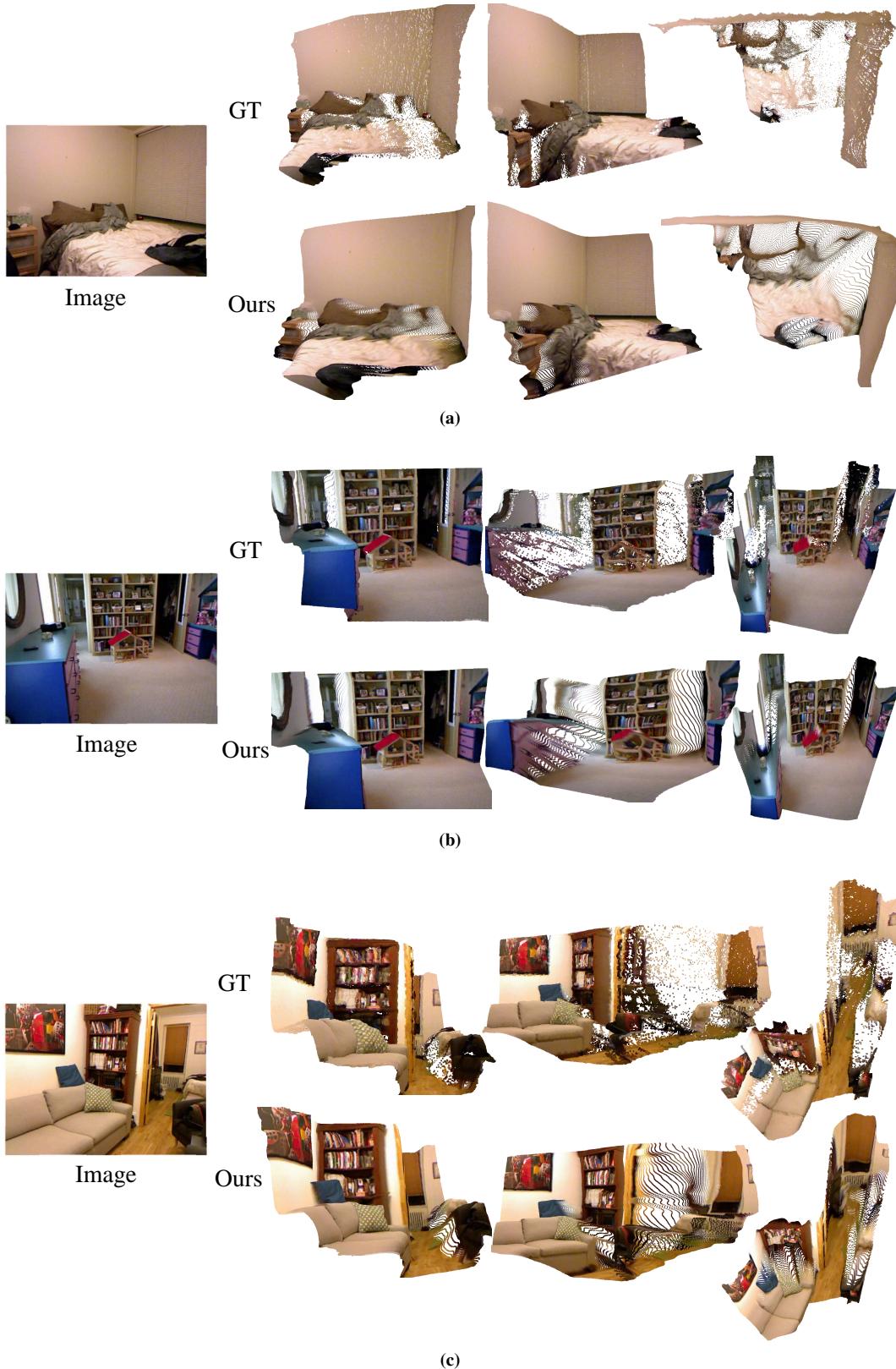


Figure 15. Reconstructed point clouds. Three scenes are randomly selected from NYUD-V2. For the reconstructed point cloud of each scene, 3 views are selected to demonstrate the point cloud. (a) Scene 1; (b) Scene 2; (c) Scene 3.

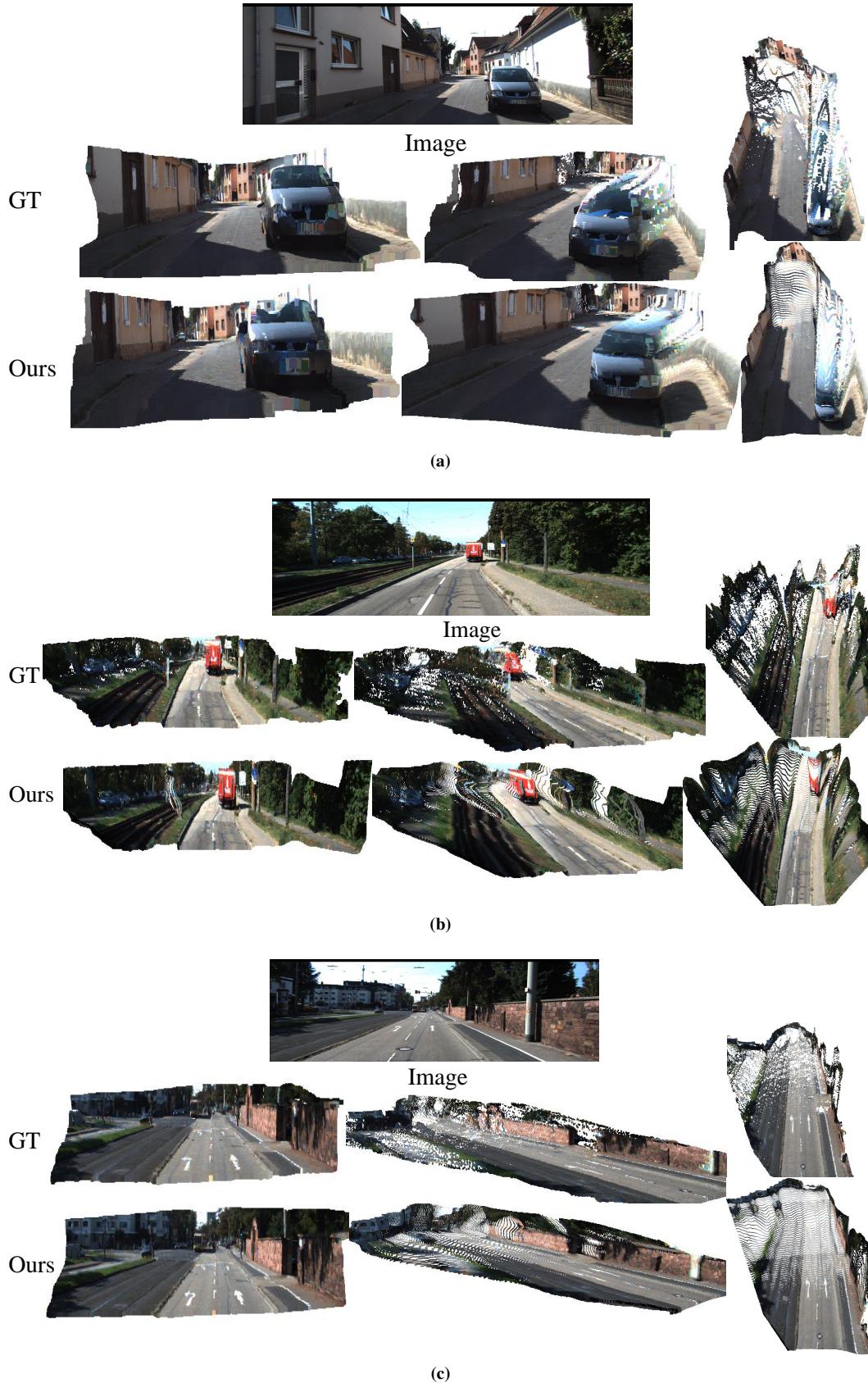


Figure 16. Reconstructed point clouds. Three scenes are randomly selected from KITTI. For the reconstructed point cloud of each scene, 3 views are selected to demonstrate the quality of the point cloud. (a) Scene 1; (b) Scene 2; (c) Scene 3.

- 32(11):1231–1237, 2013.
- [14] X. Guo, H. Li, S. Yi, J. Ren, and X. Wang. Learning monocular depth by distilling cross-domain stereo networks. In *Proc. Eur. Conf. Comp. Vis.*, pages 484–500, 2018.
- [15] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik. Learning rich features from rgb-d images for object detection and segmentation. In *Proc. Eur. Conf. Comp. Vis.*, pages 345–360. Springer, 2014.
- [16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 770–778, 2016.
- [17] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, and V. Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 858–865. IEEE, 2011.
- [18] J. Hu, M. Ozay, Y. Zhang, and T. Okatani. Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries. In *IEEE Winter Conf. on Applications of Comp. Vis.*, 2019.
- [19] J. Jiao, Y. Cao, Y. Song, and R. Lau. Look deeper into depth: Monocular depth estimation with semantic booster and attention-driven loss. In *Proc. Eur. Conf. Comp. Vis.*, pages 53–69, 2018.
- [20] K. Karsch, C. Liu, and S. B. Kang. Depth transfer: Depth extraction from video using non-parametric sampling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(11):2144–2158, 2014.
- [21] K. Klasing, D. Althoff, D. Wollherr, and M. Buss. Comparison of surface normal estimation methods for range sensing applications. In *IEEE Int. Conf. Robotics & Automation*, pages 3206–3211. IEEE, 2009.
- [22] Y. Kuznetsov, J. Stückler, and B. Leibe. Semi-supervised deep learning for monocular depth map prediction. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 2215–2223. IEEE, 2017.
- [23] L. Ladicky, J. Shi, and M. Pollefeys. Pulling things out of perspective. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 89–96, 2014.
- [24] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *Int. Conf. on 3D Vision*, pages 239–248. IEEE, 2016.
- [25] B. Li, C. Shen, Y. Dai, A. Van Den Hengel, and M. He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 1119–1127, 2015.
- [26] J. Li, R. Klein, and A. Yao. A two-streamed network for estimating fine-scaled depth maps from single rgb images. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 22–29, 2017.
- [27] R. Li, K. Xian, C. Shen, Z. Cao, H. Lu, and L. Hang. Deep attention-based classification network for robust depth prediction. In *arXiv: Comp. Res. Repository*, volume abs/1807.03959, 2018.
- [28] 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.*, 38(10):2024–2039, 2016.
- [29] M. Liu, M. Salzmann, and X. He. Discrete-continuous depth estimation from a single image. In *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 716–723, 2014.
- [30] V. Nekrasov, T. Dharmasiri, A. Spek, T. Drummond, C. Shen, and I. Reid. Real-time joint semantic segmentation and depth estimation using asymmetric annotations. In *arXiv: Comp. Res. Repository*, volume abs/1809.04766, 2018.
- [31] X. Qi, R. Liao, Z. Liu, R. Urtasun, and J. Jia. Geonet: Geometric neural network for joint depth and surface normal estimation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 283–291, 2018.
- [32] A. Roy and S. Todorovic. Monocular depth estimation using neural regression forest. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 5506–5514, 2016.
- [33] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz. Aligning point cloud views using persistent feature histograms. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots & Systems*, pages 3384–3391. IEEE, 2008.
- [34] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 4510–4520, 2018.
- [35] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(5):824–840, 2009.
- [36] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *Proc. Eur. Conf. Comp. Vis.*, pages 746–760. Springer, 2012.
- [37] A. Spek, T. Dharmasiri, and T. Drummond. Cream: Condensed real-time models for depth prediction using convolutional neural networks. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots & Systems*, pages 540–547. IEEE, 2018.
- [38] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. L. Yuille. Towards unified depth and semantic prediction from a single image. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 2800–2809, 2015.
- [39] X. Wang, D. Fouhey, and A. Gupta. Designing deep networks for surface normal estimation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 539–547, 2015.
- [40] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 5987–5995. IEEE, 2017.
- [41] W. Yin, X. Cheng, J. Xie, H. Cui, and Y. Chen. High-speed 3d profilometry employing hsi color model for color surface with discontinuities. *Elsevier Optics & Laser Technology.*, 96:81–87, 2017.
- [42] B. Zeisl, M. Pollefeys, et al. Discriminatively trained dense surface normal estimation. In *Proc. Eur. Conf. Comp. Vis.*, pages 468–484. Springer, 2014.
- [43] K. Zheng, Z.-J. Zha, Y. Cao, X. Chen, and F. Wu. La-net: Layout-aware dense network for monocular depth estimation. pages 1381–1388. ACM, 2018.