Estimating and Exploiting the Aleatoric Uncertainty in Surface Normal Estimation

Gwangbin Bae Ignas Budvytis Roberto Cipolla
University of Cambridge
{gb585,ib255,rc10001}@cam.ac.uk

Abstract

Surface normal estimation from a single image is an important task in 3D scene understanding. In this paper, we address two limitations shared by the existing methods: the inability to estimate the aleatoric uncertainty and lack of detail in the prediction. The proposed network estimates the per-pixel surface normal probability distribution. We introduce a new parameterization for the distribution, such that its negative log-likelihood is the angular loss with learned attenuation. The expected value of the angular error is then used as a measure of the aleatoric uncertainty. We also present a novel decoder framework where pixel-wise multi-layer perceptrons are trained on a subset of pixels sampled based on the estimated uncertainty. The proposed uncertainty-guided sampling prevents the bias in training towards large planar surfaces and improves the quality of prediction, especially near object boundaries and on small structures. Experimental results show that the proposed method outperforms the state-of-the-art in ScanNet [4] and NYUv2 [33], and that the estimated uncertainty correlates well with the prediction error. Code is available at https://github.com/baegwangbin/surface_normal_uncertainty.

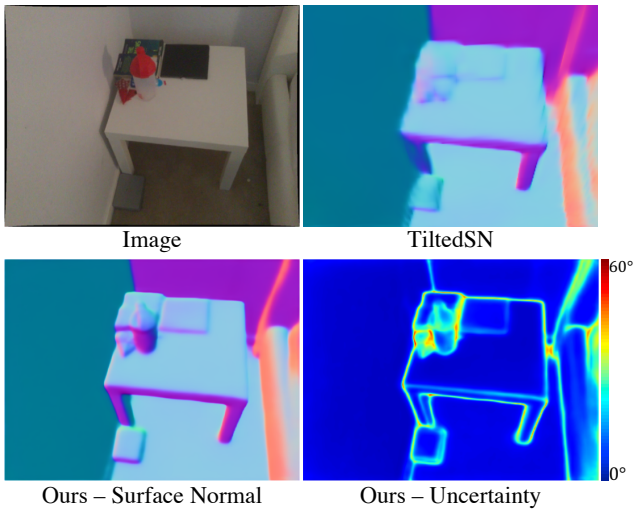

Figure 1. Comparison between our method and TiltedSN [6]. The proposed network estimates the *surface normal probability distribution*, from which the expected angular error can be inferred. The prediction made by our method shows clearer object boundaries and preserves a higher level of detail. This is due to the proposed uncertainty-guided sampling which prevents the bias in training towards large planar surfaces.

1. Introduction

The ability to estimate surface normal from a single RGB image plays a crucial role in understanding the 3D scene geometry. The estimated normal can be used to build augmented reality (AR) applications [18] or to control autonomous robots [41]. In this work, we address two limitations shared by the state-of-the-art methods.

(1) Inability to estimate the aleatoric uncertainty. State-of-the-art learning-based approaches [39, 7, 1, 31, 14, 18, 42, 24, 32, 6, 38] train deep networks by minimizing some distance metric (e.g., L_2) between the predicted normal and the ground truth. However, the ground truth normal, calculated from a measured depth map, can be sensitive to the depth noise and to the algorithm used to compute the normal (see Fig. 2 for examples of inaccurate ground truth). The network should be able to capture such aleatoric uncertainty, in order to be deployed in real-world applications.

(2) Lack of detail in the prediction. An indoor scene generally consists of large planar surfaces (e.g., walls and floors) and small objects with complex geometry. Therefore, if the training loss is applied to all pixels, the learning becomes biased to large surfaces, resulting in an over-smoothed output. Such bias can be solved by applying the loss on a carefully selected *subset* of pixels. For example, in [40], pair-wise ranking loss was applied to the pixels near instance boundaries to improve the quality of monocular depth estimation. However, such effort has not been made for surface normal estimation.

Figure 2. Ground truth surface normal of NYUv2 [33], generated by Ladicky et al. [22]. The ground truth is unreliable especially near object boundaries and on small structures.

In this work, we estimate the aleatoric uncertainty by predicting the *probability distribution* of the per-pixel surface normal. While the von Mises-Fisher distribution [8] can be used for this purpose, minimizing its negative log-likelihood (NLL) is equivalent to minimizing the L_2 distance between the predicted normal and the ground truth with learned loss attenuation. As the error metric of our interest is the *angle* between the two vectors, we introduce a new parameterization for the distribution such that its NLL is the angular loss with learned attenuation. At test time, the expected angular error is calculated from the estimated distribution, and used as a measure of the aleatoric uncertainty.

We also propose a novel decoder framework to improve the level of detail in the prediction. The network initially makes a coarse prediction for which the training loss is applied to all pixels. Then, the coarse prediction and feature-map are bilinearly upsampled by a factor of 2, and are passed through a pixel-wise multi-layer perceptron (MLP) to yield a refined output. This process is repeated until reaching the desired resolution. The MLPs are trained on a subset of pixels selected based on the uncertainty: Pixels with the highest uncertainty are selected and are complemented with uniformly sampled pixels. Such uncertainty-guided sampling prevents the bias in training towards large planar surfaces (for which the network estimates low uncertainty), thereby improving the quality of prediction near object boundaries and on small structures.

Our contributions can be summarized as follows:

- **Estimation of the aleatoric uncertainty in surface normal.** To the best of our knowledge, we are the first to estimate the aleatoric uncertainty in CNN-based surface normal estimation. We introduce a new parameterization for the surface normal probability distribution and show that the estimated uncertainty correlates well with the prediction error.
- **Uncertainty-guided sampling for pixel-wise refinement.** We introduce a novel decoder module where the loss is applied to a subset of pixels selected based on the uncertainty. We show that this module significantly improves the quantitative and qualitative performance.
- **State-of-the-art performance.** Experimental results show that the proposed method achieves state-of-the-art performance on ScanNet [4] and NYUv2 [33]. Qualitatively, the prediction made by our method contains a higher level of detail (see Fig. 1).

2. Related Work

Surface normal estimation. Surface normal estimation from a single RGB image has been studied extensively in literature [9, 10, 22, 39, 7, 1, 37, 31, 18, 42, 24, 32, 6, 38]. The existing methods generally consist of a feature extractor followed by a prediction head. For example, Ladicky et al. [22] extracted hand-crafted features (e.g., SIFT [26]) and applied multi-class Ada-boost [36] to regress the output as a linear combination of a discrete set of normals. Following the success of deep learning, recent methods replace both components with convolutional neural networks (CNNs).

Wang et al. [39] introduced two-stream CNNs to learn global and local cues, and fused them with another CNN. Eigen and Fergus [7] proposed a multi-scale architecture to jointly predict depth, surface normals and semantic labels. Following these early attempts, contributions have been made by enforcing depth-normal consistency [31, 32], formulating the task as spherical regression [24], and introducing a spatial rectifier to handle tilted images [6]. In this work, we address the aleatoric uncertainty in surface normal, which has not been studied in previous literature.

Uncertainty in deep learning. Two major types of uncertainty are epistemic and aleatoric [5]. Epistemic uncertainty (i.e. uncertainty in model) can be modelled by approximating the posterior over the model weights. For example, by applying dropout [35] at test time, N networks can be sampled from the approximate posterior, and the variance of the outputs can be used as a measure of uncertainty [11]. The posterior can also be approximated by training N networks on random subsets of data [23], or by taking N snapshots during a single training [17]. The aforementioned approaches are task-independent and can easily be applied to surface normal estimation.

The focus of this paper is on the aleatoric uncertainty, which captures the noise inherent in the data. We assume that the uncertainty is heteroscedastic [20] (i.e. certain pixels have higher uncertainty than the others). For such a scenario, a commonly used approach is to estimate the per-pixel probability distribution over the output, and train the network by maximizing the likelihood of the ground truth [20, 12]. This requires a task-specific formulation and has not been studied for CNN-based surface normal estimation.

Distribution on a unit sphere. The surface normal probability distribution should be defined on a unit sphere. An example of such distribution is the von Mises-Fisher distribution [8], a rotationally symmetric uni-modal distribution defined on an n-sphere. In this paper, we introduce a variant

of the von Mises-Fisher distribution, such that minimizing its negative log-likelihood is equivalent to minimizing the angle between the predicted normal and the ground truth, which is the error metric of our interest.

Uncertainty-guided sampling. PointRend [21] is a neural network module designed for instance/semantic segmentation. As making inference on a regular grid leads to undersampling of the pixels near object boundaries, PointRend uses a point-wise MLP to make inference on a subset of pixels with high uncertainty. Our decoder module is a novel extension of such a framework to surface normal estimation.

3. Method

This section provides the details of our method. Firstly, we introduce a new parameterization for the surface normal probability distribution that can be used for uncertainty estimation. Secondly, we explain the network architecture and the uncertainty-guided sampling used for training the pixel-wise refinement networks.

3.1. Aleatoric Uncertainty in Surface Normal

Our goal is to learn the per-pixel surface normal probability distribution $p_i(\mathbf{n}_i|\mathcal{I})$, where i is the pixel index and $\mathcal{I}$ is the input image. In practice, we parameterize the distribution with a set of parameters $\boldsymbol{\theta}_i$, which is estimated by a network of weights $\mathbf{W}$. The network is trained by minimizing the negative log-likelihood (NLL) of the ground truth $\mathbf{n}_i^{\text{gt}}$. The training loss can thus be written as

$$\mathcal{L} = -\frac{1}{N}\sum_i \log p_i(\mathbf{n}_i^{\text{gt}}|\boldsymbol{\theta}_i(\mathcal{I}, \mathbf{W})), \tag{1}$$

where N is the number of pixels with ground truth. Finding a suitable parameterization for the distribution is important as it determines which quantity will be minimized (or maximized) during training.

von Mises-Fisher distribution. We use the von Mises-Fisher distribution [8] (abbreviated hereafter as vonMF) as a baseline. It is a spherical analogue to the normal distribution, defined on a unit n-sphere in $\mathbb{R}^{n+1}$ [15]. For $n = 2$, the probability density function (PDF) is given as

$$p_{\text{vonMF},i}(\mathbf{n}_i|\boldsymbol{\mu}_i, \kappa_i) = \frac{\kappa_i \exp(\kappa_i \boldsymbol{\mu}_i^T \mathbf{n}_i)}{4\pi \sinh \kappa_i}, \tag{2}$$

where $\boldsymbol{\mu}_i$ is the mean direction and κ_i is the concentration parameter. Both $\mathbf{n}_i$ and $\boldsymbol{\mu}_i$ are unit vectors and $\kappa_i \geq 0$. Higher value of κ_i means that the distribution is more concentrated around $\boldsymbol{\mu}_i$ and that the uncertainty is low for that pixel (the distribution is uniform when $\kappa_i = 0$). The pixel-wise NLL loss can be written as

$$\mathcal{L}_{\text{vonMF},i} = -\log \kappa_i + \log \sinh \kappa_i - \kappa_i \boldsymbol{\mu}_i^T \mathbf{n}_i^{\text{gt}}. \tag{3}$$

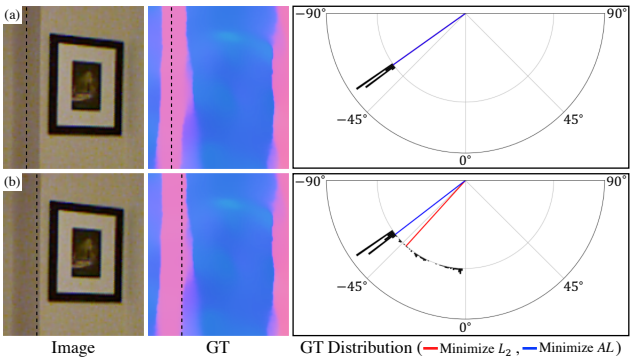

Figure 3. Each histogram shows the distribution of ground truth along the dashed line. The red and the blue lines show the direction that minimizes the L_2 loss and the angular loss, respectively (the lines overlap for (a)). In both examples, the pixels along the dashed line have similar visual features and belong to the same plane. However, the pixels in (b) suffer from the noise caused by the neighboring pixels belonging to a different plane. The angular loss is more robust in the presence of such asymmetric noise.

Maximizing $\boldsymbol{\mu}_i^T \mathbf{n}_i^{gt}$ is equivalent to minimizing the L_2 distance $|\boldsymbol{\mu}_i - \mathbf{n}_i^{gt}|_2^2$. The loss is attenuated for the pixels with high uncertainty. The first two terms in Eq. 3 prevent the network from predicting infinite κ for all pixels. To summarize, Eq. 3 is an L_2 loss with learned attenuation.

***Angular* vonMF distribution.** While Eq. 3 minimizes L_2, we argue that the loss should minimize the *angle* between the predicted normal and the ground truth, $\cos^{-1} \boldsymbol{\mu}_i^T \mathbf{n}_i^{gt}$. Firstly, this makes the loss consistent with the error metric. Secondly, this makes the network more robust against the *asymmetric* noise in the ground truth surface normal.

The ground truth surface normal of a pixel is obtained by fitting a plane to the point cloud defined by the pixel and its local neighborhood. If some of the neighboring pixels belong to a different plane (e.g., because the central pixel is close to the plane boundary), the ground truth will be affected accordingly and the noise in the ground truth will be asymmetric around the true normal. The mean direction, which minimizes the L_2 loss, is sensitive to such asymmetric noise. The angular loss, on the other hand, is minimized at the median direction, which is more robust against such noise (see Fig. 3). To this end, we introduce a distribution such that its NLL is the angular loss with learned attenuation. The PDF and the NLL loss are given as,

$$p_{\text{AngMF},i}(\mathbf{n}_i|\boldsymbol{\mu}_i, \kappa_i) = \frac{(\kappa_i^2 + 1)\exp(-\kappa_i \cos^{-1} \boldsymbol{\mu}_i^T \mathbf{n}_i)}{2\pi(1 + \exp(-\kappa_i \pi))} \tag{4}$$

$$\text{and} \quad \mathcal{L}_{\text{AngMF},i} = -\log(\kappa_i^2 + 1) + \log(1 + \exp(-\kappa_i \pi)) + \kappa_i \cos^{-1} \boldsymbol{\mu}_i^T \mathbf{n}_i^{\text{gt}}. \tag{5}$$

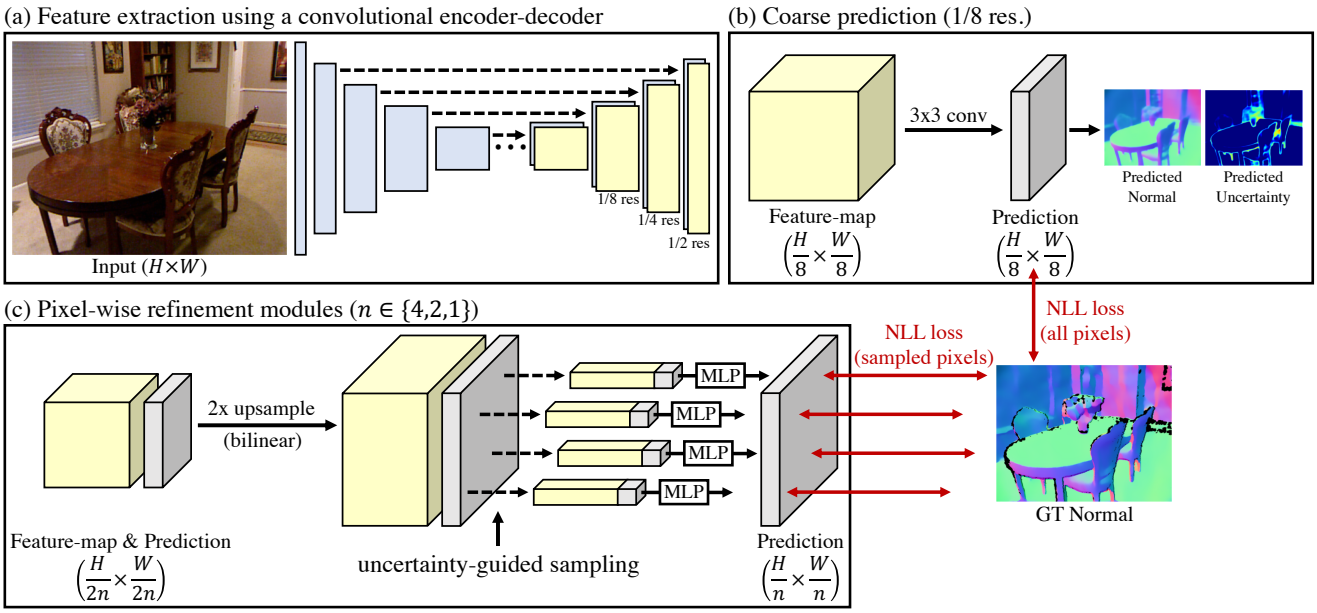

Figure 4. Illustration of the proposed pipeline. Initially, a coarse prediction is made from the 1/8 resolution feature-map and the loss is applied to all pixels. Then, a refinement module upsamples the coarse feature-map and prediction by a factor of 2, and applies a pixel-wise MLP to yield a refined, higher resolution output. Full-resolution output is obtained by applying three refinement modules. The MLPs are trained on a subset of pixels selected based on the uncertainty, to prevent the bias in training towards low-uncertainty pixels.

We call this the *Angular* vonMF distribution (abbreviated hereafter as AngMF). Eq. 4 is obtained by setting the NLL as $\mathcal{L}_i = C(\kappa_i) + \kappa_i \cos^{-1} \boldsymbol{\mu}_i^T \mathbf{n}_i$ and finding the expression for $C(\kappa_i)$ via normalization (derivation in the supplementary material). Minimizing Eq. 5 is equivalent to minimizing the angular error, while attenuating the loss for the pixels with high uncertainty (i.e. low κ). We show in the experiments that using the proposed AngMF leads to higher accuracy than using the vonMF.

Measure of uncertainty. In the proposed distribution (Eq. 4), κ_i encodes the network's confidence in the predicted mean $\boldsymbol{\mu}_i$. To translate this into an intuitive quantity, we calculate the *expected value* of the angular error

$$E[\cos^{-1} \boldsymbol{\mu}_i^T \mathbf{n}_i] = \frac{2\kappa_i}{\kappa_i^2 + 1} + \frac{\exp(-\kappa_i \pi)\pi}{1 + \exp(-\kappa_i \pi)}, \quad (6)$$

and use it as a measure of the pixel-wise aleatoric uncertainty (derivation in the supplementary material).

3.2. Uncertainty-Guided Sampling for Pixel-Wise Refinement

The NLL losses (Eq. 3 and Eq. 5) are more robust against noisy data than their counterparts (L_2 and angular loss) as the loss is attenuated for high-uncertainty pixels. However, this also makes the training more biased to large planar surfaces that have low surface normal uncertainty. Such bias leads to the lack of detail in the prediction, as the network is not encouraged to make accurate predictions for the challenging pixels, most of which are near object boundaries and on small structures. To this end, we propose a novel decoder framework, where pixel-wise multilayer perceptrons (MLPs) are trained on a subset of pixels selected based on the estimated uncertainty.

Feature extraction. The proposed pipeline is illustrated in Fig. 4. The input to the network is an RGB image of size $(H \times W)$. We first generate feature-maps of different resolutions, using a convolutional encoder-decoder with skip-connections. We use the same architecture as [2].

Coarse prediction. The network initially makes a coarse prediction from the 1/8 resolution feature-map, using a 3×3 convolutional layer. The number of output channels is 4 (3 for $\boldsymbol{\mu}$ and 1 for κ). The first three channels are L_2-normalized to ensure $||\boldsymbol{\mu}|| = 1$. We apply the modified ELU function [3], $f(x) = \text{ELU}(x) + 1$, for the last channel to ensure that κ is positive. For the coarse prediction, the training loss (Eq. 5) is applied to *all* pixels.

Pixel-wise refinement modules. The coarse prediction is then passed through three pixel-wise refinement modules of the same architecture. The input to each module is a low-resolution feature-map and prediction of size $(H/2n \times W/2n)$, and the output is a refined prediction of size $(H/n \times W/n)$. The forward pass in each module consists of three steps. *(1) Upsampling:* Both the feature-map

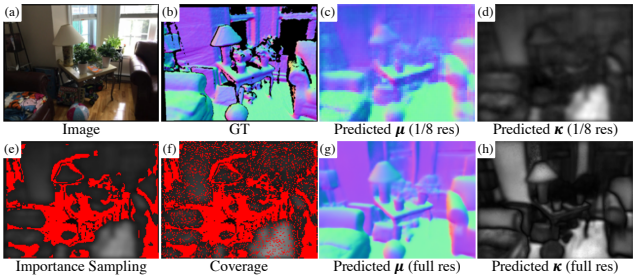

Figure 5. (a-b) Input image and the ground truth. (c-d) Prediction made in coarse resolution, during the first epoch. In (d) and (h), white means high κ. The network estimates low κ (i.e. high uncertainty) for most pixels except for those on the floor. If the NLL loss is applied to all pixels, the pixels on the floor will dominate the training as our loss is weighted by κ. (e-f) Uncertainty-guided sampling. We sample the pixels with high uncertainty (importance sampling) and add uniformly sampled pixels (coverage). Such sampling helps the network to focus on the challenging pixels. (g-h) Prediction made in full-resolution in the final epoch. The prediction is improved especially on the challenging pixels near object boundaries and on small structures. The network also becomes more confident about such pixels.

and prediction are bilinearly upsampled by a factor of 2. *(2) Uncertainty-guided sampling:* During training, a subset of pixels is selected base on the uncertainty. The sampling strategy is explained below in more detail. *(3) Pixel-wise refinement:* An MLP with three hidden layers, each with 128 nodes and a ReLU [27] activation, estimates a refined output for the sampled pixels. The input to the MLP is a concatenated vector of the pixel-wise feature and prediction. Same as in the coarse prediction layer, L_2 normalization and the modified ELU activation are applied to $\boldsymbol{\mu}$ and κ. During training, the loss is calculated only for the sampled pixels. At test time, the trained MLPs are applied to all pixels.

Uncertainty-guided sampling. Suppose that there are $h \cdot w$ pixels in the bilinearly upsampled prediction. In total, we sample $N_s = r_s \cdot h \cdot w$ pixels, where r_s is set to 0.4 in all experiments. Firstly, we sample $\beta_{\text{UG}} \cdot N_s$ pixels with the highest uncertainty (i.e. importance sampling). Then, $(1 - \beta_{\text{UG}}) \cdot N_s$ pixels are sampled uniformly from the remaining pixels (i.e. coverage). β_{UG}, which can have values from 0 to 1, determines how biased the sampling is towards the high-uncertainty pixels. Fig. 5 illustrates the sampling process.

4. Experimental Setup

Datasets. We evaluate our method on two datasets: ScanNet [4] and NYUv2 [33]. ScanNet contains RGB-D frames from 1613 scans acquired in 807 different scenes. We use the ground truth surface normal and data split provided by FrameNet [18]. NYUv2 consists of RGB-D video sequences capturing 464 indoor scenes. We evaluate on the official test set using the ground truth generated by Ladicky et al. [22]. As the official training set only contains 795 images, state-of-the-art methods sample additional images from the training sequences [39, 1, 31, 32] or supplement with other datasets [24, 14]. To ensure a fair comparison, we use the same training set as GeoNet++ [32].

Surface normal accuracy metrics. Angular error is measured for the pixels with valid ground truth. Following [9], we report the mean, median and root-mean-squared error (lower the better), and the percentage of pixels with error below thresholds $t \in [11.25°, 22.5°, 30°]$ (higher the better).

Uncertainty metrics. The significance of the estimated uncertainty can be evaluated using sparsification curves [30]. The pixels are sorted based on the uncertainty and an error metric ϵ is evaluated on the top $x\%$ of pixels with low uncertainty. Following [30], we transform the accuracy metric (% of pixels with error less than $t°$) into an error metric by subtracting it from 100%. We vary x from 1 to 100, incrementing by 1, and report the area under the sparsification curve (AUSC) as in [16]. AUSC is affected by two factors: how accurate the predictions are, and how similar the uncertainty-based sorting is to the actual error-based sorting. To only evaluate the latter, we also report the area under the sparsification error (AUSE) [19], by subtracting the oracle sparsification (obtained via error-based sorting) from the estimated sparsification.

Implementation details. The proposed network is implemented with PyTorch [28]. For training, we use the AdamW optimizer [25] and schedule the learning rate using 1cycle policy [34] with $lr_{\text{max}} = 3.5 \times 10^{-4}$ (other hyper-parameters are set as their default values). The batch size is 4 and the number of epochs is 5 unless specified otherwise.

5. Experiments

Firstly, we perform a set of ablation studies to demonstrate the effectiveness of the proposed approach. Then, the accuracy is compared against the state-of-the-art methods. Lastly, we evaluate the quality of the estimated uncertainty and compare it against alternative methods of uncertainty estimation.

5.1. Ablation Study

The ablation study experiments are performed on a subset of ScanNet [4], obtained by sampling 20% of the images in the training set (which contains 190K images).

Training loss. *NLL-vonMF* (Eq. 3) is the L_2 loss with learned attenuation, and the proposed *NLL-AngMF* (Eq. 5) is the angular loss (*AL*) with learned attenuation. We compare the four loss functions in Tab. 1 (top). As L_2 and *AL* cannot be used for uncertainty estimation, the decoder modules are removed, and the surface normal is directly estimated from the convolutional encoder-decoder, by adding a 3×3 convolutional layer to the final feature-map. Following are the key insights we can obtain from this experiment.

Architecture	Loss fn.	mean	median	rmse	5.0°	7.5°	11.25°	22.5°	30°
baseline (convolutional encoder-decoder with skip connections [2])	L_2	13.53	7.22	**21.16**	35.10	51.44	65.08	82.38	87.83
	NLL-vonMF	14.10	7.19	22.14	36.20	51.46	64.09	80.80	86.34
	AL	13.45	6.70	21.78	38.65	54.04	66.73	82.46	87.53
	NLL-AngMF	13.82	6.60	22.47	39.69	54.30	65.97	81.64	86.71
baseline + pixel-wise MLPs	*NLL-AngMF*	13.59	6.53	22.23	39.92	54.79	67.03	82.18	87.06
baseline + pixel-wise MLPs + uncertainty-guided sampling		**13.17**	**6.48**	21.57	**40.09**	**55.19**	**67.62**	**83.10**	**87.97**

Table 1. (top) The baseline network is trained with different loss functions. The proposed *NLL-AngMF* shows higher accuracy than *NLL-vonMF*, except for RMSE. *NLL-AngMF* and *NLL-vonMF* are *AL* and L_2 with learned attenuation, respectively. As the training is biased to low-uncertainty pixels, the median error decreases, while RMSE increases. (bottom) The bias in training is solved by the proposed decoder modules. Both the pixel-wise MLPs and the uncertainty-guided sampling lead to improvement in all metrics.

- ***NLL-AngMF* vs. *NLL-vonMF*.** While *NLL-vonMF* minimizes L_2, the proposed *NLL-AngMF* minimizes the angular error, which is more consistent with the error metrics. As a result, *NLL-AngMF* achieves significantly higher accuracy than *NLL-vonMF*, except for RMSE.
- ***NLL-AngMF* vs. *AL*.** Our *NLL-AngMF* is *AL* with learned attenuation. As the training is biased to low-uncertainty pixels (mostly on large surfaces), the median error decreases and the accuracy for low thresholds (5.0° and 7.5°) increases. On the contrary, the mean error and RMSE increase and the accuracy for higher thresholds decreases. This is because the network is not penalized strongly for making inaccurate predictions for the challenging pixels.

Decoder architecture. Tab. 1 (bottom) demonstrates the effectiveness of the proposed decoder modules. Firstly, we add the pixel-wise MLPs and train them on all pixels. Then, we apply the uncertainty-guided sampling during training (with $\beta_{\text{UG}} = 0.7$). Both components lead to improvement in all metrics. As the uncertainty-guided sampling prevents the bias in training towards large planar surfaces, the quality of prediction is improved especially near object boundaries and on small structures, as shown in Fig. 6.

Sampling strategy. Tab. 2 shows how the accuracy changes for different values of β_{UG}. β_{UG} determines the ratio of the importance sampling. If $\beta_{\text{UG}} = 1.0$, only the pixels with the highest uncertainty are sampled. If $\beta_{\text{UG}} = 0.0$, the pixels are sampled uniformly. Finding the right balance between the two is important for minimizing the bias in training. Best performance is achieved when $\beta_{\text{UG}} = 0.7$.

5.2. Comparison with the State-of-the-Art

NYUv2. Tab. 3 compares the accuracy of different methods on NYUv2 [33]. Note that, compared to ScanNet [4], the quality of the ground truth is noticeably worse for NYUv2. While the ground truth for ScanNet is calculated from a 3D mesh that is obtained by fusing thousands of RGB-D frames, the ground truth for NYUv2 is calculated from a single noisy depth map. Nonetheless, the proposed training loss (angular loss with learned attenuation) and decoder framework (trained with uncertainty-guided sampling) help the network to learn from noisy data. As a result, our network shows a decisive improvement over GeoNet++ [32]. Qualitative comparison in Fig. 7 shows that the predic-

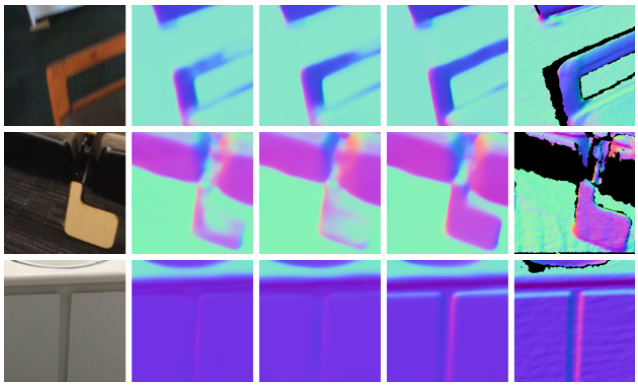

Figure 6. Qualitative comparison between the networks with different decoder architecture (showing crops of 200 pixels × 200 pixels). The proposed uncertainty-guided sampling (UG) enforces the network to focus on the challenging pixels (i.e. those with high uncertainty). This improves the level of detail in the prediction.

β_{UG}	mean	median	rmse	11.25°	22.5°	30°
0.0	13.58	6.52	22.18	66.68	82.09	87.09
0.6	13.34	6.56	21.76	66.99	82.78	87.74
0.7	**13.17**	**6.48**	**21.57**	**67.62**	**83.10**	**87.97**
0.8	13.28	6.56	21.69	67.45	83.00	87.90
1.0	13.26	6.59	**21.57**	67.16	82.98	87.92

Table 2. Influence of β_{UG} on the accuracy (r_s is fixed to 0.4). β_{UG} is the ratio of the importance sampling. Best performance is achieved when $\beta_{\text{UG}} = 0.7$.

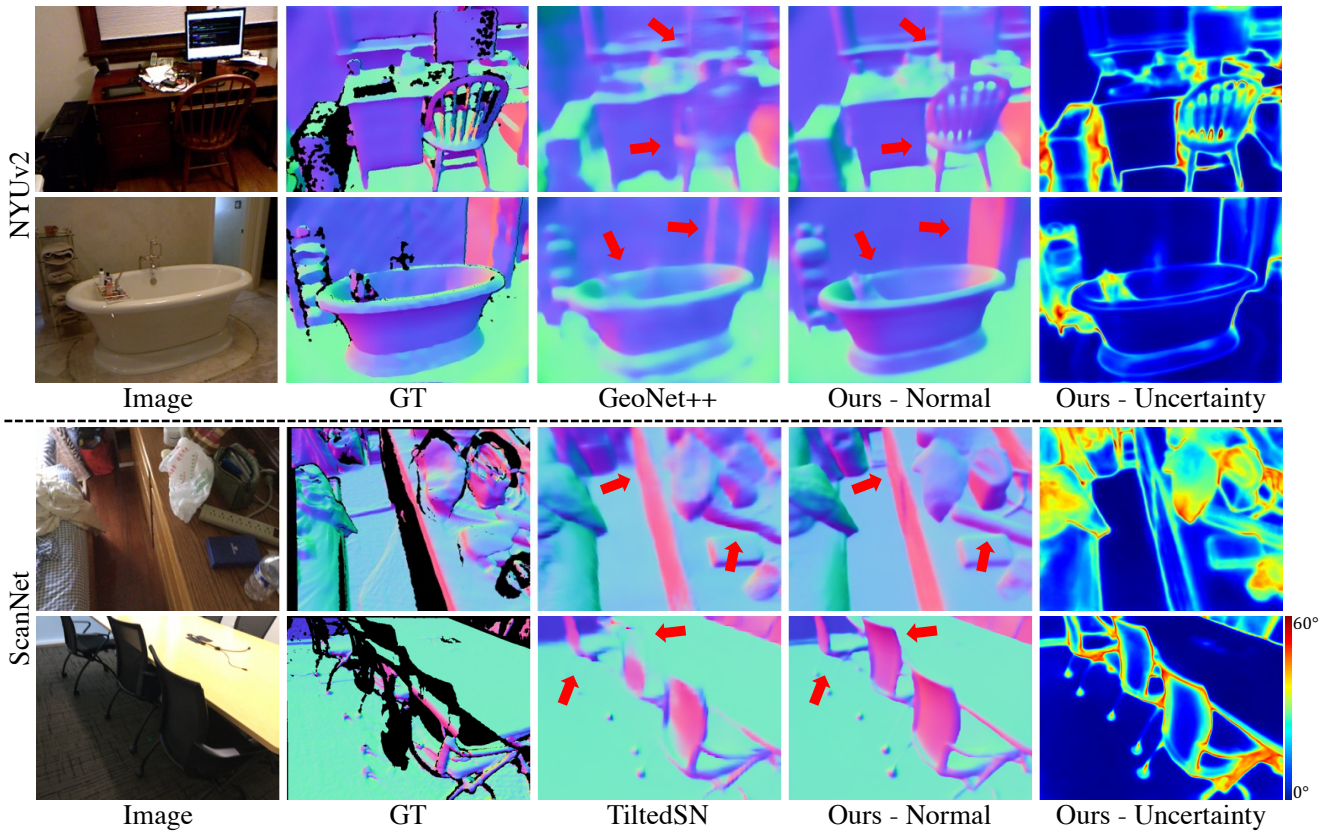

Figure 7. Qualitative comparison against GeoNet++ [32] and TiltedSN [6]. The predictions made by our method show clearer object boundaries and preserve the fine-details of the scene geometry (see the regions pointed by the red arrows). The estimated uncertainty is high near object boundaries and on small structures. More examples are provided in the supplementary material.

Method	Train	mean	median	rmse	11.25°	22.5°	30°
Ladicky et al. [22]		33.5	23.1	-	27.5	49.0	58.7
Fouhey et al. [10]		35.2	17.9	-	40.5	54.1	58.9
Deep3D [39]		26.9	14.8	-	42.0	61.2	68.2
Eigen et al. [7]		20.9	13.2	-	44.4	67.2	75.9
SkipNet [1]		19.8	12.0	28.2	47.9	70.0	77.8
SURGE [37]	N	20.6	12.2	-	47.3	68.9	76.6
GeoNet [31]		19.0	11.8	26.9	48.4	71.5	79.5
PAP [42]		18.6	11.7	25.5	48.8	72.2	79.8
GeoNet++ [32]		18.5	11.2	26.7	50.2	73.2	80.7
Ours	N	**14.9**	**7.5**	**23.5**	**62.2**	**79.3**	**85.2**
FrameNet[18]		18.6	11.0	26.8	50.7	72.0	79.5
VPLNet[38]	S	18.0	9.8	-	54.3	73.8	80.7
TiltedSN[6]		16.1	**8.1**	25.1	**59.8**	77.4	83.4
Ours	S	**16.0**	8.4	**24.7**	59.0	**77.5**	**83.7**

Table 3. Surface normal accuracy on NYUv2 [33]. The proposed method shows state-of-the-art performance. (top) The networks are trained on NYUv2. (bottom) The networks are trained on ScanNet [4] and tested on NYUv2 without fine-tuning.

Method	mean	median	rmse	11.25°	22.5°	30°
FrameNet[18]	14.7	7.7	22.8	62.5	80.1	85.8
VPLNet[38]	13.8	6.7	-	66.3	81.8	87.0
TiltedSN[6]	12.6	6.0	21.1	69.3	83.9	88.6
Ours	**11.8**	**5.7**	**20.0**	**71.1**	**85.4**	**89.8**

Table 4. Surface normal accuracy on ScanNet [4]. Our method outperforms other methods across all metrics.

tions made by our method contain a higher level of detail. We also train the network on ScanNet and test on NYUv2 without fine-tuning. In this cross-dataset evaluation, we win against other methods except for the median error and 11.25°, suggesting that the network can generalize well to an unseen dataset.

ScanNet. Tab. 4 compares different methods trained and tested on ScanNet [4]. The batch size is set to 16 for this experiment. We outperform the state-of-the-art methods across all metrics. Qualitative comparison against TiltedSN [6] is provided in Fig. 7.

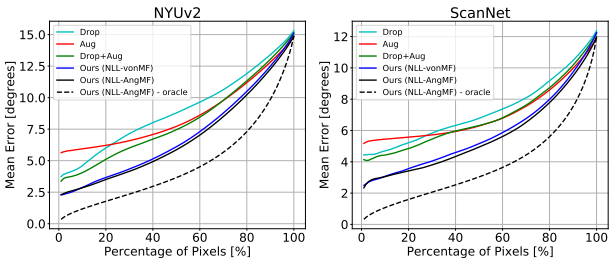

Figure 8. Sparsification curves obtained by different methods of estimating the surface normal uncertainty.

Method	AUSC ↓			AUSE ↓		
	mean	rmse	11.25°	mean	rmse	11.25°
Drop	9.01	15.84	19.32	4.02	9.61	10.23
Aug	8.64	15.08	18.75	3.93	9.14	10.25
Drop + Aug	8.16	14.32	16.73	3.22	8.15	7.75
Ours *(NLL-vonMF)*	7.03	10.96	14.24	**2.11**	**4.80**	5.10
Ours *(NLL-AngMF)*	**6.83**	**10.92**	**13.47**	2.13	4.98	**5.01**

Table 5. Quantitative evaluation of uncertainty on NYUv2 [33].

Method	AUSC ↓			AUSE ↓		
	mean	rmse	11.25°	mean	rmse	11.25°
Drop	7.25	12.51	13.95	3.24	7.55	8.58
Aug	7.06	12.58	13.72	3.32	7.92	8.81
Drop + Aug	6.87	12.07	12.73	2.93	7.20	7.49
Ours *(NLL-vonMF)*	5.84	9.30	10.31	**1.85**	**4.38**	4.69
Ours *(NLL-AngMF)*	**5.64**	**9.07**	**9.48**	1.88	**4.38**	**4.47**

Table 6. Quantitative evaluation of uncertainty on ScanNet [4].

5.3. Quality of Uncertainty

Lastly, we evaluate the quality of the estimated uncertainty by plotting the sparsification curves. As no previous work has estimated the surface normal uncertainty, we compare our method against task-independent approaches. *(1) Test-time dropout (Drop):* 2D dropout ($p = 0.2$) is added after each 2D convolutional block in decoder. After training, 8 forward passes are performed, with dropout enabled. *(2) Test-time augmentation (Aug):* Following [30], we perform 2 forward passes by flipping the input image. *(3) Combined approach (Drop + Aug):* We apply the image flipping to the network with dropout to make $2 \times 8 = 16$ forward passes. For all three methods, the uncertainty is measured as the average angular error with respect to the mean direction. As the uncertainty cannot be estimated in a single forward pass, the uncertainty-guided sampling is disabled, and the networks are trained with the angular loss. Quantitative results in Tab. 5 and Tab. 6 show that the proposed method outperforms other methods across all metrics. Fig. 8 compares the sparsification curves. When evaluated on all pixels, all methods perform similarly. However, as the pixels with high uncertainty are removed, our method gets significantly more accurate than the others. This suggests that our uncertainty correlates better with the prediction error (see Fig. 9 for qualitative comparison).

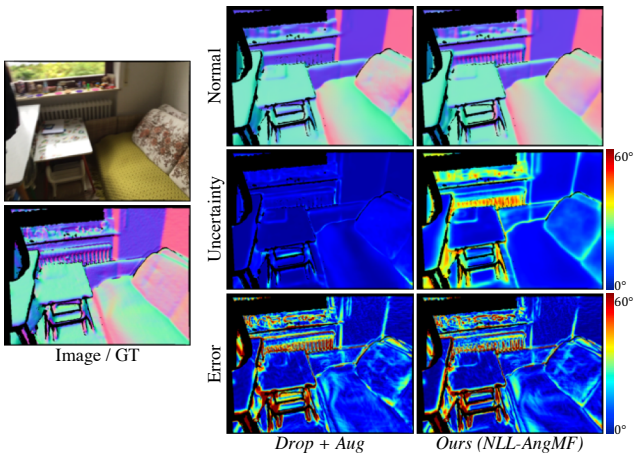

Figure 9. We compare the uncertainty estimated by our method against the uncertainty estimated by applying test-time dropout and augmentation (*Drop+Aug*). The uncertainty estimated by our method shows higher correlation with the prediction error.

5.4. Supplementary Material

In the supplementary material, we provide the derivations for the AngMF distribution, quantitative evaluation with additional metrics, cross-dataset evaluation on KITTI [13] and DAVIS [29] and discussion on failure modes.

6. Conclusion

In this work, we estimated and evaluated the aleatoric uncertainty in CNN-based surface normal estimation, for the first time in literature. The proposed method estimates the per-pixel surface normal probability distribution, from which the expected angular error can be inferred to quantify the aleatoric uncertainty. We introduced a new parameterization for the surface normal probability distribution, such that its negative log-likelihood is the angular loss with learned attenuation. We also proposed a novel decoder framework where pixel-wise MLPs are trained on a subset of pixels selected based on the uncertainty. Such uncertainty-guided sampling prevents the bias in training towards large planar surfaces, thereby improving the level of detail in the prediction. Experimental results show that the proposed method achieves state-of-the-art performance on ScanNet [4] and NYUv2 [33], and that the estimated uncertainty correlates well with the prediction error.

References

[1] Aayush Bansal, Bryan Russell, and Abhinav Gupta. Marr revisited: 2d-3d alignment via surface normal prediction. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 2, 5, 7

[2] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 4, 6

[3] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). In *International Conference on Learning Representations (ICLR)*, 2016. 4

[4] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2, 5, 6, 7, 8

[5] Armen Der Kiureghian and Ove Ditlevsen. Aleatory or epistemic? does it matter? *Structural Safety*, 31(2):105–112, 2009. 2

[6] Tien Do, Khiem Vuong, Stergios I Roumeliotis, and Hyun Soo Park. Surface normal estimation of tilted images via spatial rectifier. In *Proc. of European Conference on Computer Vision (ECCV)*, 2020. 1, 2, 7

[7] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proc. of IEEE/CVF International Conference on Computer Vision (ICCV)*, 2015. 1, 2, 7

[8] Nicholas I Fisher, Toby Lewis, and Brian JJ Embleton. *Statistical analysis of spherical data*. Cambridge university press, 1993. 2, 3

[9] David F Fouhey, Abhinav Gupta, and Martial Hebert. Data-driven 3d primitives for single image understanding. In *Proc. of IEEE/CVF International Conference on Computer Vision (ICCV)*, 2013. 2, 5

[10] David Ford Fouhey, Abhinav Gupta, and Martial Hebert. Unfolding an indoor origami world. In *Proc. of European Conference on Computer Vision (ECCV)*, 2014. 2, 7

[11] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning (ICML)*, 2016. 2

[12] Jochen Gast and Stefan Roth. Lightweight probabilistic deep networks. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[13] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 32(11):1231–1237, 2013. 8

[14] Steven Hickson, Karthik Raveendran, Alireza Fathi, Kevin Murphy, and Irfan Essa. Floors are flat: Leveraging semantics for real-time surface normal prediction. In *Proc. of IEEE/CVF International Conference on Computer Vision Workshops*, 2019. 1, 5

[15] Thomas Hillen, Kevin J Painter, Amanda C Swan, and Albert D Murtha. Moments of von mises and fisher distributions and applications. *Mathematical Biosciences & Engineering*, 14(3):673, 2017. 3

[16] Xiaoyan Hu and Philippos Mordohai. A quantitative evaluation of confidence measures for stereo vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34(11):2121–2133, 2012. 5

[17] Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E Hopcroft, and Kilian Q Weinberger. Snapshot ensembles: Train 1, get m for free. In *International Conference on Learning Representations (ICLR)*, 2017. 2

[18] Jingwei Huang, Yichao Zhou, Thomas Funkhouser, and Leonidas J Guibas. Framenet: Learning local canonical frames of 3d surfaces from a single rgb image. In *Proc. of IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 1, 2, 5, 7

[19] Eddy Ilg, Ozgun Cicek, Silvio Galesso, Aaron Klein, Osama Makansi, Frank Hutter, and Thomas Brox. Uncertainty estimates and multi-hypotheses networks for optical flow. In *Proc. of European Conference on Computer Vision (ECCV)*, 2018. 5

[20] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 2

[21] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

[22] Lubor Ladicky, Bernhard Zeisl, and Marc Pollefeys. Discriminatively trained dense surface normal estimation. In *Proc. of European Conference on Computer Vision (ECCV)*, 2014. 2, 5, 7

[23] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 2

[24] Shuai Liao, Efstratios Gavves, and Cees GM Snoek. Spherical regression: Learning viewpoints, surface normals and 3d rotations on n-spheres. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 5

[25] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[26] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110, 2004. 2

[27] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning (ICML)*, 2010. 5

[28] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 5

[29] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung.

A benchmark dataset and evaluation methodology for video object segmentation. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 8

[30] Matteo Poggi, Filippo Aleotti, Fabio Tosi, and Stefano Mattoccia. On the uncertainty of self-supervised monocular depth estimation. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 5, 8

[31] Xiaojuan Qi, Renjie Liao, Zhengzhe Liu, Raquel Urtasun, and Jiaya Jia. Geonet: Geometric neural network for joint depth and surface normal estimation. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2, 5, 7

[32] Xiaojuan Qi, Zhengzhe Liu, Renjie Liao, Philip HS Torr, Raquel Urtasun, and Jiaya Jia. Geonet++: Iterative geometric neural network with edge-aware refinement for joint depth and surface normal estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020. 1, 2, 5, 6, 7

[33] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Proc. of European Conference on Computer Vision (ECCV)*, 2012. 1, 2, 5, 6, 7, 8

[34] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of residual networks using large learning rates. *arXiv preprint arXiv:1708.07120*, 2018. 5

[35] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research (JMLR)*, 15(1):1929–1958, 2014. 2

[36] Antonio Torralba, Kevin P Murphy, and William T Freeman. Sharing features: efficient boosting procedures for multiclass object detection. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004. 2

[37] Peng Wang, Xiaohui Shen, Bryan Russell, Scott Cohen, Brian Price, and Alan L Yuille. Surge: Surface regularized geometry estimation from a single image. In *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, 2016. 2, 7

[38] Rui Wang, David Geraghty, Kevin Matzen, Richard Szeliski, and Jan-Michael Frahm. Vplnet: Deep single view normal estimation with vanishing points and lines. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 7

[39] Xiaolong Wang, David Fouhey, and Abhinav Gupta. Designing deep networks for surface normal estimation. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1, 2, 5, 7

[40] Ke Xian, Jianming Zhang, Oliver Wang, Long Mai, Zhe Lin, and Zhiguo Cao. Structure-guided ranking loss for single image depth prediction. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[41] Peijiang Yuan, Qishen Wang, Tianmiao Wang, Chengkun Wang, and Bo Song. Surface normal measurement in the end effector of a drilling robot for aviation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2014. 1

[42] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Yan Yan, Nicu Sebe, and Jian Yang. Pattern-affinitive propagation across depth, surface normal and semantic segmentation. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 7