Deep Surface Normal Estimation on the 2-Sphere with Confidence Guided Semantic Attention

Quewei Li[1][0000−0003−3828−6756], Jie Guo[1], Yang Fei[1][0000−0001−7874−9175], Qinyu Tang[1][0000−0002−0130−7448], Wenxiu Sun[2], Jin Zeng[2], and Yanwen Guo[1]

[1] State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China
[2] SenseTime Research, China
{queweili, yangf, tangqinyu}@smail.nju.edu.cn, {guojie, ywguo}@nju.edu.cn, {irene.wenxiu.sun, jzeng2010}@gmail.com

Abstract. We propose a deep convolutional neural network (CNN) to estimate surface normal from a single color image accompanied with a low-quality depth channel. Unlike most previous works, we predict the normal on the 2-sphere rather than the 3D Euclidean space, which produces naturally normalized values and makes the training stable. Although the depth information is beneficial for normal estimation, the raw data contain missing values and noises. To alleviate this problem, we employ a confidence guided semantic attention (CGSA) module to progressively improve the quality of depth channel during training. The continuously refined depth features are fused with the normal features at multiple scales with the mutual feature fusion (MFF) modules to fully exploit the correlations between normals and depth, resulting in high quality normals and depth with fine details. Extensive experiments on multiple benchmark datasets prove the superiority of the proposed method.

Keywords: Normal estimation, 2-sphere, Confidence guided semantic attention, Mutual feature fusion

1 Introduction

This paper aims to recover high-quality surface normal and depth from a single RGB-D image using a deep neural network. Recently, the availability of depth information has promoted a great enhancement in the applications of object recognition [10, 31], sematic segmentation [4, 13, 20, 28], 3D scene reconstruction [17, 22, 25], pose estimation [5, 37], *etc.* From the depth channel, we can easily obtain surface normals with least square optimization [24, 29]. Unfortunately, depth images captured by low-cost depth sensors (*e.g.*, Microsoft Kinect and ASUS Xtion Pro) are notoriously corrupted by noises and contain missing/invalid values. These artifacts degrade the visual quality of both depth and normal images, influencing their usage in different tasks.

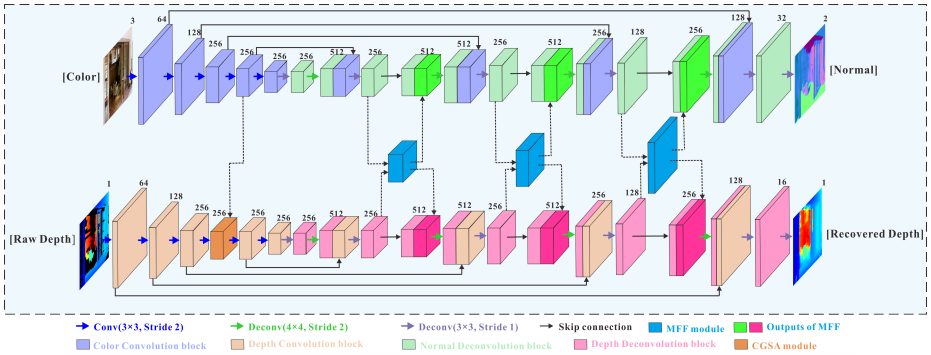


Fig. 1. The network architecture of our method

To improve the quality of normals and/or depth, a variety of methods have been proposed recently, among which the deep learning based solutions gain the most excellent results. These methods usually view the normal map as a conventional color image and predict the values directly in the 3D Euclidean space (*i.e.*, $\mathbb{R}^3$) [1, 29, 33, 38]. In this way, the three components of any output normal are learned without any restriction. However, we know that the values of normal should lie on a unit 3D ball (*i.e.*, S^2). A straightforward strategy is to explicitly normalize the result before calculating the loss [38]. However, such a strategy still cannot guarantee that the final output is normalized and is sub-optimal since the gradients propagated backwards to the model are unconstrained [21]. In contrast to this strategy, we opt to predict surface normals on the 2-sphere by learning two independent parameters: azimuth angle θ and elevation angle ϕ. Both parameters are constrained: $\theta \in [-\pi, \pi]$ and $\phi \in [0, \pi]$. The benefit of this solution is two-fold. First, it ensures that the output normals are naturally normalized without any explicit normalization operation. Second, it makes the training stable since the regression gradients are constrained and well-behaved.

In this paper, we use RGB and depth images as input in the task of normal estimation. These two inputs are fed into two separate branches of our neural network and fused at multiple scales with a new mutual feature fusion (MFF) module, as shown in Fig. 1. As a main difference from previous methods [38, 39], this module does not simply fuse features of different branches by concatenation but learns a mapping function from one feature map to the other. This fusion strategy can be viewed as using one branch as the guided filter to re-weight the feature maps in another branch via pixel-wise transformations, making a new form of hyper-network.

To handle missing values and noises in the raw depth data, Zeng *et al.* [38] suggested using a learned confidence map to mask out these invalid regions before feature fusion. However, we observe that these invalid regions actually contain many fine details such as corners and edges. Neglecting these features in the depth image will over-blur the estimated normal map. To tackle this problem, we introduce a confidence guided semantic attention (CGSA) module

for depth feature inpainting during network training, which enhances the depth image in high-level semantics. With the progressively improved depth channel, our network not only produces surface normals with clearer edges and more details, but also provides a high-quality depth map as a by-product.

To summarize, the main contributions of our work are:

- a deep learning based solution that estimates surface normal on the 2-sphere with a well defined loss function,
- a CGSA module designed to enhance depth feature by exploiting high-level semantics, and
- the multi-scale MFF modules to effectively combine features from different branches.

2 Related Work

Per-pixel normal estimation has been extensively studied in the past years. Traditional methods like [24] estimated surface normals via least square optimization. Qi *et al*. [29] integrated traditional methods into a deep neural network architecture and jointly predicted depth and surface normals with the 3D geometric information.

With the emergence of deep learning, recent methods predict surface normals under the framework of deep neural network, most of which use a single RGB image as the input [2, 8, 19, 40]. Wang *et al*. [33] proposed a network that integrates local, global, and vanishing point information to predict surface normals. Bansal *et al*. [1] proposed a skip connection network architecture to fuse features from different layers for normal estimation. Zhang *et al*. [41] designed a pattern-affinitive propagation network to predict surface normals and depth jointly by the affinity matrices. Due to the lack of geometric information in these RGB based methods, the details of the estimated results are not satisfactory, easily incurring over-blurriness or strange artifacts.

Compared with the RGB based methods, the study of RGB-D based surface normal estimation is far not enough. The 3D reconstruction based methods like [25] can be used for surface normal estimation but a sequence of RGB-D images are usually required for these methods. Zhang *et al*. briefly discussed normal estimation with RGB-D input in [39] and reported that their network produced better predictions with RGB input than the RGB-D input. More recently, Zeng *et al*. [38] proposed a hierarchical fusion network for surface normal estimation, which achieved the state-of-the-art performance. Notably, these works always treat invalid areas in the raw depth data as smooth plane that enables smooth transition in the corresponding areas of surface normals. However, as holes mainly exist along the boundaries of objects, the smooth transition will over-blur edges and erase details in the given scene.

To alleviate such a problem, we can first fill holes in the depth map in a preprocessing step with the guidance of the RGB image and further improve the performance of the estimated normals. Unfortunately, the accuracy of enhanced depth is still limited as single image depth enhancement itself is also a

challenging problem [11, 14, 15, 23, 24, 26, 39]. Considering the strong correlations between depth and surface normals, we design a unified network to predict surface normals and enhance low-quality depth jointly. To fill large missing areas in the raw depth, we perform depth inpainting for high-level depth feature map by updating unreliable regions with patches in the reliable regions, which lessens the influence of big holes to a large extent and makes our method more stable.

3 Surface Normal Estimation on the 2-Sphere

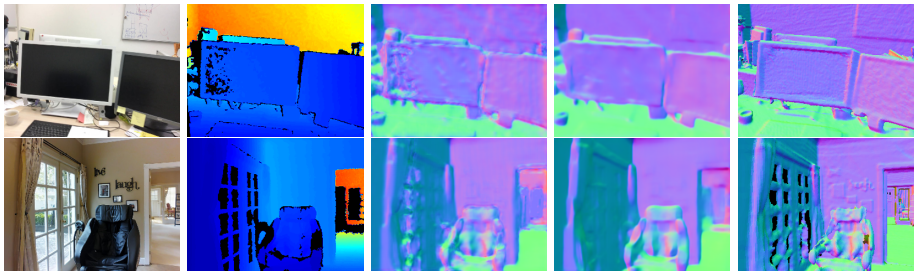
The following sections detail our network to estimate per-pixel surface normal from a single RGB-D image. As illustrated in Fig. 1, the basic architecture of our network contains two autoencoders with skip connections (Sec. 3.1). One autoencoder maps a color image to a normal map with multi-scale features fused from the corresponding depth branch using the MFF modules (Sec. 3.2). The output normals are represented in the polar coordinate (Sec. 3.3) with a well defined loss function (Sec. 3.4), ensuring proper normalization and stable training. Another autoencoder progressively refines the raw depth image leveraging the color features and a CGSA module specifically designed for depth feature map inpainting (Sec. 4).

3.1 Network Architecture

Our network comprises two autoencoders, both of which are similar in their architectures. In the color branch of the first autoencoder, we adopt a modified VGG-16 network [32] by reducing channel numbers of the last two convolution blocks, *i.e.*, conv4 and conv5 in the original VGG-16, from 512 to 256. The raw depth branch of the second autoencoder is organized in a similar way to the color branch except that a CGSA module is inserted before the fourth block, *i.e.*, conv4, of the encoder as shown in Fig. 1 (dark orange box). Two decoders are symmetric to the encoders and are equipped with skip connections. Three MFF modules are used to fuse feature maps at different scales of the decoders. The last deconvolution layer with stride 1 and kernel size 3×3 outputs a 2-channel image (representing θ and ϕ of surface normal) and a single channel image (representing depth) in each decoder, respectively. We use ReLU activation function for all the (de)convolution layers except the last layer, which is equipped with the sigmoid activation function, to ensure that the output values fall in the range of [0,1].

3.2 Multi-scale Mutual Feature Fusion

To fully utilize the close geometry relationship between depth and surface normals, some previous works resort to fuse information from different feature maps. The most widely used strategy, as adopted in [38, 39], is by a simple concatenation operation. We observe that this is sub-optimal since the feature maps are from different domains such that they can not be properly handled with the same convolution operations. Zhang *et al.* [39] validated that such a strategy



(a)Color (b)Raw depth (c)Concatenation (d)MFF (e)GT

Fig. 2. Visual quality comparisons between different feature fusion strategies, *i.e.*, the simple concatenation operation and the mutual feature fusion (MFF) module

will make the network learn from the inaccurate and incomplete depth directly, lowering the influence of the color information. Consequently, as demonstrated in Fig. 2(c), the invalid or unreliable areas of the depth map will mislead the prediction of surface normals, leading to strange artifacts around the boundaries of depth holes.

Instead of simply concatenating features from different branches, we design a multi-scale fusion strategy in which the two autoencoders exchange features at multiple scales by several MFF modules. As illustrated in Fig. 3, each MFF module contains four conditional feature transform (CFT) blocks [3] (light blue boxes) and four convolution layers (dark blue arrays). The motivation of using the CFT blocks is to view the normal estimation as a conditional generative problem in which the depth image serves as an auxiliary feature. Similarly, the normal map is considered as an auxiliary feature in the task of depth inpainting. Such a strategy has been previously used in image dehazing [12], image translation [16] and Monte Carlo denoising [36]. All these methods including ours rely on some auxiliary features as a condition to address the ill-posed image generation problems.

Supposing $\mathcal{F}_a$ is an auxiliary image (or feature map) and $\mathcal{F}_s$ is the source image, the output of the CFT block at scale l is the target image $\mathcal{F}_t$ defined as

$$\mathcal{F}_t^l = \mathrm{CFT}(\mathcal{F}_s^l, \mathcal{F}_a^l | \boldsymbol{\gamma}, \boldsymbol{\beta}) = \boldsymbol{\gamma}(\mathcal{F}_a^l) \otimes \mathcal{F}_s^l \oplus \boldsymbol{\beta}(\mathcal{F}_a^l) \tag{1}$$

where $(\boldsymbol{\gamma}, \boldsymbol{\beta})$ is the modulation parameter pair with $\boldsymbol{\gamma}$ denoting the scaling operation matrix and $\boldsymbol{\beta}$ denoting the shifting operation matrix. $\otimes$ and $\oplus$ represent the element-wise multiplication and addition operation, respectively. Intuitively, the CFT block learns a mapping function that outputs the $(\boldsymbol{\gamma}, \boldsymbol{\beta})$ pair under some auxiliary feature conditions. Applying this mapping function to any source image yields a new target image.

In our MFF module, we utilize such conditioned mapping functions (*i.e.*, CFT blocks) mutually to generate parameters for pixel-wise transformation and

[3] We use four CFT blocks to improve the MFF module's ability of representing more complex feature transformations.

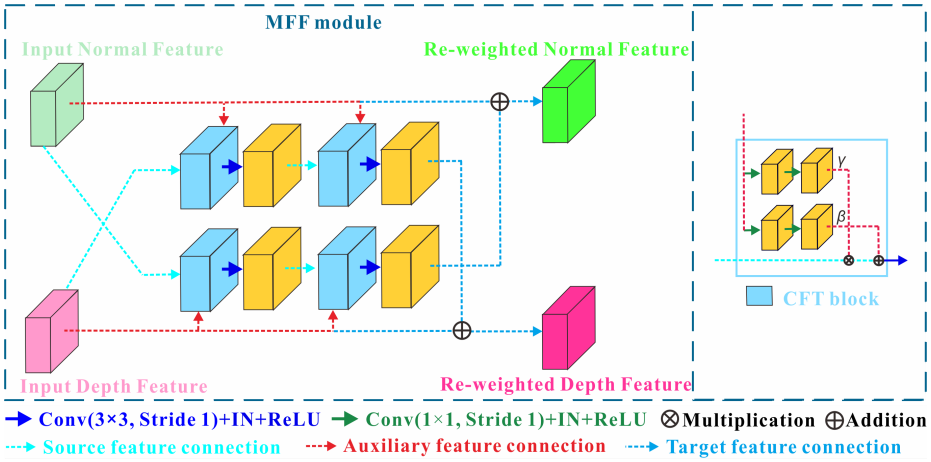


Fig. 3. The MFF module. IN represents the InstanceNorm layer and ReLU represents the ReLU activation function

modify the weights of each feature map in different branches at multiple scales. Specifically, the normal feature map acts as $\mathcal{F}_s$ and the depth feature map as $\mathcal{F}_a$ in the normal branch of the decoder while the roles exchange in the depth branch. Rather than simple concatenation, we re-weight the feature maps through additive and multiplicative interactions based on the conditioning representation. To better utilize feature information from high-level to low-level, we embed the MFF module at three scales in the decoder. As evidenced in Fig. 2(d), the MFF modules avoid the risk of incurring strange artifacts due to unreliable depth values. Note that masking out the invalid areas in the depth map, as suggested by Zeng *et al.* [38], will also remove these artifacts but lead to loss of details as compared in Fig. 6.

3.3 2-Sphere vs. 3D Euclidean Space

One of our important insights is that the 2-sphere space is more suitable for estimating surface normals than the 3D Euclidean space. Estimating surface normals in the 3D Euclidean space has several problems. First, the output 3-channel normals are learned without any restriction. Zeng *et al.* [38] performed a normalization before calculating the loss. However, this does not always guarantee that the outputs form unit vectors that indicate directionality. Moreover, such a normalized operation is sub-optimal since the gradients propagated backwards to the model are not constrained [21], potentially leading to unstable training or convergence. To tackle such a problem, Liao *et al.* [21] suggested modifying the traditional normalization with a spherical exponential function to enable stable training. However, since this function always outputs positive values, an additional classification branch is required to predict the sign values, which complicates the prediction.

Instead, our method deals with the problem in a more straightforward and efficient way by predicting two independent angles directly on the 2-sphere: azimuth angle θ and elevation angle ϕ. The two spaces are linked by the following formulas:

$$\begin{cases} \theta = \text{arctan2 } (x, y) \\ \phi = \text{arctan2 } (z, \sqrt{x^2 + y^2}) \end{cases} \tag{2}$$

where arctan2 is the two-dimension form of the arctan function. (x, y, z) is the Cartesian coordinate of the normal.

The benefit of predicting surface normals on the 2-sphere is at least two-fold. First of all, the 2-sphere is a naturally closed geometric manifold defined in the $\mathbb{R}^3$ that the output normal is expected to be normalized when getting back to the 3D Euclidean space. Second, without the need of any explicit normalization operation, the gradients escape from passing through the normalization layer for backward propagation, which makes the gradients constrained and enables stable training and easy convergence since the final layer is activated by the sigmoid function whose gradient is only determined by the constrained output. To train our model on the 2-sphere, we convert the ground-truth normals to the polar coordinate with Eq. 2. After prediction, we convert the learned θ and ϕ back to the 3D Euclidean space.

3.4 Loss Function

The loss function for our network is the L_1 norm which reflects the median angle difference between the predicted result and the ground truth. Denoting the input RGB and depth image by I_c and I_d, the output normal and depth image by T_n and T_d and the ground-truth/target normal and depth image by G_n and G_d, we aim at minimizing the distance between the ground truth and the output generated from the input RGB-D image. The loss function contains two components: one for surface normal and the other for depth. Though we represent the azimuth angle θ in the range of $[-\pi, \pi]$ (or in the range of $[0, 1]$ after normalization), $-\pi$ and π actually indicate the same direction on S^2. Considering this property, we define the "circle loss" operator as $\ominus$ for θ where

$$\|T_n^\theta \ominus G_n^\theta\|_1 = 2 \min \left(\|T_n^\theta - G_n^\theta\|_1, 1 - \|T_n^\theta - G_n^\theta\|_1 \right). \tag{3}$$

Then, our loss function is defined specifically as

$$\begin{aligned} \mathcal{L}(T_n, T_d, G_n, G_d | I_c, I_d) = \frac{1}{N} \sum_{i=1}^{N} (& \|T_{n,i}^\phi - G_{n,i}^\phi\|_1 \\ & + \|T_{n,i}^\theta \ominus G_{n,i}^\theta\|_1 + \lambda \|T_{d,i} - G_{d,i}\|_1) \end{aligned} \tag{4}$$

where N is the total pixel number, $\|\cdot\|_1$ denotes the L_1 norm and λ is a balanced factor between depth and surface normal.

(a) (b) (c)

Fig. 4. Visualization of the raw depth images (the left image in each group) and the corresponding depth confidence maps (the right image in each group)

4 Confidence Guided Semantic Attention

In this section, we introduce the CGSA module for depth feature inpainting. Although the depth information is becoming easier to obtain with the development of the RGB-D sensors, it is not fully exploited in the problem of surface normal estimation. The most important factor is that the depth data is not always reliable. Recently, Zeng *et al*. [38] proposed a confidence guided RGB-D fusion scheme to make use of limited geometric information from raw depth data for surface normal estimation. Their confidence map, which is also learned from a neural network, acts as a mask that masks out depth features with the low confidence before passing to the RGB branch. However, low confidence areas are mostly important scene details such as edges. Simply eliminating these areas will lead to the loss of details in the final results. Considering this, we propose a CGSA module to utilize spatial attention to recover missing values of the depth map conditioned on the valid patches from the reliable regions.

4.1 Confidence Map for Raw Depth

Before discussing the details of the CGSA module, we first introduce our definition of depth confidence map which will be used in the CGSA module. We assume that if there is no missing value in the neighboring region of a pixel i in the input depth image I_d and the variance of its local region is small, the depth confidence of i is high. Therefore, $C(I_d, i)$ is given by

$$C(I_d, i) = \begin{cases} 1, & \text{if } \widetilde{C}(I_d, i) > T_s \\ \widetilde{C}(I_d, i), & \text{otherwise} \end{cases} \tag{5}$$

$$\widetilde{C}(I_d, i) = \mathbb{I}(I_{d,i} > 0) \Gamma(I_{d,i}) \cdot \exp(-\frac{\sigma_i^2}{\gamma_d}) \tag{6}$$

with $\mathbb{I}(I_{d,i} > 0)$ being an indication function. γ_d is the controlling factor and T_s is a pre-defined threshold. $\Gamma(I_{d,i})$ represents the percentage of valid pixels in a neighboring region of i with the form

$$\Gamma(I_{d,i}) = \frac{\sum_{j \in \mathcal{N}(i)} \mathbb{I}(I_{d,j} > 0)}{|\mathcal{N}(i)|} \tag{7}$$

where $|\mathcal{N}(i)|$ returns the total number of the neighboring pixels. $\sigma_i^2 = \sum_{j \in \mathcal{N}(i)} \|I_{d,i} - I_{d,j}\|^2$ represents the variance among the neighboring region $\mathcal{N}(i)$ of i. We use

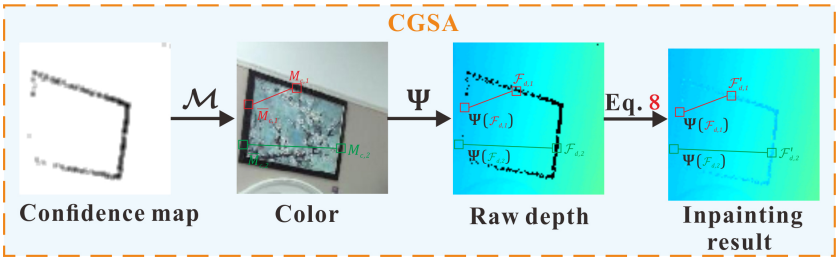


Fig. 5. The process of CGSA module. The confidence values of the confidence map in the white regions are 1 and the dark regions are lower than 1

neighborhood of a 3×3 size of pixel i for all the scenes. Several examples of depth confidence maps are shown in Fig. 4. As seen, pixels with low confidence values always exist in large holes and along object boundaries. We down-sample the confidence map to match the spatial size of feature map in the CGSA module.

4.2 The CGSA Module

Considering the high correlation between RGB and depth, most existing methods [15, 24, 27, 35] use RGB images to recover the missing depth values under the assumption that pixels with similar colors tend to have similar depth in a local region. Similarly, we perform depth feature inpainting to enhance the depth channel with the CGSA module. The workflow of CGSA is shown in Fig. 5, where the key point is to update each low confidence depth feature patch with the most similar high confidence depth feature patch.

More specifically, we first define the attention map $\mathcal{M}$ based on the calculated confidence map that $\mathcal{M}_i$ is set to 0 if $\hat{C}(i) = 1$ and set to 1 in other cases. Here, $\hat{C}$ denotes the down-sampled confidence map from C that $\hat{C}(i)$ returns the confidence value of patch i. We employ $\mathcal{M}$ on the color feature map $\mathcal{F}_c$ so that $\mathcal{F}_c$ is divided into the attention regions M_c (regions with low confidence) and the reference regions $\bar{M}_c$ (regions with confidence equal to 1). For each attention patch $M_{c,i}$, we find the closest-matching patch $\bar{M}_{c,i}$ in the reference regions. The relevant degree between these patches is measured with the squared L_2 distance defined as $\|\bar{M}_{c,i} - M_{c,i}\|_2^2$. By doing this, we actually find a mapping $\mathbf{\Psi}$ for every patch from the attention regions to the reference regions. We then apply $\mathbf{\Psi}$ to the depth feature map $\mathcal{F}_d$ so that each patch $\mathcal{F}_{d,i}$ in the low confidence regions couples with the most similar reference patch $\mathbf{\Psi}(\mathcal{F}_{d,i})$. Finally, we update $\mathcal{F}_{d,i}$ with the following scheme:

$$\mathcal{F}'_{d,i} = \hat{C}(i) \cdot \mathcal{F}_{d,i} + (1 - \hat{C}(i)) \cdot \mathbf{\Psi}(\mathcal{F}_{d,i}). \tag{8}$$

Considering that enhancing depth locally fails to fill large holes, we perform depth feature inpainting on high-level semantics, which avoids filling big holes directly at large scales and makes the module work more stable, efficiently improving the qualities of both the estimated surface normals and recovered depth.

The detailed implementation of the CGSA module is provided in the supplemental material.

5 Experiments

5.1 Implementation Details

Datasets. We evaluate our method on three RGB-D datasets: NYUD-v2 [24], ScanNet [6] and Matterport3D [3]. NYUD-v2 dataset consists of RGB-D images collected from 464 different indoor scenes, among which 1449 images are provided with ground-truth normals and depth. We randomly choose 1200 of them for training and the remaining 249 images for testing. For the ScanNet and Matterport3D datasets, we use the ground-truth data provided by Zhang *et al*. [39] and follow their training and testing lists. Specifically, we use 105432 images for training and 12084 for testing of Matterport3D; 59743 for training and 7517 for testing of ScanNet. We convert all the normal maps in the training datasets from $\mathbb{R}^3$ to S^2 with Eq. 2 before training. After that, the training process carries on S^2. We train our network and test its performance on the three datasets, respectively. All the methods in comparison are tested with the same testing lists.

Training Details. Our network generally converges after 60 epochs. We implement it with PyTorch on four NVIDIA GTX 2080Ti GPUs. We use RMSprop optimizer and adjust the learning rate with the initial rate of $1e^{-3}$ and the power of 0.95 every 10 epochs. The hyper-parameters $\{\lambda, \gamma_d, T_s\}$ are set to $\{2.0, 0.2, 0.98\}$ according to validation on a 5% randomly split training data.

Evaluation Metrics. We adopt four metrics to evaluate the qualities of estimated normals: the mean of angle error (mean), the median of angle error (median), the root mean square error (rmse) and the pixel accuracy with angle difference with ground truth less than t_n where $t_n \in \{11.25°, 22.5°, 30°\}$ [29, 38, 40]. The qualities of the recovered depth are also evaluated with four metrics: the root mean square error (rmse), the mean relative error (rel), the mean log 10 error (log 10), and the pixel accuracy with $\max(\frac{T_d}{G_d}, \frac{G_d}{T_d})$ less than t_d where $t_d \in \{1.25, 1.25^2, 1.25^3\}$ [29, 41].

5.2 Comparisons with the State-of-the-arts

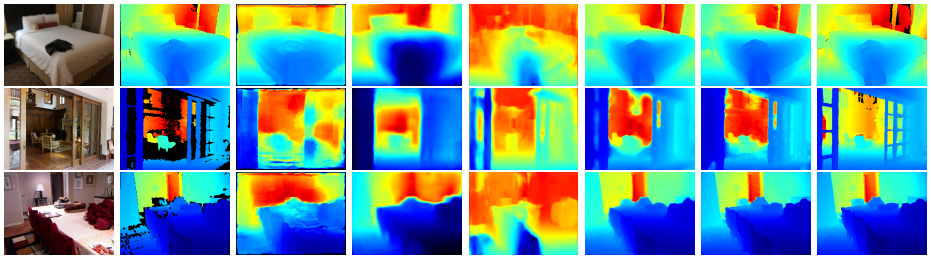
In this section, we compare our network with the state-of-the-art normal and depth estimation methods.

Comparisons of Surface Normal Estimation. We compare the results of our method with some high-ranking surface normal estimation methods, including Zhang's network [40], GeoNet [29], SharpNet [30] and HFM-Net [38] with their public available pre-trained models fine-tuned on each dataset. Fig. 6 shows the visual quality comparisons among all these methods. As seen, the results produced by Zhang's network and SharpNet have many unwanted details, *e.g.*, the highlights on the desk, due to the lack of depth information. GeoNet alleviates

(a)Color (b)Raw depth (c)Zhang's (d)GeoNet (e)SharpNet (f)HFM-Net (g)Ours (h)GT

Fig. 6. Visual quality comparisons with the state-of-the-art surface normal estimation methods on ScanNet (the first and second rows), Matterport3D (the third and fourth rows) and NYUD-v2 (the last row) datasets

this problem by jointly learning depth and surface normals, incorporating geometric relation between them. However, it tends to generate over-blurred results. HFM-Net generally performs better than these previous methods, but still has the problem of detail losing, especially in the areas that have been masked out by the depth confidence map, *e.g.*, the edges of the desks in the first two scenes, the leaf stalk in the third scene and the lamp in the last scene. In comparison, our method achieves better visual effects that are very close to the ground truths and preserves most of scene details without introducing artifacts.



(a)Color (b)Raw depth (c)GeoNet (d)SharpNet (e)MonoD (f)LabDEN (g)Ours (h)GT

Fig. 7. Visual quality comparisons with the state-of-the-art depth estimation methods on ScanNet (the first row), Matterport3D (the second row) and NYUD-v2 (the third row) datasets

To further validate the accuracy of our method, we provide quantitative analysis for different datasets in Table 1. The best results are highlighted in

Table 1. Performance of surface normal estimation on the NYUD-v2, Matterport3D and ScanNet datasets. The last three columns are the different variants of the proposed method, *i.e.*, the model trained on $\mathbb{R}^3$, the model without CGSA (-CGSA) and the model without MFF (-MFF)

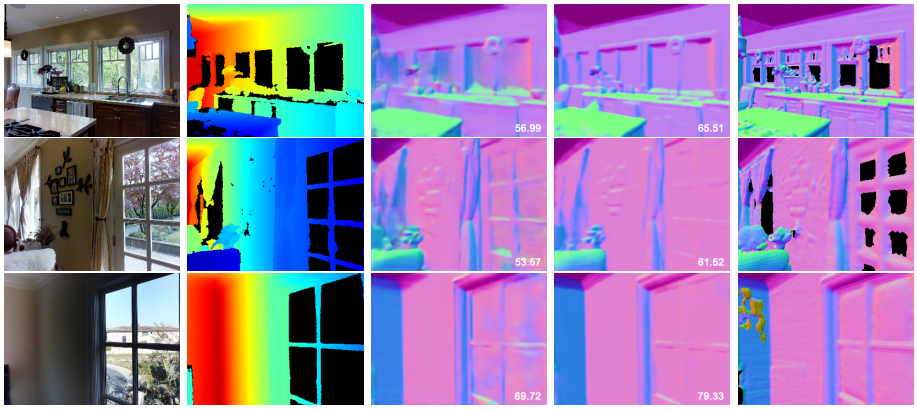
	Metrics	Zhang's	GeoNet	SharpNet	HFM-Net	Ours	$\mathbb{R}^3$	-CGSA	-MFF
NYUD-v2	mean	23.430	21.385	21.226	14.188	**12.172**	12.850	12.790	13.001
	median	14.446	12.810	14.084	6.827	**6.377**	7.222	7.595	7.193
	rmse	30.162	30.257	28.912	23.139	**19.152**	20.027	19.317	20.335
	11.25°	39.95	44.93	41.39	65.91	**69.41**	66.39	65.66	65.31
	22.5°	66.11	68.16	67.24	82.03	**85.90**	84.89	84.43	84.35
	30°	75.35	76.27	76.43	87.36	**90.40**	89.96	89.90	89.51
Matter-port3D	mean	21.920	24.277	25.599	17.140	**14.687**	15.903	16.147	15.768
	median	11.039	15.975	18.319	6.483	**4.885**	5.759	6.336	6.010
	rmse	32.041	34.454	34.806	27.339	**25.308**	26.476	26.654	26.179
	11.25°	48.25	40.17	27.56	61.05	**69.74**	66.65	65.79	65.95
	22.5°	67.13	62.42	62.60	77.51	**82.55**	80.44	80.31	80.71
	30°	75.00	71.65	73.27	83.14	**86.73**	85.13	85.08	85.44
Scan-Net	mean	23.306	23.289	23.977	14.590	**13.508**	14.205	14.390	14.425
	median	15.950	15.725	17.038	7.468	**6.739**	6.938	7.019	6.971
	rmse	31.371	29.902	31.974	23.638	**21.991**	23.024	22.933	23.093
	11.25°	40.43	46.41	27.95	65.65	**67.21**	65.70	65.18	65.24
	22.5°	63.08	64.04	63.91	81.21	**82.85**	81.72	81.27	81.21
	30°	71.88	76.78	75.74	86.21	**87.68**	86.75	86.38	86.29

Table 2. Performance of depth estimation on NYUD-v2 dataset

	Metrics	GeoNet	SharpNet	MonoD	LabDEN	Ours
NYUD-v2	rmse	0.106	0.104	0.197	0.028	**0.015**
	log10	0.121	0.150	0.198	0.030	**0.019**
	rel	0.283	0.278	0.610	0.069	**0.044**
	1.25	56.50	50.38	34.87	95.26	**97.80**
	1.25^2	84.98	75.16	58.20	98.74	**99.69**
	1.25^3	95.08	86.37	75.51	99.57	**99.96**

bold. As seen, in all the cases, our method ranks first among these peer-reviewed methods according to the metrics mentioned above. It is worth noticing that our method achieves a significant improvement on the metrics of angle difference, especially in the case of $t_n = 11.25°$. This further proves the benefit of normal estimation on the 2-sphere since the 2-sphere is friendly to angle differences.

Comparisons of Depth Estimation. As a by-product, our method also produces high-quality depth maps. We conduct comparisons with some popular depth estimation methods to verify the effectiveness and robustness of our proposed method. The methods in comparison are GeoNet [29], SharpNet [30], MonoD [9] and LabDEN [18]. Both visual comparisons in Fig. 7 and quantitative comparisons in Table 2 reveal that our method outperforms these previous methods, achieving the state-of-the-art performance in depth recovery.



(a) Color (b) Raw depth (c) $\mathbb{R}^3$ (d) S^2 (e) GT

Fig. 8. Visual quality comparions between the $\mathbb{R}^3$ space and the S^2 space

5.3 Ablation Study

To validate the effectiveness of each module in our method, we conduct several ablation studies.

Comparisons between Different Spaces. We compare surface normal estimation on the 3D Euclidean space (*i.e.*, $\mathbb{R}^3$) and the 2-sphere space (*i.e.*, S^2) in Fig. 8. As expected, normal estimation on S^2 provides higher-quality results than directly regressing the Cartesian coordinate in the $\mathbb{R}^3$ space. It achieves an obvious improvement in accuracy in terms of angle differences. As reported in the bottom right corner of each image, our method on S^2 significantly surpasses that in $\mathbb{R}^3$ in the metric of the angle difference of $t_n = 11.25°$.

Effectiveness of the MFF Module. In Fig. 2 we show that the proposed MFF module is better than simple concatenation on fusing information from different branches. For simple concatenation, artifacts occur in the areas where the depth information is unreliable. It considers the inaccurate regions in the depth map as extra features and produces strange artifacts in these areas. The MFF module avoids this problem by not directly using the depth feature maps but re-weighting the normal feature maps via pixel-wise transformation based on conditioning representation, producing more stable and pleasing results.

Effectiveness of the CGSA Module. We verify the effectiveness of the CGSA module by removing it from our complete method. As shown in Fig. 9, without the depth inpainting procedure, wrong predictions appear in the large holes of the depth map. With the CGSA module enhancing depth feature map in high-level semantics, our complete method produces more plausible results in the regions where the raw depth is absent.

However, the enhanced depth values are not always reliable when our inpainting fails to find similar patches. As shown on the LCD screen of the first scene and the windows of the second scene in Fig. 2(c), these inaccurate depth values may affect the estimated normal values, leading to strange artifacts. The pro-

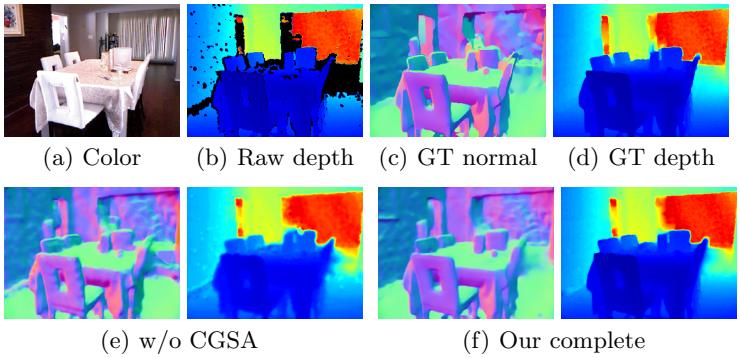


Fig. 9. Visual quality comparisons between with and w/o CGSA module

posed MFF can avoid this as explained before. Nevertheless, CGSA is necessary to avoid wrong predictions in large holes of raw depth maps.

We also conduct quantitative analysis of different situations of our method on the above three datasets. The quantitative results in the last four columns of Table 1 show that our complete model consistently shows superior performance compared with other models.

6 Conclusion and Future Work

In this work, we prove the superiority of estimating surface normals on the naturally normalized 2-sphere than in the unconstrained $\mathbb{R}^3$ space. To improve the feature quality of the depth channel, we design a CGSA module to recover depth feature maps in high-level semantics. Our network fuses RGB-D features at multiple scales with the MFF modules, which organizes the two branches as a new form of hyper-network. Moreover, we design a loss function which is more suitable for the 2-sphere. Extensive experimental results verify that our method outperforms the state-of-the-art methods in providing high-quality surface normals with clearer edges and more details.

Although our method achieves the state-of-the-art performance, it suffers from some limitations. Notably, our network fails to capture some sharp details especially for distant objects, *e.g.*, the details on the wall. This is probably due to the low-quality of ground-truth normal maps in current datasets. We hope this would be solved by introducing high-quality datasets or by developing GAN-based generative models [7, 34] to recover these sharp features.

Acknowledgement

The corresponding authors of this work are Jie Guo and Yanwen Guo. This research was supported by the National Natural Science Foundation of China under Grants 61772257 and the Fundamental Research Funds for the Central Universities 020914380080.

References

1. Bansal, A., Russell, B., Gupta, A.: Marr Revisited: 2D-3D model alignment via surface normal prediction. In: CVPR (2016)
2. Bo Li, Chunhua Shen, Yuchao Dai, van den Hengel, A., Mingyi He: Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1119–1127 (June 2015). https://doi.org/10.1109/CVPR.2015.7298715
3. Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3D: Learning from RGB-D data in indoor environments. International Conference on 3D Vision (3DV) (2017)
4. Cheng, Y., Cai, R., Li, Z., Zhao, X., Huang, K.: Locality-sensitive deconvolution networks with gated fusion for rgb-d indoor semantic segmentation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1475–1483 (July 2017). https://doi.org/10.1109/CVPR.2017.161
5. Christian Zimmermann, Tim Welschehold, C.D.W.B., Brox, T.: 3d human pose estimation in rgbd images for robotic task learning. In: IEEE International Conference on Robotics and Automation (ICRA) (2018), https://lmb.informatik.uni-freiburg.de/projects/rgbd-pose3d/
6. Dai, A., Nießner, M., Zollöfer, M., Izadi, S., Theobalt, C.: Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface re-integration. ACM Transactions on Graphics 2017 (TOG) (2017)
7. Denton, E.L., Chintala, S., szlam, a., Fergus, R.: Deep generative image models using a laplacian pyramid of adversarial networks. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems 28, pp. 1486–1494. Curran Associates, Inc. (2015), http://papers.nips.cc/paper/5773-deep-generative-image-models-using-a-laplacian-pyramid-of-adversarial-networks.pdf
8. Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: 2015 IEEE International Conference on Computer Vision (ICCV). pp. 2650–2658 (Dec 2015). https://doi.org/10.1109/ICCV.2015.304
9. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. In: CVPR (2017)
10. Gupta, S., Girshick, R., Arbeláez, P., Malik, J.: Learning rich features from rgb-d images for object detection and segmentation. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision – ECCV 2014. pp. 345–360. Springer International Publishing, Cham (2014)
11. Haefner, B., Quéau, Y., Möllenhoff, T., Cremers, D.: Fight ill-posedness with ill-posedness: Single-shot variational depth super-resolution from shading. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 164–174 (2018)
12. He, K., Sun, J., Tang, X.: Single image haze removal using dark channel prior. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(12), 2341–2353 (Dec 2011). https://doi.org/10.1109/TPAMI.2010.168
13. He, Y., Chiu, W., Keuper, M., Fritz, M.: Std2p: Rgbd semantic segmentation using spatio-temporal data-driven pooling. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7158–7167 (July 2017). https://doi.org/10.1109/CVPR.2017.757

14. Herrera C., D., Kannala, J., Ladický, L., Heikkilä, J.: Depth map inpainting under a second-order smoothness prior. In: Kämäräinen, J.K., Koskela, M. (eds.) Image Analysis. pp. 555–566. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
15. Hui, T.W., Loy, C.C., Tang, X.: Depth map super-resolution by deep multi-scale guidance. In: Proceedings of European Conference on Computer Vision (ECCV) (2016)
16. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)
17. Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., Fitzgibbon, A.: Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera. In: UIST '11 Proceedings of the 24th annual ACM symposium on User interface software and technology. pp. 559–568. ACM (October 2011), https://www.microsoft.com/en-us/research/publication/kinectfusion-real-time-3d-reconstruction-and-interaction-using-a-moving-depth-camera/
18. Jeon, J., Lee, S.: Reconstruction-based pairwise depth dataset for depth image enhancement using cnn. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision – ECCV 2018. pp. 438–454. Springer International Publishing, Cham (2018)
19. Ladický, L., Zeisl, B., Pollefeys, M.: Discriminatively trained dense surface normal estimation. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision – ECCV 2014. pp. 468–484. Springer International Publishing, Cham (2014)
20. Lee, S., Park, S.J., Hong, K.S.: Rdfnet: Rgb-d multi-level residual feature fusion for indoor semantic segmentation. 2017 IEEE International Conference on Computer Vision (ICCV) pp. 4990–4999 (2017)
21. Liao, S., Gavves, E., Snoek, C.G.M.: Spherical regression: Learning viewpoints, surface normals and 3d rotations on n-spheres. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
22. Litany, O., Bronstein, A.M., Bronstein, M.M., Makadia, A.: Deformable shape completion with graph convolutional autoencoders. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition pp. 1886–1895 (2017)
23. Liu, J., Gong, X., Liu, J.: Guided inpainting and filtering for kinect depth maps. In: Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012). pp. 2055–2058. IEEE (2012)
24. Nathan Silberman, Derek Hoiem, P.K., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: ECCV (2012)
25. Newcombe, R.A., Fox, D., Seitz, S.M.: Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 343–352 (June 2015). https://doi.org/10.1109/CVPR.2015.7298631
26. Or-El, R., Rosman, G., Wetzler, A., Kimmel, R., Bruckstein, A.M.: Rgbd-fusion: Real-time high precision depth recovery. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5407–5416 (2015)
27. Park, J., Kim, H., Yu-Wing Tai, Brown, M.S., Kweon, I.: High quality depth map upsampling for 3d-tof cameras. In: 2011 International Conference on Computer Vision. pp. 1623–1630 (Nov 2011). https://doi.org/10.1109/ICCV.2011.6126423
28. Qi, X., Liao, R., Jia, J., Fidler, S., Urtasun, R.: 3d graph neural networks for rgbd semantic segmentation. In: ICCV (2017)

29. Qi, X., Liao, R., Liu, Z., Urtasun, R., Jia, J.: Geonet: Geometric neural network for joint depth and surface normal estimation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
30. Ramamonjisoa, M., Lepetit, V.: Sharpnet: Fast and accurate recovery of occluding contours in monocular depth estimation. The IEEE International Conference on Computer Vision (ICCV) Workshops (2019)
31. Ruizhongtai Qi, C., Liu, W., Wu, C., Su, H., Guibas, L.: Frustum pointnets for 3d object detection from rgb-d data. pp. 918–927 (06 2018). https://doi.org/10.1109/CVPR.2018.00102
32. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
33. Wang, X., Fouhey, D.F., Gupta, A.: Designing deep networks for surface normal estimation. In: CVPR (2015)
34. Wu, J., Zhang, C., Xue, T., Freeman, B., Tenenbaum, J.: Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In: Advances in neural information processing systems. pp. 82–90 (2016)
35. Xie, J., Feris, R.S., Sun, M.: Edge-guided single depth image super resolution. IEEE Transactions on Image Processing **25**(1), 428–438 (Jan 2016). https://doi.org/10.1109/TIP.2015.2501749
36. Xu, B., Zhang, J., Wang, R., Xu, K., Yang, Y.L., Li, C., Tang, R.: Adversarial monte carlo denoising with conditioned auxiliary feature. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH Asia 2019) **38**(6), 224:1–224:12 (2019)
37. Yang, Z., Pan, J.Z., Luo, L., Zhou, X., Grauman, K., Huang, Q.: Extreme relative pose estimation for rgb-d scans via scene completion. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
38. Zeng, J., Tong, Y., Huang, Y., Yan, Q., Sun, W., Chen, J., Wang, Y.: Deep surface normal estimation with hierarchical rgb-d fusion. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
39. Zhang, Y., Funkhouser, T.: Deep depth completion of a single rgb-d image. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
40. Zhang, Y., Song, S., Yumer, E., Savva, M., Lee, J.Y., Jin, H., Funkhouser, T.: Physically-based rendering for indoor scene understanding using convolutional neural networks. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
41. Zhang, Z., Cui, Z., Xu, C., Yan, Y., Sebe, N., Yang, J.: Pattern-affinitive propagation across depth, surface normal and semantic segmentation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)