

# GeoNet: Geometric Neural Network for Joint Depth and Surface Normal Estimation

Xiaojuan Qi<sup>†</sup> Renjie Liao<sup>§,‡</sup> Zhengzhe Liu<sup>†</sup> Raquel Urtasun<sup>§,‡</sup> Jiaya Jia<sup>†,§</sup>

<sup>†</sup> The Chinese University of Hong Kong

<sup>‡</sup> University of Toronto

<sup>§</sup> Uber Advanced Technologies Group

<sup>§</sup> YouTu Lab, Tencent

## Abstract

In this paper, we propose Geometric Neural Network (GeoNet) to jointly predict depth and surface normal maps from a single image. Building on top of two-stream CNNs, our GeoNet incorporates geometric relation between depth and surface normal via the new depth-to-normal and normal-to-depth networks. Depth-to-normal network exploits the least square solution of surface normal from depth and improves its quality with a residual module. Normal-to-depth network, contrarily, refines the depth map based on the constraints from the surface normal through a kernel regression module, which has no parameter to learn. These two networks enforce the underlying model to efficiently predict depth and surface normal for high consistency and corresponding accuracy. Our experiments on NYU v2 dataset verify that our GeoNet is able to predict geometrically consistent depth and normal maps. It achieves top performance on surface normal estimation and is on par with state-of-the-art depth estimation methods.

## 1. Introduction

We tackle the important problem of joint estimation of depth and surface normal from a single RGB image. The 2.5D geometric information is beneficial to various computer vision tasks, including structure from motion (SfM), 3D reconstruction, pose estimation, object recognition, and scene classification.

There exist a large amount of methods on depth estimation [25, 19, 8, 7, 21, 31, 24, 16, 20, 34, 18] and surface normal estimation [7, 33, 3, 2, 18] from a single image. Among them, deep-neural-network-based methods achieve very promising results.

**Challenges** Albeit the great advancement in this field, we notice that most previous methods deal with depth and normal estimation independently, which possibly make their prediction inconsistent without considering the close underlying geometry relationship. For example, as demonstrated

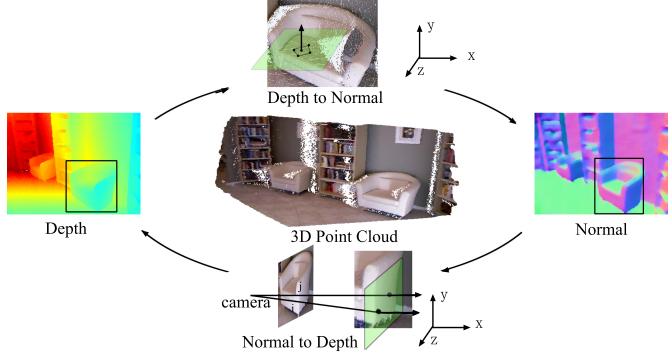


Figure 1: Geometric relationship of depth and surface normal. Surface normal can be estimated from 3D point cloud; depth is inferred from surface normal by solving linear equations.

in [32], the predicted depth map could be distorted in planar regions. It is thus intriguing to ask what if one considers the fact that surface normal does not change much in planar regions. This thought motivates us to design new models, which are exactly based on above simple fact and yet potentially show a vital direction in this field, to exploit the inevitable geometric relationship between depth and surface normal for more accurate estimation.

We use the example in Fig. 1 to illustrate the common-knowledge relation. On the one hand, surface normal is determined by local surface tangent plane of 3D points, which can be estimated from depth; on the other hand, depth is constrained by the local surface tangent plane determined by surface normal. Although it looks straightforward, it is not trivial to design neural networks to properly make use of these geometric conditions.

We note incorporating geometric relationship into traditional models via hand-crafted feature is already feasible, as explained in [25, 4]. However, there is no much research to make it happen in neural networks. One possible design is to build a convolutional neural network (CNN) to directly learn such geometric relationship from data. However, our

experiments in section 4.2 demonstrate that even with the common successful CNN architectures, *e.g.*, VGG-16, we cannot obtain any reasonable normal results from depth, not even close. It is found that training always converges to very poor local minima given carefully tuned architectures and hyper-parameters.

These extensive experiments manifest that current classification CNN architectures do *not* have the necessary ability to learn such geometric relationship from data. This finding motivates us to design specialized architecture to explicitly incorporate and enforce geometric conditions.

**Our Contributions** We in this paper propose the Geometric Neural Networks (GeoNet) to infer depth and surface normal in one unified system. The architecture of GeoNet involves a two-stream CNN, which predicts depth and surface normal from a single image respectively. The two networks manage the two streams to model the depth-to-normal and normal-to-depth mapping.

In particular, relying on least-square and residual modules, the depth-to-normal network effectively captures the geometric relationship. Normal-to-depth network updates estimates of depth via a kernel regression module; it does not require any parameters that should be learned. With these coupled networks, our GeoNet enforces the final prediction of depth and surface normal to follow the underlying conditions. Further, these two networks are computationally efficient since they do not have many parameters to learn.

Experimental results on NYU v2 dataset show that our GeoNet achieves state-of-the-art performance in terms of most of the evaluation metrics and is more efficient than other alternatives.

## 2. Related Work

2.5D geometry estimation from a single image has been intensively studied in past years. Previous work can be roughly divided into two categories.

Traditional methods did not use deep neural networks, and mainly focused on exploiting low-level image cues and geometric constraints. For example, the method of [30] estimates mean depth of the scene by recognizing the structures presented in the image, and inferring the scale of the scene. Based on Markov random fields (MRF), Saxena et al. [25] predicts a depth map given the hand-crafted features of a single image. Vanishing points and lines are utilized in [12] for recovering the surface layout.

Besides, Liu et al. [19] leveraged predicted labels of semantic segmentation to incorporate geometry constraints. A scale-dependent classifier was proposed in [15] to jointly learn semantic segmentation and depth estimation. Shi et al. [27] showed that estimating the defocus blur is beneficial for recovering the depth map. In [4], a unified optimization problem was formed, which aims at recovering the intrinsic scene

property, *e.g.*, shape, illumination, and reflectance from shading. Relying on specially designed features, above methods directly incorporate geometric constraints. However, their model capacity and generality may be unsatisfactory to deal with different types of images.

With deep learning, many methods were recently proposed for single-image depth or/and surface normal prediction. Eigen et al. [8] directly predicted the depth map by feeding the image to CNNs. Shelhamer et al. [26] proposed a fully convolutional network (FCN) based solution to learn the full intrinsic decomposition of a single image, which involves inferring the depth map as the first intermediate step. In [7], a unified coarse-to-fine hierarchical network was adopted for depth/normal prediction.

For predicting single-image surface normal, Wang et al. [33] incorporated local, global, and vanishing point information in designing the network architecture. In [20], a continuous conditional random field (CRF) was built on top of CNN to smooth super-pixel-based depth prediction. There is also a skip-connected architecture [3] to fuse hidden representations of different layers for surface normal estimation.

All these methods regard depth and surface normal prediction as independent tasks, thus ignoring their basic geometric relationship. The most related work to ours is that of [32], which designed a CRF with a 4-stream CNN, considering the consistency of predicted depth and surface normal in planar regions. Nevertheless, it may fail when planar regions are uncommon in images. In comparison, our GeoNet exploits the geometric relationship between depth and surface normal for *general* situations without making any planar or curvature assumptions. It is not limited to particular types of regions, and is computationally efficient.

## 3. Geometric Neural Networks

In this section, we first introduce the depth-to-normal network, which refines the surface normal from the given depth map. Then we explain the normal-to-depth network to update depth from the given surface normal map. It is followed by the overall architecture of our GeoNet, which utilizes these new modules.

### 3.1. Depth-to-Normal Network

As aforementioned, learning geometrically consistent surface normal from depth via directly applying neural networks is surprisingly hard. Inspired from the geometry-based solution [9], we propose a novel neural network architecture, which takes initial surface normal and depth maps as input and predicts a better surface normal. We start with introducing the geometric model, which can be viewed as a fix-weight neural network. Then we explain the residual module that aims at smoothing and combining the predictions of surface normal.

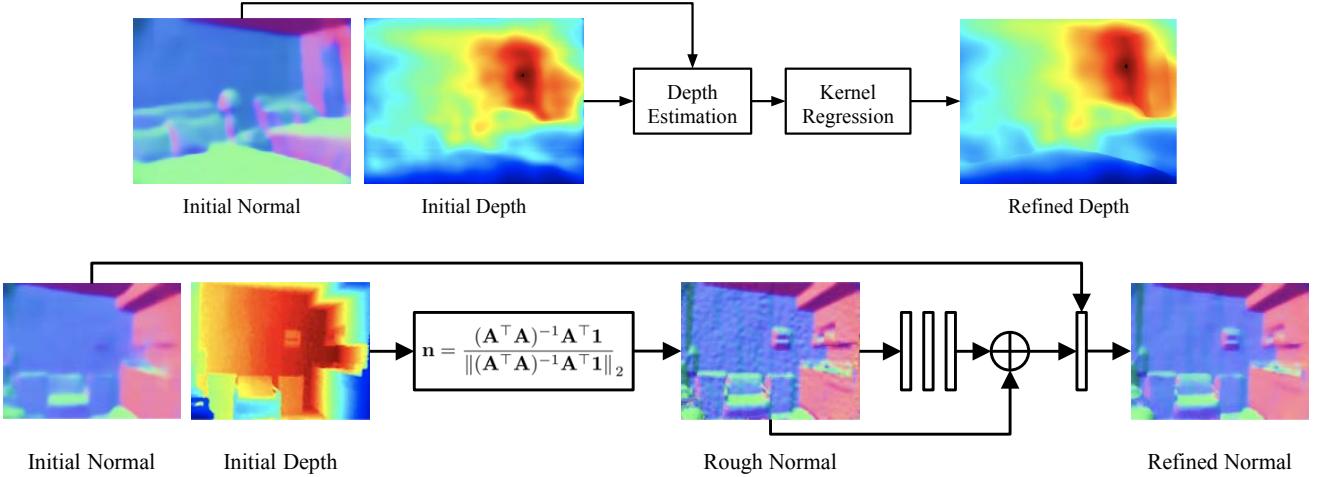


Figure 2: Upper row: normal-to-depth network. Bottom row: depth-to-normal network.

**Pinhole Camera Model** As a common practice, the pin-hole camera model is adopted. We denote  $(u_i, v_i)$  as the location of pixel  $i$  in the 2D image. Its corresponding location in 3D space is  $(x_i, y_i, z_i)$ , where  $z_i$  is the depth. Based on the geometry of perspective projection, we obtain

$$\begin{aligned} x_i &= (u_i - c_x) * z_i / f_x, \\ y_i &= (v_i - c_y) * z_i / f_y, \end{aligned} \quad (1)$$

where  $f_x$  and  $f_y$  are the focal length along the  $x$  and  $y$  directions respectively.  $c_x$  and  $c_y$  are coordinates of the principal points.

**Least Square Module** Following [9], we formulate inference of surface normal from the depth map as a least square problem. Specifically, for any pixel  $i$ , given its depth  $z_i$ , we first compute its 3D coordinates  $(x_i, y_i, z_i)$  from its 2D coordinates  $(u_i, v_i)$  relying on the pinhole camera model. In order to compute the surface normal of pixel  $i$ , we need to determine the tangent plane, which crosses pixel  $i$  in 3D space. We follow traditional assumption that pixels within a local neighborhood of pixel  $i$  lie on the same tangent plane. In particular, we define the set of neighboring pixels, including pixel  $i$  itself, as

$$\mathcal{N}_i = \{(x_j, y_j, z_j) | |u_i - u_j| < \beta, |v_i - v_j| < \beta, |z_i - z_j| < \gamma z_i\}, \quad (2)$$

where  $\beta$  and  $\gamma$  are hyper-parameters controlling the size of neighborhood along  $x$ - $y$  and depth axes respectively. With these pixels on the tangent plane, the surface normal estimate  $\mathbf{n} = [n_x, n_y, n_z]$  should satisfy the over-determined linear system of

$$\mathbf{A}\mathbf{n} = \mathbf{b}, \quad \text{subject to } \|\mathbf{n}\|_2^2 = 1. \quad (3)$$

where

$$\mathbf{A} = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_K & y_K & z_K \end{bmatrix} \in R^{K \times 3}, \quad (4)$$

and  $\mathbf{b} \in R^{K \times 1}$  is a constant vector.  $K$  is the size of  $\mathcal{N}_i$ . The least square solution of this problem, which minimizes  $\|\mathbf{A}\mathbf{n} - \mathbf{b}\|^2$  has the closed form of

$$\mathbf{n} = \frac{(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{1}}{\|(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{1}\|_2}, \quad (5)$$

where  $\mathbf{1} \in R^K$  is a vector with all 1 elements. It is not surprising that Eq. (5) can be regarded as a fix-weight neural network, which predicts surface normal given the depth map.

**Residual Module** This least square module occasionally produces noisy estimate of surface normal due to noise and other image issues. A rough normal map is shown in Fig. 2. To improve accuracy, we propose a residual module, which consists of a 3-layer CNN with skip-connection and  $1 \times 1$  convolutional layer, as shown in Fig. 2. The goal is to smooth out noise and combine the initial guess of surface normal to further enhance the quality. In particular, before fed to the  $1 \times 1$  convolution, the output of this CNN is concatenated with initial estimation of surface normal, which could be output of another network.

The architecture of this depth-to-normal network is illustrated in the bottom row of Fig. 2. By explicitly leveraging the geometric relationship between depth and surface normal, our network circumvents the aforementioned difficulty in learning geometrically consistent surface normal. It is computationally efficient since the least-square module is

just a fix-weight layer. The extra important benefit stems from using ground-truth depth as the input to pre-train the network. It permits concatenation and joint fine-tuning with other networks, which predict depth maps from raw images.

### 3.2. Normal-to-Depth Network

Now we turn to the normal-to-depth network. For any pixel  $i$ , given its surface normal  $(n_{ix}, n_{iy}, n_{iz})$  and an initial estimate of depth  $z_i$ , the goal is to refine depth.

First, note that given the 3D point  $(x_i, y_i, z_i)$  and its surface normal  $(n_{ix}, n_{iy}, n_{iz})$ , we can uniquely determine the tangent plane  $\mathcal{P}_i$ , which satisfies the equation of

$$n_{ix}(x - x_i) + n_{iy}(y - y_i) + n_{iz}(z - z_i) = 0. \quad (6)$$

As explained in section 3.1, we can still assume that pixels within a small neighborhood of pixel  $i$  lie on this tangent plane  $\mathcal{P}_i$ , as shown in Fig. 2 (bottom row). This neighborhood  $\mathcal{M}_i$  is defined as

$$\mathcal{M}_i = \{(x_j, y_j, z_j) | \mathbf{n}_j^\top \mathbf{n}_i > \alpha, |u_i - u_j| < \beta, |v_i - v_j| < \beta\},$$

where  $\beta$  is the hyper-parameter to control the size of neighborhood along  $x - y$  axes.  $\alpha$  is a threshold to rule out spatially close points, which are not approximately coplanar.  $(u_i, v_i)$  are the coordinates of pixel  $i$  in the 2D image.

For any pixel  $j \in \mathcal{M}_i$ , if we assume its depth  $z_j$  is accurate, we can compute the depth estimate of pixel  $i$  as  $z'_{ji}$  relying on Eqs. (1) and (6). It is expressed as

$$z'_{ji} = \frac{n_{ix}x_j + n_{iy}y_j + n_{iz}z_j}{(u_i - c_x)n_{ix}/f_x + (v_i - c_y)n_{iy}/f_y + n_{iz}}. \quad (7)$$

After getting it, to refine depth of pixel  $i$ , we use kernel regression to aggregate estimation from all pixels in the neighborhood as

$$\hat{z}_i = \frac{\sum_{j \in \mathcal{M}_i} \mathbf{K}(\mathbf{n}_j, \mathbf{n}_i) z'_{ji}}{\sum_{j \in \mathcal{M}_i} \mathbf{K}(\mathbf{n}_j, \mathbf{n}_i)}, \quad (8)$$

where  $\hat{z}_i$  is the refined depth,  $\mathbf{n}_i = [n_{ix}, n_{iy}, n_{iz}]$  and  $\mathbf{K}$  is the kernel function. We use linear kernel due to its simplicity, *i.e.*,  $\mathbf{K}(\mathbf{n}_j, \mathbf{n}_i) = \mathbf{n}_j^\top \mathbf{n}_i$ . In this case, the smaller the angle between normals  $\mathbf{n}_i$  and  $\mathbf{n}_j$  is, which means higher probability that pixels  $i$  and  $j$  are in the same tangent plane, the more accurate and important the estimate  $z'_{ji}$  is.

The above process is illustrated in the upper row of Fig. 2. It can be viewed as a voting process where every pixel  $j \in \mathcal{M}_i$  gives a “vote” to determine the depth of pixel  $i$ . By utilizing the geometric relationship between surface normal and depth, we efficiently improve the quality of depth estimate without the need to learn any weights.

### 3.3. GeoNet

**Full Architecture** With above two networks, we now explain our full model illustrated in Fig. 3. We first use two-stream CNNs to predict the initial depth and surface normal

maps, as shown in Fig. 3(a) and (b) respectively. The fundamental structures we adopted are (1) VGG-16 [29] and (2) ResNet-50 [11].

Based on the initial depth map predicted by one CNN, we apply the depth-to-normal network explained in Section 3.1 to refine normal as shown in Fig. 3(c). Similarly, as shown in Fig. 3(d), given the surface normal estimate, we refine depth using the normal-to-depth network described in Section 3.2. We pre-train the depth-to-normal network taking ground-truth depth as input. For the normal-to-depth network, we do not need to learn any weights.

**Loss Functions** We now explain the loss functions associated with our GeoNet. For pixel  $i$ , we denote the initial, refined and ground-truth depth as  $z_i$ ,  $\hat{z}_i$  and  $z_i^{\text{gt}}$  respectively. Similarly, we have these classes of surface normal as  $\mathbf{n}_i$ ,  $\hat{\mathbf{n}}_i$  and  $\mathbf{n}_i^{\text{gt}}$  respectively. The total number of pixels is  $M$ .

The overall loss function is the summation of two terms, *i.e.*,  $L = l_{\text{depth}} + l_{\text{normal}}$ . The depth loss  $l_{\text{depth}}$  is expressed as

$$l_{\text{depth}} = \frac{1}{M} \left( \sum_i \|z_i - z_i^{\text{gt}}\|_2^2 + \eta \sum_i \|\hat{z}_i - z_i^{\text{gt}}\|_2^2 \right).$$

The surface normal loss  $l_{\text{normal}}$  is

$$l_{\text{normal}} = \frac{1}{M} \left( \sum_i \|\mathbf{n}_i - \mathbf{n}_i^{\text{gt}}\|_2^2 + \lambda \sum_i \|\hat{\mathbf{n}}_i - \mathbf{n}_i^{\text{gt}}\|_2^2 \right).$$

Here  $\lambda$  and  $\eta$  are hyper-parameters to balance contribution of different terms. The final predictions of our GeoNet are the optimized depth and surface normal estimates. GeoNet is trained by back-propagation in an end-to-end manner.

## 4. Experiments

We evaluate the effectiveness of our method on the NYU v2 dataset [28]. It contains 464 video sequences of indoor scenes, which are further divided into 249 sequences for training and 215 for testing. We sample 30,816 frames from the training video sequences as the training data. Note that the methods of [7], [34] and [16] used 120K, 90K and 95K data for training, which are all significantly more than ours.

For the training set, we use the inpainting method of [17] to fill in invalid or missing pixels in the ground-truth depth maps. Then we generate ground-truth surface normal maps following the procedure of [9]. Our GeoNet is implemented in TensorFlow.

We initialize the two-stream CNNs with networks pre-trained on ImageNet. In particular, we try two different choices. The first is a modified VGG-16 network based on FCN [23] with dilated convolutions [6, 35] and global pooling [22]. The second is a ResNet-50 following the model of [16]. We use Adam [14] to optimize the network and clip the norm of gradients so that they are no larger than 5. The

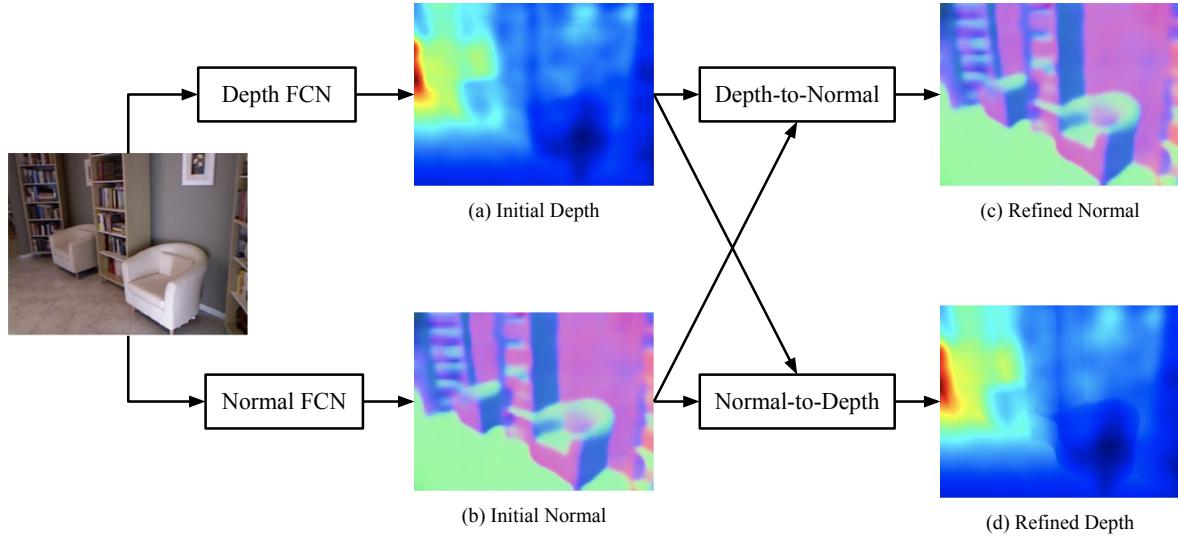


Figure 3: Overall framework of our Geometric Neural Networks.

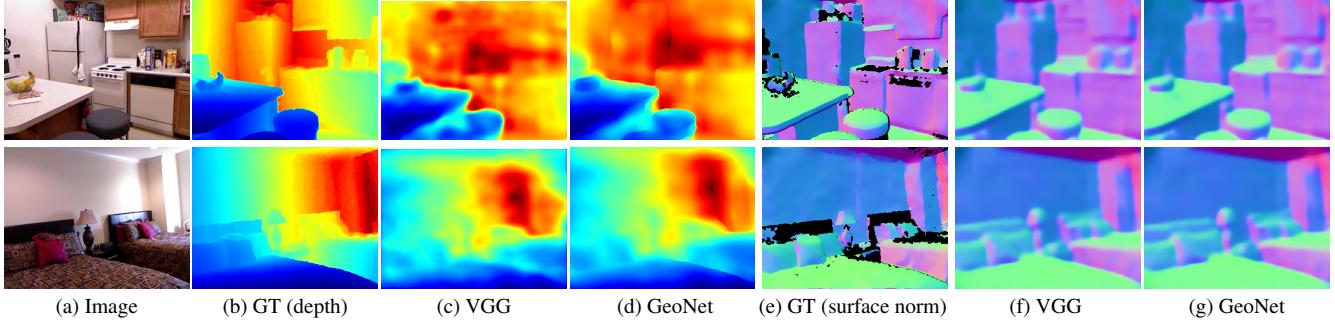


Figure 4: Visual comparison on joint prediction with VGG-16 as backbone architecture. GT stands for “ground truth”.

initial learning rate is  $1e^{-4}$  and is adjusted following the polynomial decay strategy with the power parameter 0.9. Random horizontal flip is utilized to augment training data. While flipping images, we multiply the corresponding x-direction of surface normal maps with  $-1$ .

The whole system is trained with batch-size 4 for 40,000 iterations. Hyper-parameters  $\{\alpha, \beta, \gamma, \lambda, \eta\}$  are set to  $\{0.95, 9, 0.05, 0.01, 0.5\}$  according to validation on a 5% randomly split training data.  $\lambda$  is set to a small value due to numerical instability when computing the matrix inverse in the least square module – gradient of Eq. (5) needs inverse of matrix  $A^T A$ , which might be erroneous if the condition number is small. Setting  $\lambda = 0.01$  mitigates this effect.

Following [8, 16, 34], we adopt four metrics to evaluate resulting depth map quantitatively. They are root mean square error (rmse), mean log 10 error (log 10), mean relative error (rel), and pixel accuracy as percentage of pixels with  $\max(z_i/z_i^{gt}, z_i^{gt}/z_i) < \delta$  for  $\delta \in [1.25, 1.25^2, 1.25^3]$ . The evaluation metrics for surface normal prediction [33, 3, 7] are mean of angle error (mean), medians of the angle error

(median), root mean square error (rmse), and pixel accuracy as percentage of pixels with angle error below threshold  $t$  where  $t \in [11.25^\circ, 22.5^\circ, 30^\circ]$ .

#### 4.1. Comparison with State-of-the-Art

In this section, we compare our GeoNet with existing methods in terms of depth and/or surface normal prediction.

**Surface Normal Prediction** For surface normal prediction, the results are listed in Table 1. Our GeoNet consistently outperforms previous approaches regarding all different metrics. Note that since we use the same backbone network architecture VGG-16, the improvement stems from our depth-to-normal network, which effectively correct errors during estimation.

**Depth Prediction** In the task of depth prediction, since most state-of-the-art methods adopt either backbone network between VGG-16 and ResNet-50, we thus conduct experiments under both settings. The complete results are shown

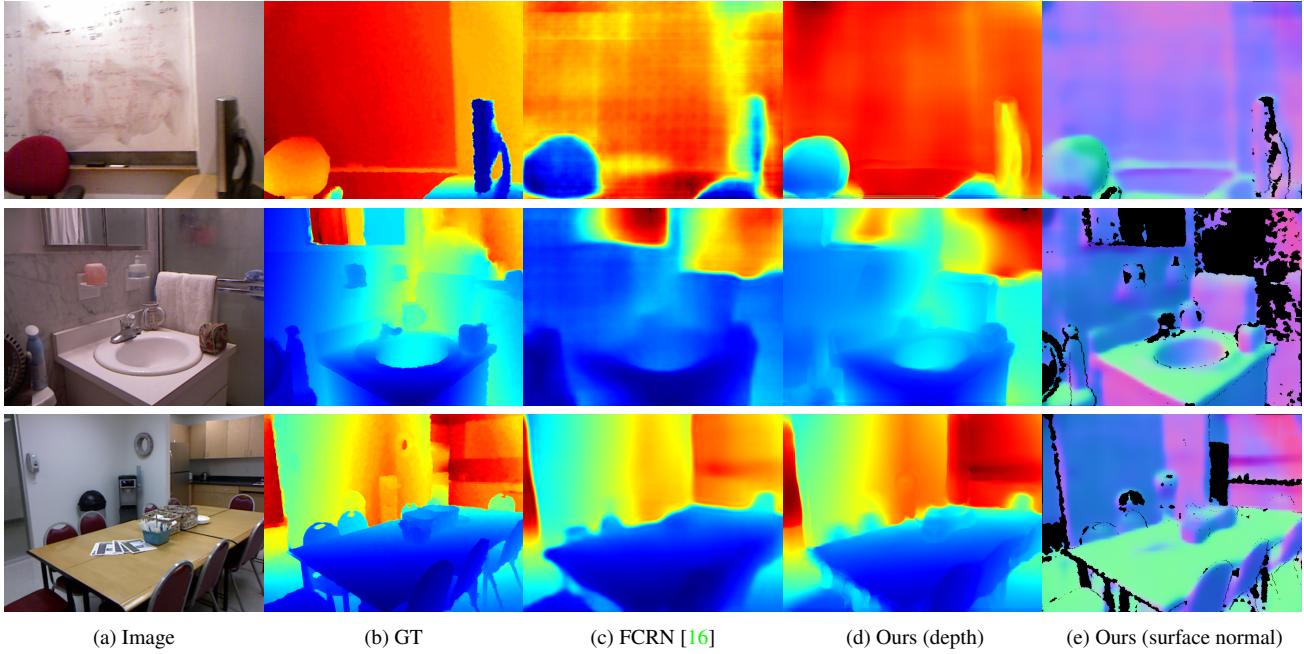


Figure 5: Visual comparison on depth prediction with ResNet-50 as backbone architecture. GT stands for “ground truth”.

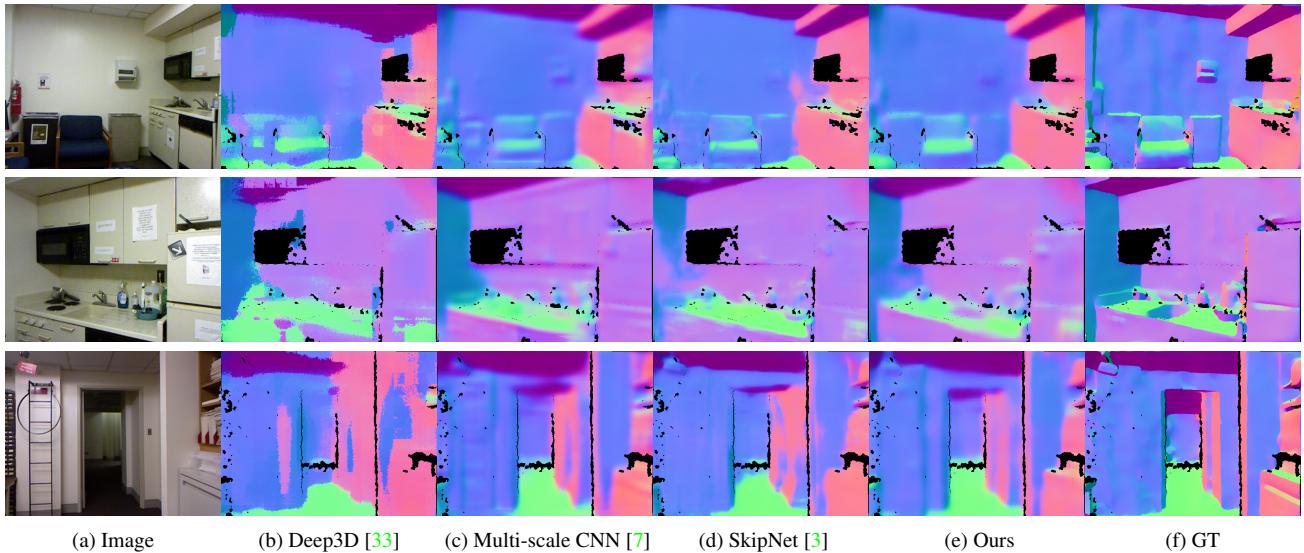


Figure 6: Visual comparison on surface normal prediction with VGG-16 being the backbone architecture. GT stands for “ground truth”.

in Table 2. Our GeoNet performs again better than state-of-the-art methods on 4 out of total 6 evaluation metrics. It performs comparably on the remaining two. Among all these methods, SURGE [32] is the only one, which shares the same objective – that is, jointly predicting depth and surface normal. It builds CRFs on top of a VGG-16 network. Using the same backbone network, as summarized in the table, our GeoNet significantly outperforms it. It is because

our model does not impose special assumptions on surface shape and underlying geometry.

**Visual Comparisons** We show visual examples of predicted depth and surface normal maps. First, in Fig. 5, we show visual comparisons with state-of-the-art method FCRN [16] on depth prediction. Our GeoNet generates more accurate depth maps with regard to the washbasin and small

	Error			Accuracy		
	mean	median	rmse	11.25°	22.5°	30°
3DP [9]	35.3	31.2	-	16.4	36.6	48.2
3DP (MW) [9]	36.3	19.2	-	39.2	52.9	57.8
UNFOLD [10]	35.2	17.9	-	40.5	54.1	58.9
Discr. [36]	33.5	23.1	-	27.7	49.0	58.7
Multi-scale CNN [7]	23.7	15.5	-	39.2	62.0	71.1
Deep3D [33]	26.9	14.8	-	42.0	61.2	68.2
SkipNet [3]	19.8	12.0	28.2	47.9	70.0	77.8
SURGE [32]	20.6	12.2	-	47.3	68.9	76.6
Baseline	19.4	12.5	27.0	46.0	70.3	78.9
SkipNet [3]+GeoNet	19.7	11.7	28.4	<b>48.8</b>	70.5	78.2
GeoNet	<b>19.0</b>	<b>11.8</b>	<b>26.9</b>	48.4	<b>71.5</b>	<b>79.5</b>

Table 1: Performance of surface normal prediction on NYU v2 test set. ‘‘Baseline’’ refers to using VGG-16 network with global pooling to directly predict surface normal from raw images. ‘‘SkipNet [3]+GeoNet’’ means building GeoNet on top of the normal result of [3].

	Error			Accuracy		
	rmse	log 10	rel	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
DepthTransfer [13]	1.214	-	0.349	0.447	0.745	0.897
SemanticDepth [15]	-	-	-	0.542	0.829	0.941
DC-depth [21]	1.06	0.127	0.335	-	-	-
Global-Depth [37]	1.04	0.122	0.305	0.525), 829	0.941	
CNN + HCRF [31]	0.907	-	0.215	0.605	0.890	0.970
Multi-scale CNN [7]	0.641	-	0.158	0.769	0.950	0.988
NRF [24]	0.744	0.078	0.187	0.801	0.950	0.986
Local Network [5]	0.620	-	0.149	0.806	0.958	0.987
SURGE [32]	0.643	-	0.156	0.768	0.951	0.989
GCL/RCL [1]	0.802	-	-	0.605	0.890	0.970
FCRN [16]	0.790	0.083	0.194	0.629	0.889	0.971
VGG+Multi-scale CRF [34]	0.584	0.059	0.136	0.822	0.955	0.971
ResNet+Multi-scale CRF [34]	0.586	<b>0.052</b>	<b>0.121</b>	0.811	0.954	0.988
Baseline	0.626	0.068	0.155	0.768	0.951	0.988
GeoNet-VGG	0.608	0.065	0.149	0.786	0.956	0.990
GeoNet-ResNet	<b>0.569</b>	0.057	0.128	<b>0.834</b>	<b>0.960</b>	<b>0.990</b>

Table 2: Performance of depth prediction on NYU v2 test set. ‘‘Baseline’’ means using VGG-16 to directly predict depth from raw images. VGG and ResNet are short for VGG-16 and ResNet-50 respectively.

objects on the table in the 2nd and 3rd rows respectively.

We also show the corresponding predictions of surface normal to verify that our GeoNet takes the advantage of surface normal to improve depth. The usefulness is illustrated regarding the whiteboard in the 1st row. 3D visualization of our depth prediction is shown in Fig. 7. The wall region of our prediction is much smoother than previous state-of-the-art FRCN [16], manifesting the necessity of incorporating geometric consistency.

Moreover, we compare results with those of other methods, including Deep3D [33], Multi-scale CNN [7] and SkipNet [3] on surface normal prediction in Fig. 6. GeoNet actually can produce results with better details on, for ex-

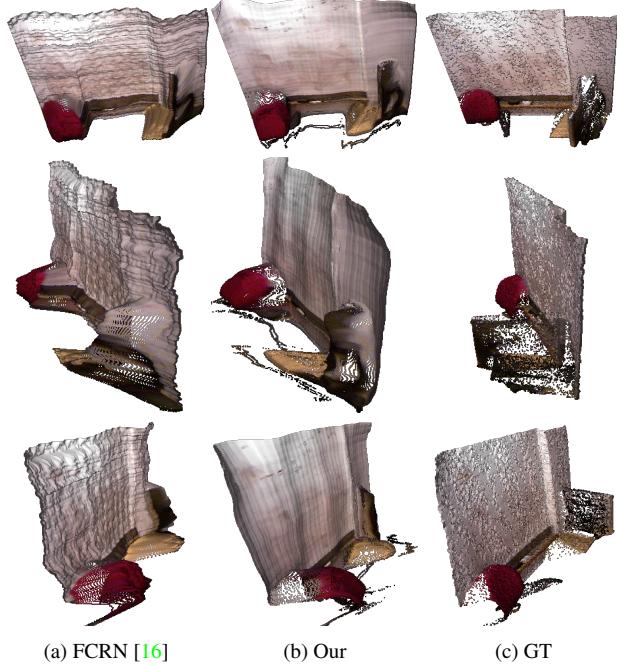


Figure 7: 3D visualization of point cloud with depth from FCRN [16], our prediction and ground truth. Each row shows the point cloud observed from one viewpoint.

ample, the chair, washbasin and wall from the 1st, 2nd, 3rd rows respectively. More results of joint prediction are shown in Fig. 4. From these figures, it is clear that our GeoNet does a much better job in terms of geometry estimation compared with the baseline VGG-16 network, which was not designed for this task in the first place.

**Running-time Comparison** We test our GeoNet on a PC with Intel i7-6950 CPU and a single TitanX GPU. When taking VGG-16 as the backbone network, our GeoNet obtains both surface normal and depth using 0.87s for an image with size  $480 \times 640$ . In comparison, *Local Network* [5] takes around 24s to predict the depth map of the same-sized image; SURGE [32]<sup>1</sup> also takes a lot of time due to the fact that it has to go through the forward-pass 10 times on the same VGG-16 network and it needs the inference of CRFs.

## 4.2. CNNs and Geometric Conditions

In this section, we verify our motivation through experiments and evaluate if previous CNNs can directly learn a mapping from depth to surface normal, implicitly following the geometric relationship.

To this end, we train CNNs, which take ground-truth depth and surface normal maps as input and supervision respectively. We tried different architectures, which include

<sup>1</sup>We do not have exact time without available public code.

	Error			Accuracy		
	mean	median	rmse	11.25°	22.5°	30°
4-layer	39.5	37.6	44.0	6.1	21.4	35.5
7-layer	39.8	38.2	44.3	6.5	21.0	34.2
VGG	47.8	47.3	52.1	2.8	11.8	20.7
LS	11.5	6.4	18.8	70.0	86.7	91.3
D-N	<b>8.2</b>	<b>3.0</b>	<b>15.5</b>	<b>80.0</b>	<b>90.3</b>	<b>93.5</b>

Table 3: Performance evaluation of depth-to-normal on NYU v2 test set. VGG stands for VGG-16 network. LS means our least square module. D-N is our depth-to-normal network without the last  $1 \times 1$  convolution layer. Ground-truth depth maps are used as input.

the first 4 layers of VGG-16, the first 7 layers of VGG-16, and full VGG-16 network. Before fed it to networks, the depth map is transformed into a 3-channel image encoding  $\{x, y, z\}$  coordinates respectively.

We provide the test performance on NYU v2 dataset in Table 3. All alternatives converge to very poor local minima. For fair comparison and clear illustration, we provide the test performance of surface normal predicted by our depth-to-normal network *without* combining the initial surface normal estimation. In particular, since the depth-to-normal network contains least-square and residual modules, we also show the surface normal map predicted by the least square module *only*, denoting as “LS”. The table reveals that LS module is already significantly better than the vanilla CNN baselines in all aspects. Moreover, with the residual module, our depth-to-normal network accomplishes superior results compared to using the least-square module alone.

These experiments preliminarily lead us to the following important findings.

1. Learning a mapping from depth to normal directly via vanilla CNNs hardly respects the underlying geometric relationship.
2. Despite its simplicity, the least square module is very effective in incorporating geometric conditions into neural networks, thus leading to better performance.
3. Our overall depth-to-normal network further improves the quality of normal prediction compared to the single least-square module.

### 4.3. Geometric Consistency

In this section, we verify if the predictions of depth and surface normal maps made by our GeoNet are consistent. To this end, we first pre-trained our depth-to-normal network without the last  $1 \times 1$  convolution layer using ground-truth depth and surface normal maps and regard it as an accurate

	Error			Accuracy		
	mean	median	rmse	11.25°	22.5°	30°
Pred-Baseline	42.2	39.8	48.9	9.8	25.2	35.9
Pred-GeoNet	<b>34.9</b>	<b>31.4</b>	<b>41.4</b>	<b>15.3</b>	<b>35.0</b>	<b>47.7</b>
GT-Baseline	47.8	47.3	52.1	2.8	11.8	20.7
GT-GeoNet	<b>36.8</b>	<b>32.1</b>	<b>44.5</b>	<b>15.0</b>	<b>34.5</b>	<b>46.7</b>

Table 4: Depth-to-normal consistency evaluation on the NYU v2 test set. “Pred” means that we transform predicted depth to surface normal and compare it with the predicted surface normal. “GT” means that we transform predicted depth to surface normal and compare it with the ground-truth surface normal. “Baseline” and “GeoNet” indicate that predictions are from baseline and our model respectively. The backbone network of baseline is VGG-16.

transformation. Given the predicted depth map, we compute the transformed surface normal map using the pre-trained network.

With these preparations, we compare error and accuracy under the following 4 settings. (1) Metrics between transformed and predicted normal (depth and surface normals generated by baseline CNNs). (2) Metrics between transformed and predicted normal (depth and surface normals generated by our GeoNet). (3) Metrics between transformed and ground-truth normal (depth generated by baseline CNNs). (4) Metrics between transformed and ground-truth normal (depth generated by our GeoNet). Here we also use the VGG-16 network as the baseline CNN.

The results are shown in Table 4. The “Pred” columns of the table show that our GeoNet can generate predictions of depth and surface normal more consistent than those of the baseline CNNs. From the “GT” columns of the table, it is also obvious that, compared to the baseline CNN, the predictions yielded from our GeoNet are consistently closer to the ground truth.

## 5. Conclusion

In this paper, we propose Geometric Neural Networks (GeoNet) to jointly predict depth and surface normal from a single image. Our GeoNet involves depth-to-normal and normal-to-depth networks. It effectively enforces the geometric conditions that computation should obey regarding depth and surface normal. They make the final prediction geometrically consistent and more accurate. Our extensive experiments show that GeoNet achieves state-of-the-art performance.

In the future, we would like to apply our GeoNet to tasks with inherent lighting and color constraints, such as intrinsic image decomposition and 3D reconstruction.

## References

- [1] M. H. Baig and L. Torresani. Coupled depth learning. In *WACV*, 2016. 7
- [2] A. Bansal, X. Chen, B. Russell, A. G. Ramanan, et al. Pixelnet: Representation of the pixels, by the pixels, and for the pixels. *arXiv*, 2017. 1
- [3] A. Bansal, B. Russell, and A. Gupta. Marr revisited: 2d-3d alignment via surface normal prediction. In *CVPR*, 2016. 1, 2, 5, 6, 7
- [4] J. T. Barron and J. Malik. Shape, illumination, and reflectance from shading. *PAMI*, 37(8):1670–1687, 2015. 1, 2
- [5] A. Chakrabarti, J. Shao, and G. Shakhnarovich. Depth from a single image by harmonizing overcomplete local network predictions. In *NIPS*, 2016. 7
- [6] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv*, 2014. 4
- [7] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015. 1, 2, 4, 5, 6, 7
- [8] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*, 2014. 1, 2, 5
- [9] D. F. Fouhey, A. Gupta, and M. Hebert. Data-driven 3d primitives for single image understanding. In *ICCV*, 2013. 2, 3, 4, 7
- [10] D. F. Fouhey, A. Gupta, and M. Hebert. Unfolding an indoor origami world. In *ECCV*, pages 687–702. Springer, 2014. 7
- [11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4
- [12] D. Hoiem, A. A. Efros, and M. Hebert. Recovering surface layout from an image. *IJCV*, 2007. 2
- [13] K. Karsch, C. Liu, and S. B. Kang. Depth extraction from video using non-parametric sampling. In *ECCV*, pages 775–788. Springer, 2012. 7
- [14] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv*, 2014. 4
- [15] L. Ladicky, J. Shi, and M. Pollefeys. Pulling things out of perspective. In *CVPR*, pages 89–96, 2014. 2, 7
- [16] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *3DV*, 2016. 1, 4, 5, 6, 7
- [17] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. In *ToG*, 2004. 4
- [18] 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 *CVPR*, 2015. 1
- [19] B. Liu, S. Gould, and D. Koller. Single image depth estimation from predicted semantic labels. In *CVPR*, pages 1253–1260, 2010. 1, 2
- [20] F. Liu, C. Shen, G. Lin, and I. Reid. Learning depth from single monocular images using deep convolutional neural fields. *PAMI*, 2016. 1, 2
- [21] M. Liu, M. Salzmann, and X. He. Discrete-continuous depth estimation from a single image. In *ICCV*, 2014. 1, 7
- [22] W. Liu, A. Rabinovich, and A. C. Berg. Parsenet: Looking wider to see better. *arXiv*, 2015. 4
- [23] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 4
- [24] A. Roy and S. Todorovic. Monocular depth estimation using neural regression forest. In *CVPR*, 2016. 1, 7
- [25] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In *NIPS*, pages 1161–1168, 2006. 1, 2
- [26] E. Shelhamer, J. T. Barron, and T. Darrell. Scene intrinsics and depth from a single image. In *ICCV Workshops*, pages 37–44, 2015. 2
- [27] J. Shi, X. Tao, L. Xu, and J. Jia. Break ames room illusion: depth from general single images. *SIGGRAPH*, 2015. 2
- [28] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. *ECCV*, 2012. 4
- [29] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv*, 2014. 4
- [30] A. Torralba and A. Oliva. Depth estimation from image structure. *PAMI*, 24(9):1226–1238, 2002. 2
- [31] 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 *CVPR*, 2015. 1, 7
- [32] P. Wang, X. Shen, B. Russell, S. Cohen, B. Price, and A. L. Yuille. Surge: Surface regularized geometry estimation from a single image. In *NIPS*, 2016. 1, 2, 6, 7
- [33] X. Wang, D. Fouhey, and A. Gupta. Designing deep networks for surface normal estimation. In *CVPR*, 2015. 1, 2, 5, 6, 7
- [34] D. Xu, E. Ricci, W. Ouyang, X. Wang, and N. Sebe. Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. *arXiv*, 2017. 1, 4, 5, 7
- [35] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv*, 2015. 4
- [36] B. Zeisl, M. Pollefeys, et al. Discriminatively trained dense surface normal estimation. In *ECCV*, pages 468–484. Springer, 2014. 7
- [37] W. Zhuo, M. Salzmann, X. He, and M. Liu. Indoor scene structure analysis for single image depth estimation. In *CVPR*, pages 614–622, 2015. 7