Uncertainty-Aware CNNs for Depth Completion: Uncertainty from Beginning to End

Abdelrahman Eldesokey Michael Felsberg Karl Holmquist Michael Persson
Computer Vision Laboratory, Linköping University, Sweden

Abstract

The focus in deep learning research has been mostly to push the limits of prediction accuracy. However, this was often achieved at the cost of increased complexity, raising concerns about the interpretability and the reliability of deep networks. Recently, an increasing attention has been given to untangling the complexity of deep networks and quantifying their uncertainty for different computer vision tasks. Differently, the task of depth completion has not received enough attention despite the inherent noisy nature of depth sensors. In this work, we thus focus on modeling the uncertainty of depth data in depth completion starting from the sparse noisy input all the way to the final prediction.

We propose a novel approach to identify disturbed measurements in the input by learning an input confidence estimator in a self-supervised manner based on the normalized convolutional neural networks (NCNNs). Further, we propose a probabilistic version of NCNNs that produces a statistically meaningful uncertainty measure for the final prediction. When we evaluate our approach on the KITTI dataset for depth completion, we outperform all the existing Bayesian Deep Learning approaches in terms of prediction accuracy, quality of the uncertainty measure, and the computational efficiency. Moreover, our small network with 670k parameters performs on-par with conventional approaches with millions of parameters. These results give strong evidence that separating the network into parallel uncertainty and prediction streams leads to state-of-the-art performance with accurate uncertainty estimates.

1. Introduction

The recent surge of deep neural networks (DNNs) has led to remarkable breakthroughs on several computer vision tasks, *e.g.* object classification and detection [31, 25, 22, 2], semantic segmentation [37, 30], and object tracking [6, 34]. However, this was achieved at the cost of increased model complexity, inducing new concerns such as: how do these black-box models infer their predictions? and how certain are they about these predictions? Failing to address these concerns impairs the reliability of DNNs. For instance,

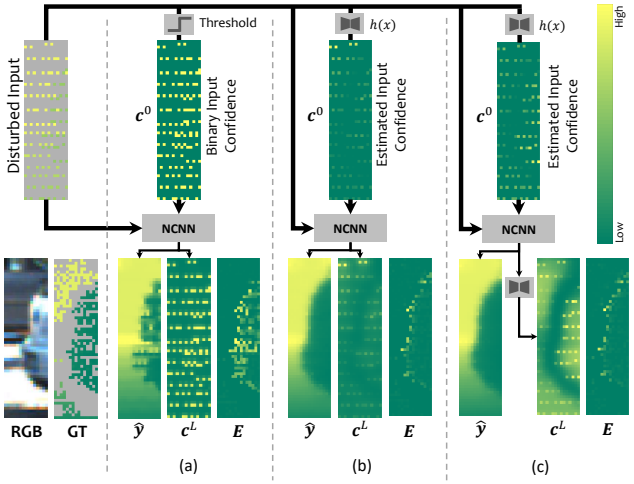

Figure 1. The confidence $\mathbf{c}^0$ for the input data is usually unknown. NCNNs [8] assume binary input confidence, which leads to severe artifacts **(a)**. We propose to learn the input confidence in a self-supervised manner, which leads to improved prediction **(b)**. However, the output confidence $\mathbf{c}^L$ is not strongly correlated with the error $\mathbf{E}$. Therefore, we propose a probabilistic version of NCNN that produces a proper output uncertainty measure **(c)**.

Huang *et al.* [13] showed that it is possible to fool state-of-the-art object detectors to produce false and highly certain predictions using physical and digital manipulations. Therefore, there is a compelling need for investigating interpretability and uncertainty of DNNs to be able to trust them in safety-critical environments.

Recently, a growing attention was given towards untangling the complexity of DNNs to enhance their reliability by analyzing how they make predictions and quantifying the uncertainty of these predictions. Probabilistic approaches such as Bayesian deep learning (BDL) have contributed to this endeavor by modifying DNNs to output the parameters of a probabilistic distribution, *e.g.* mean and variance, which yields uncertainty information about the predictions [18]. The availability of a reliable uncertainty measure facilitates the understanding of DNNs and applying safety procedures in case of model failure or high uncertainty. Several BDL approaches were proposed for different computer vision tasks such as object classification and segmen-

tation [9, 20, 18], optical flow [15, 10], and object detection [21, 5]. All these approaches assume undisturbed dense input images, but to the best of our knowledge, there exist no statistical approach that addresses sparse problems.

An essential task of this type is *scene depth completion*. Modeling uncertainty for this task is crucial due to the inherent noisy and sparse nature of depth sensors, caused by multi-path interference and depth ambiguities [11]. Previous approaches proposed to learn some intermediate confidence masks to mitigate the impact of disturbed measurements inside their networks [28, 33, 36]. However, none of these approaches has demonstrated the probabilistic validity of the intermediate confidence masks. Moreover, they do not provide an uncertainty measure for the final prediction. Therefore, it is still an open problem to fully model the uncertainty in DNN approaches to scene depth completion.

Gustafsson *et al.* [12] made an attempt by evaluating two of the existing BDL approaches for dense regression problems, *i.e*. MC-Dropout [9] and ensembling [20], on the task of depth completion. They utilized the Sparse-to-Dense network [24] as a baseline and modified it to estimate the parameters of a Gaussian distribution. Experiments on the KITTI-Depth dataset [32] showed that both approaches can produce high-quality uncertainty maps for the final prediction, but with the prediction accuracy severely degraded compared to the baseline model. Besides, both approaches train an ensemble of the baseline model requiring multiple inferences during test time. This leads to computational and memory overhead making these approaches unsuitable for the task of depth completion in practice due to their poor prediction accuracy and computational inefficiency.

Specifically designed for confidence-accompanied and sparse data are the normalized convolutional neural networks (NCNNs) [7, 8]. NCNNs consist of a serialization of confidence-equipped convolution layers that make use of an input confidence map. These layers produce the output of the convolution operation as well as an output confidence that is propagated to the following layer. When applied to the problem of depth completion, input confidences at the first layer are assumed to be binary following [32], ones at valid input points and zeros otherwise. However, this assumption is problematic since depth data can be disturbed as noted in the KITTI-Depth dataset [28]. Therefore, the use of binary masks for modeling input uncertainty in NCNNs becomes inappropriate, and hinders their use as the true input confidence is *unknown*. Also, the output confidence from NCNNs according to [7, 8] lacks any probabilistic interpretation that qualifies it as a reliable uncertainty measure.

1.1. Contributions

In this paper, we propose two main contributions. *First*, we employ the inherent dependency of NCNNs on the input confidence to train an estimator for this confidence in a self-supervised manner. Since disturbed measurements are expected to increase the prediction error, we back-propagate the error gradients to learn the input confidence that minimizes the error. This way, the network learns to assign low confidences to disturbed measurements that increase the error and high confidences to valid measurements. This approach establishes a new methodology for handling sparse and noisy data by *suppressing* the disturbed measurements before feeding them to the network. As shown empirically, this approach is more interpretable and efficient than utilizing a complex black-box model that is expected to implicitly rectify for the disturbed measurements.

Second, we derive a probabilistic NCNN (pNCNN) framework that produces meaningful uncertainty estimates in the probabilistic sense, whereas the output confidence from the standard NCNNs lacks any probabilistic characteristics. We formulate the training process as a maximum likelihood estimation problem and we derive the loss function for pNCNN training. These reformulations are the necessary extensions for fully Bayesian NCNNs.

By applying our approach to the task of unguided depth completion on the KITTI-Depth dataset [32], we achieve a remarkably better prediction accuracy at a very low computational cost compared to the existing BDL approaches. Moreover, the quality of the uncertainty measure from our *single* network is better than BDL approaches with ensembles of 1-32 networks. When compared against non-statistical approaches, we perform on par with state-of-the-art methods with millions of parameters using a significantly smaller network (670k parameters). Besides, and contrarily to state-of-the-art methods, we produce a high-quality prediction uncertainty measure aside with the prediction. Finally, we show that our approach is applicable to other sparse problems by evaluating it on multi-path interference correction [11] and sparse optical flow rectification.

2. Related Work

The task of scene depth completion is receiving an increasing attention due to the impact of depth information on different computer vision tasks. Typically, it aims to produce a dense and denoised depth map $\mathbf{y}$ from a noisy sparse input $\mathbf{x}$. Several approaches were proposed to learn a mapping $\mathbf{y} = f(\mathbf{x})$ by exploiting different input modalities, where f is a DNN. Ma *et al.* [24] proposed a deep regression model that combines the sparse input depth with the corresponding RGB modality. Jaritz *et al.* [16] evaluated different fusion schemes to combine the sparse depth with RGB images. Chen *et al.* [3] proposed a joint network that exploits 2D and 3D representations for the depth data. The key similarity between these approaches is that they all perform very well in terms of prediction accuracy and they implicitly handle disturbed measurements in the network. Nonetheless, none of these methods considered modeling

the uncertainty of the data or the prediction.

Recently, several approaches promoted the use of confidences to filter out noisy predictions within the network. Qui *et al.* [28] learned confidence masks from RGB images to mask out noisy depth measurements at occluded regions. Gansbeke *et al.* [33] proposed the use of confidences to fuse two network streams utilizing sparse depth and RGB images respectively. Similarly, Xu *et al.* [36] predict a confidence mask that is used to mitigate the impact of noisy measurements on different components of their network. However, none of these methods provided any prediction uncertainty measure for the final prediction.

This was addressed by another approach that utilizes confidences and provides an output confidence for the final prediction. Normalized convolutional neural networks (NCNNs) [7, 8] take sparse depth $\mathbf{x}$ and a confidence mask $\mathbf{c}^0$ as input, propagate the confidence, and produce a dense output $\mathbf{y}$ as well as an output confidence map $\mathbf{c}^L$, *i.e.*, $(\mathbf{y}, \mathbf{c}^L) = f(\mathbf{x}, \mathbf{c}^0)$, for a DNN with L layers. However, since the input confidence is unknown, a binary input confidence $\mathbf{c}^0$ is assumed, which is problematic in case of disturbed input as shown in Figure (1a). Further, the output confidence $\mathbf{c}^L$ has no probabilistic interpretation and shows no significant correlation with the prediction error.

To address these challenges, we look at the problem from a different perspective. We propose to learn the input confidence from the disturbed measurements by employing the confidence propagation property of NCNNs. We attach a network h to a NCNN and we train them end-to-end to learn the input confidence that minimizes the prediction error, *i.e.*, $(\mathbf{y}, \mathbf{c}^{\mathbf{L}}) = f(\mathbf{x}, h(\mathbf{x}))$. Further, to produce accurate uncertainty measure for the final prediction, we derive a probabilistic version of the NCNNs and we formulate the training as a maximum likelihood problem. When our proposed approach is evaluated on the KITTI-Depth dataset [32], it performs on par with state-of-the-art approaches with millions of parameters using a significantly smaller network, while providing a highly accurate uncertainty measure for the final prediction. In contrast to BDL approaches in [12], we achieve excellent uncertainty estimation without sacrificing prediction accuracy or computational efficiency.

The rest of the paper is organized as follows. We briefly describe the method of NCNNs in 3.1 and 3.2, and our proposed approach for learning the input confidence in section 3.3. Afterwards, we introduce a probabilistic version of NCNNs, derive the loss for training, and describe our architecture in section 4. Experiments and analysis are given in section 5. Finally, we conclude the paper in section 6.

3. Self-supervised Input Confidence Learning

The signal/confidence philosophy [19] promotes the separation between the signal and its confidence for efficiently handling noisy and sparse signals. For example, this separation allows differentiating missing signal points with no information from zero-valued valid points. The normalized convolution [19] is one approach that follows the this philosophy to perform the convolution operation.

For confidence-equipped signals, the normalized convolution performs convolution using only the confident points of the signal, while estimating the non-confident ones from their vicinity using some *applicability function*. This prevents noisy and missing measurements from disturbing the calculations. In this section, we give a brief description of normalized convolution and the trainable normalized convolution layer that can estimate an optimal applicability [7, 8]. Subsequently, we propose a novel approach to learn the input confidence in a self-supervised manner.

Throughout the paper, we assume a global signal $\mathcal{Y}$ with a finite size N that is convolved in a sliding window fashion. At each point in the signal y_i, a local signal $\mathbf{y}$ of size n constitutes the neighborhood at this point. The local signal $\mathbf{y}$ will be referred to as *the signal*, and y_i will be referred to as *the signal center*.

3.1. The Normalized Convolution

The fundamental idea of the normalized convolution is to project the confidence-equipped signal $\mathbf{y} \in \mathbb{C}^n$ to a new subspace spanned by a set of basis functions $\{\mathbf{b}_j\}_{j=0}^m$ using only the confident parts of the signal. Afterwards, the full signal is reconstructed from this subspace, where the non-confident parts are interpolated from their vicinity using a weighting kernel denoted as the *applicability function*. The confidence is provided as non-negative real vector $\mathbf{c} \in \mathbb{R}_+^n$ that has the same length as the signal $\mathbf{y}$, while the applicability $\mathbf{a} \in \mathbb{R}_+^n$ is usually chosen as some low-pass filter.

If we arrange the basis functions into the columns of a matrix $\mathbf{B}$, then the image of the signal under the subspace spanned by the basis is obtained as $\mathbf{y} = \mathbf{B}\mathbf{r}$, where $\mathbf{r}$ is a vector of coordinates. These coordinates can be estimated from a weighted least-squares problem (WLS) between the signal $\mathbf{y}$ and the image of it under the new basis:

$$\hat{\mathbf{r}}_{\text{WLS}} = \arg\min_{\mathbf{r} \in \mathbb{C}^m} \| \mathbf{B}\mathbf{r} - \mathbf{y} \|_{\mathbf{W}} \ , \tag{1}$$

where the weights matrix $\mathbf{W}$ is a product of $\mathbf{W}_\mathbf{a} = \text{diag}(\mathbf{a})$ and $\mathbf{W}_\mathbf{c} = \text{diag}(\mathbf{c})$. The WLS solution is given as [19]:

$$\hat{\mathbf{r}}_{\text{WLS}} = \underbrace{(\mathbf{B}^* \mathbf{W}_\mathbf{a} \mathbf{W}_\mathbf{c} \mathbf{B})^{-1}}_{\text{Reconstruct}} \underbrace{\mathbf{B}^* \mathbf{W}_\mathbf{a} \mathbf{W}_\mathbf{c} \mathbf{y}}_{\text{Project}} \ . \tag{2}$$

Finally, the WLS solution $\hat{\mathbf{r}}_{\text{WLS}}$ can be used to approximate the signal under the new basis as:

$$\hat{\mathbf{y}} = \mathbf{B}\hat{\mathbf{r}}_{\text{WLS}} \ . \tag{3}$$

3.2. Normalized Convolutional Neural Networks

In normalized convolution, the applicability is chosen manually. Eldesokey *et al.* [8] proposed a normalized convolutional neural network layer (NCNN) that utilized the

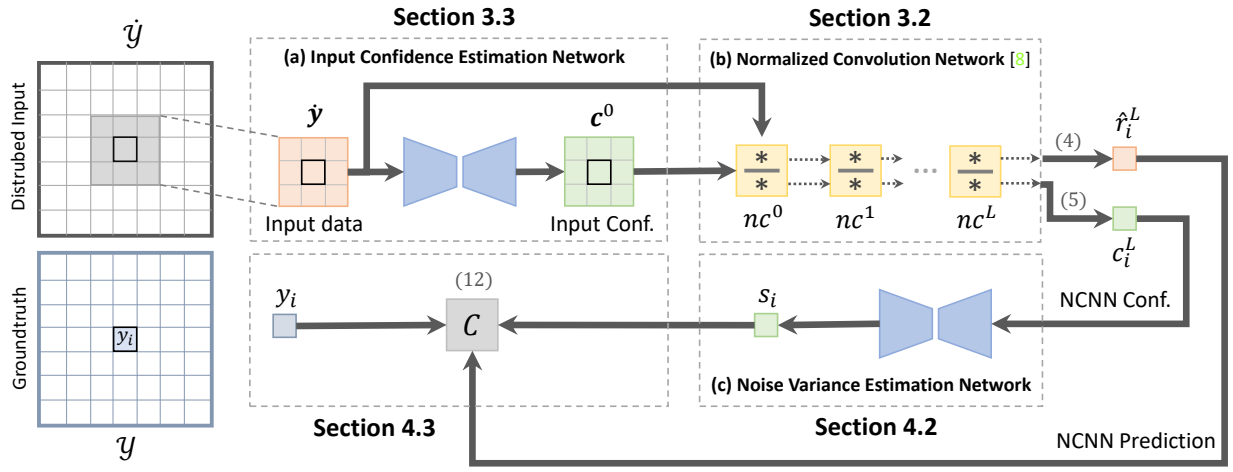

Figure 2. An overview of network architecture to predict a denoised signal $\mathcal{Y}$ from a disturbed signal $\dot{\mathcal{Y}}$. We show the pipeline for a single observation y_i of the whole signal $\mathcal{Y}$. Our contributions are described in sections 3.3, 4.2, and 4.3.

standard back-propagation in DNNs to learn the optimal applicability function $\mathbf{a}$ for a given dataset, while assuming a binary input confidence. This was achieved by using the naïve basis in (2), *i.e.* $\mathbf{B} = \mathbf{1}_n$:

$$\hat{r}_i = (\mathbf{1}_n^* \mathbf{W_a} \mathbf{W_c} \mathbf{1}_n)^{-1} \mathbf{1}_n^* \mathbf{W_a} \mathbf{W_c} \mathbf{y} = \frac{\langle \mathbf{a} | (\mathbf{y} \odot \mathbf{c}) \rangle}{\langle \mathbf{a} | \mathbf{c} \rangle}, \quad (4)$$

where $\mathbf{1}_n$ is a vector of ones, $\odot$ is the Hadamard product, $\langle .|. \rangle$ is the scalar product, $\hat{r}_i$ is a scalar which is equivalent to the estimated value at the signal center $\hat{y}_i$. They proposed to propagate the confidence from the NCNN layer as:

$$\hat{c}_i = \frac{\langle \mathbf{a} | \mathbf{c} \rangle}{\langle \mathbf{1}_n | \mathbf{a} \rangle} \quad , \quad (5)$$

where the output confidence from one layer is the input confidence to the next layer.

3.3. Self-Supervised Input Confidence Estimation using NCNNs

The assumption of binary input confidences adopted by [7, 8] can be problematic in real datasets. An example is the KITTI-Depth dataset [32], where some of the input values do not match the groundtruth due to LiDAR projection errors (shown in Figure 4 top). In this case, a binary input confidence would lead to artifacts in the output as NCNNs are dependent on the input confidence as shown in the calculations of (4). This dependency of the outputs on the input confidences facilitates learning the confidences. The inclusion of the input confidences in the calculations of the output from each layer indicates that the loss of the network would constitute gradients with respect to these confidences. Therefore, we can employ these gradients to learn input confidences that minimize the loss function.

We propose to use an *input confidence estimation* network that receives the input data and produces an estimate for the input confidence that is fed to the first layer of the NCNN. This network is trained end-to-end with the NCNN and the error gradients from the NCNN are back-propagated to the confidence estimation network, allowing it to learn the input confidence that minimizes the overall prediction error. We use a compact UNet [29] for the confidence estimation network with a Softplus activation at the final layer that will produce valid confidence values in the interval $[0, \infty[$. The pipeline is illustrated in Figure 2 (upper part).

4. Probabilistic NCNNs

Figure (1b) shows an example of the output confidence from the last NCNN layer when we estimate the input confidences using our proposed approach from the previous section. The figure shows that the output confidences do not exhibit a proper uncertainty measure that is strongly correlated with the error.

To obtain proper uncertainties from NCNNs, we introduce a probabilistic version of NCNNs by deriving the connection between the normalized convolution and statistical least-squares approaches. Then, we utilize this connection to produce reliable uncertainties with probabilistic characteristics. Finally, we apply the proposed theory to NCNNs and we derive a loss function for training them to produce accurate uncertainties.

4.1. Connection between NCNN and Generalized Least-Squares

In ordinary least-squares (OLS) problems, constant variance is assumed for all observations of the signal. Generalized least-squares (GLS), on the other hand, offers more flexibility to handle individual variance per observation. The *weighted-least squares* problem in (2) can be viewed as a special case of the GLS, where observations are heteroskedastic with unequal noise levels.

Assume the image of the signal under the subspace $\mathbf{B}$ is defined as $\mathbf{y} = \mathbf{B}\mathbf{r} + \mathbf{e}$, where $\mathbf{e}$ is a random noise variable with zero mean and variance $\sigma^2\mathbf{V}$. This variance models the heteroscedastic uncertainty of the observations in the signal, where σ^2 is global for each signal, and $\mathbf{V}$ is a positive definite matrix describing the covariance between the observations. The GLS solution to this problem reads [1]:

$$\hat{\mathbf{r}}_{\text{GLS}} = (\mathbf{B}^*\mathbf{V}^{-1}\mathbf{B})^{-1}\mathbf{B}^*\mathbf{V}^{-1}\mathbf{y} \ . \tag{6}$$

When comparing the two solutions in (2) and (6), they are only equivalent if $\mathbf{V}^{-1}$ is diagonal, which leads to $\mathbf{V} = (\mathbf{W_a}\mathbf{W_c})^{-1}$. The diagonality of the covariance matrix indicates that different samples of the signal are independent and have different variances depending on the confidence and the applicability function.

We utilize the GLS solution $\hat{\mathbf{r}}_{\text{GLS}}$ to estimate the signal similar to (3) as $\hat{\mathbf{y}} = \mathbf{B}\hat{\mathbf{r}}_{\text{GLS}}$. The uncertainty of $\hat{\mathbf{y}}$ can be estimated as:

$$\begin{aligned} \text{cov}(\hat{\mathbf{y}}) &= \text{cov}(\mathbf{B}\hat{\mathbf{r}}_{\text{GLS}}) = \mathbf{B}\ \text{cov}(\hat{\mathbf{r}}_{\text{GLS}})\mathbf{B}^* \\ &= \sigma^2\mathbf{B}(\mathbf{B}^*\mathbf{V}^{-1}\mathbf{B})^{-1}\mathbf{B}^* \\ &= \sigma^2\mathbf{B}(\mathbf{B}^*\mathbf{W_a}\mathbf{W_c}\mathbf{B})^{-1}\mathbf{B}^* \ . \end{aligned} \tag{7}$$

Note that $\mathbf{W_a}$ and $\mathbf{W_c}$ are non-stochastic, where the former is estimated during NCNN training and the latter can be learned using our proposed approach in section 3.3. On the other hand, σ^2 is unknown and needs to be estimated.

4.2. Output Uncertainty for NCNNs

In case of NCNNs with the naïve basis $\mathbf{B} = \mathbf{1}_n$, the uncertainty measure in (7) simplifies to:

$$\begin{aligned} \text{cov}(\hat{\mathbf{y}}) &= \text{cov}(\mathbf{1}_n\hat{r}) = \sigma^2\mathbf{1}_n(\mathbf{1}_n^*\mathbf{W_a}\mathbf{W_c}\mathbf{1}_n)^{-1}\mathbf{1}_n^* \\ &= \mathbf{1}_n\frac{\sigma^2}{\langle\mathbf{a}|\mathbf{c}\rangle}\mathbf{1}_n^* \ . \end{aligned} \tag{8}$$

This indicates an equal uncertainty for the whole neighborhood, but since we are only interested in signal center $\hat{y}_i$, (8) reduces to:

$$\text{var}(\hat{y}_i) = \frac{\sigma_i^2}{\langle\mathbf{a}|\mathbf{c}\rangle} \ . \tag{9}$$

It is evident that the output confidence described in (5) disregards the stochastic noise variance σ_i^2. However, to obtain a proper uncertainty measure, this variance needs to be incorporated in the output confidence. We propose to estimate the noise variance σ_i^2 from the output confidence of the last NCNN layer by means of a noise variance estimation network as illustrated in Figure 2. To achieve this, we need a loss function that allows training the proposed framework.

4.3. The Loss Function for Probabilistic NCNNs

We consider each point y_i in the global signal $\mathcal{Y}$, where the neighborhood at this point is the local signal $\mathbf{y}$. This local signal can be represented under some basis as $\hat{\mathbf{y}} = \mathbf{B}\hat{\mathbf{r}}$, where the estimated coordinates $\hat{\mathbf{r}}$ are calculated from (6,2). We assume that the estimate of the signal follows a multivariate normal distribution $\hat{\mathbf{y}} \sim \mathcal{N}_m(\mathbf{B}\hat{\mathbf{r}}, \sigma^2\mathbf{B}(\mathbf{B}^*\mathbf{W_a}\mathbf{W_c}\mathbf{B})^{-1}\mathbf{B}^*)$ where the variance is defined in (7). In case of the naïve basis, we will have a univariate normal distribution $\hat{y}_i \sim \mathcal{N}(\hat{r}_i, \sigma_i^2/\langle\mathbf{a}|\mathbf{c}\rangle)$, where the variance is defined in (9). More formally, a NCNN outputs the mean $\hat{r}_i^L$ of the normal distribution around $\hat{y}_i$, and the scalar product $\langle\mathbf{a}|\mathbf{c}\rangle$ in the denominator of the variance. Yet, the noise variance σ^2 needs to be estimated to comply with the definition in (9).

We denote the variance term as $s_i = \sigma_i^2/\langle\mathbf{a}|\mathbf{c}\rangle$, where $\mathbf{a}$ and $\mathbf{c}$ are the applicability and the output confidence from the *last* NCNN layer. The least squares solution in (4) can be formulated as a maximum likelihood problem of a Gaussian error model for the last NCNN layer L:

$$l(\mathbf{w}) = \frac{1}{\sqrt{2\pi s_i}}\exp\left(-\frac{\parallel y_i - \hat{r}_i^L \parallel^2}{2s_i}\right) \ , \tag{10}$$

where $\mathbf{w}$ denotes the network parameters, and $\hat{r}_i^L$ is calculated based on (4). By taking log likelihood of (10) instead, we obtain:

$$L(\mathbf{w}) = -\frac{1}{2}\log(2\pi) - \frac{1}{2}\log(s_i) - \frac{\parallel y_i - \hat{r}_i^L \parallel^2}{2s_i} \ . \tag{11}$$

The first term is a constant and is ignored, and the cost function is defined as minimizing the negative log likelihood:

$$C(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N}\underbrace{\frac{\parallel y_i - \hat{r}_i^L \parallel^2}{s_i}}_{\text{Data term}} + \underbrace{\log(s_i)}_{\text{Regl. term}} , \tag{12}$$

where the scalar $1/2$ has been discarded. This cost function shares similarity with the aleatoric uncertainty loss proposed in [18]. The difference is that s_i in our case depicts an uncertainty measure that encodes observation noise variance and the output confidence from NCNN, while in [18], it is the variance of the noise. Note that this cost function can be derived using any error model from the exponential family, *e.g.* Laplace distribution as in [15]. Next, we show the architecture design that is used for training our proposed probabilistic approach.

4.4. Probabilistic NCNN Architecture

Given a dataset that contains undisturbed data $\mathcal{Y}$ as groundtruth and a disturbed version $\dot{\mathcal{Y}}$ as input, we aim to train a network that produces the clean data given the disturbed one. An illustration for our full pipeline is shown

in Figure 2. The first component estimates the input confidence from the disturbed input and feed both of them to the NCNN network. The output confidence from the last NCNN layer is fed to another compact UNet to estimate the noise parameter σ_i^2 and to produce s_i in (12). Finally, the prediction from the NCNN network and the estimated uncertainty s_i are fed to the loss.

Note that the noise variance estimation network takes only the output confidence from the NCNN as input, contrarily to existing approaches that estimate the uncertainty from the final prediction [12, 15]. This indicates that our confidences can efficiently encode the uncertainty information, which is also demonstrated in the experiments section.

5. Experiments

To demonstrate the capabilities of our proposed approach, we evaluate it on the KITTI-Depth dataset [32] for the task of *unguided* depth completion (no RGB guidance is used). We first compare against Bayesian Deep Learning approaches, *e.g.* MC-Dropout [9] and ensembling [20], in terms of prediction accuracy and the quality of the uncertainty measure. Then, we show comparison against the conventional non-statistical approaches. Afterwards, we perform an ablation study for different components of our pipeline and we experiment with an ensemble of our proposed network. Finally, we demonstrate the generalization capabilities of our approach by evaluating it on multi-path interference correction [11] and optical flow rectification. The source code is available on Github [1].

5.1. Experimental Setup

Our pipeline is illustrated in Figure 2 and more details are given in the supplementary materials. We evaluate three variations of our network: our network where only the input confidence estimation part that is trained using the L1 or the L2 norm (*NCNN-Conf*), our full network trained with the proposed loss in (12) (*pNCNN*), and our full network trained with a modified version of the loss in (12), where we apply an exponential function to s_i in the data term (*pNCNN-Exp*). This modification is to robustify our loss to outliers violating the presumed Gaussian error model for the data term. Training was performed using the Adam optimizer with an initial learning rate of 0.01 that is decayed with a factor of 10^{-1} every 3 epochs.

Evaluation Metrics We use the following two measures:

Prediction Error We use the error metrics from the KITTI-Depth [32] such as Mean Average Error (MAE), Root Mean Square Error (RMSE) and their inverses.

Quality of Uncertainty We use the sparsitification error plots and the area under sparsification error plots (AUSE) [15] as a measure for the quality of the uncertainty.

[1] https://github.com/abdo-eldesokey/pncnn

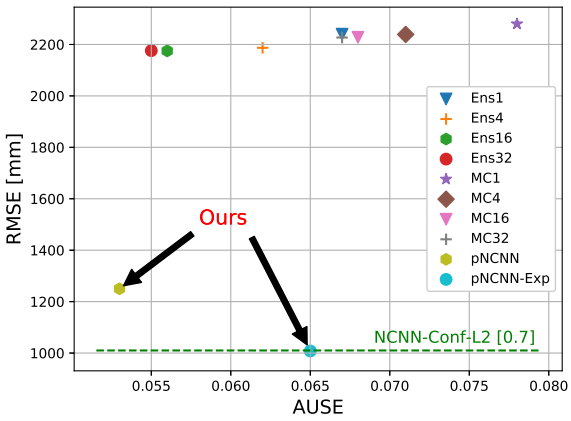

Figure 3. A comparison between statistical approaches in terms of RMSE and AUSE metrics where bottom-left is better. The two variations of our approach outperforms other methods w.r.t. RMSE and *pNCNN* trained with (12) produces the best uncertainty measure. Note that *NCNN-Conf-L2* only achieves AUSE of 0.7.

5.2. Results Compared to Statistical Methods

Gustafsson *et al.* [12] evaluated the MC-Dropout [9] and ensembling [20] by modifying the head of the Sparse-to-Dense (S2D) [24] network to output the parameters of a Gaussian distribution. They evaluated an ensemble of 1-32 instances of S2D with 26M parameters each an taking the mean of these instances for the final prediction. Note that their network utilizes both depth and RGB images, while our approach consist of a *single* network that is fully unguided and uses only depth data.

Figure 3 shows a two-metric comparison with respect to AUSE and RMSE. Our *NCNN-Conf* performs best in terms of RMSE, while it performs worst in terms of AUSE. On the other hand, our full network trained with the proposed loss, *pNCNN*, produces the best uncertainty measure with an AUSE of **0.053** outperforming an ensemble of 32 networks. Moreover, it achieves a significantly lower RMSE than MC-Dropout and ensembling. However, it performs inferior to *NCNN-Conf* in terms of RMSE with a moderate gap. The variation of our network that is trained with a modified loss, *pNCNN-Exp*, closes this gap and performs on-par with *NCNN-Conf* in terms of RMSE with a minor degradation of AUSE compared to *pNCNN*.

5.3. Results Compared to Non-Statistical Methods

We also compare our proposed approach against the non-statistical unguided approaches. Table 1 summarizes the results on the test set of the KITTI-Depth dataset. Our *NCNN-Conf-L1* outperforms all other methods on three out of four metrics when compared individually, except for *Spade*, where we are better on two metrics and on-par on one metric. Note the improvement of our approach over the standalone *NCNN*, where we achieve a performance boost of

	MAE [mm]	RMSE [mm]	iMAE [1/km]	iRMSE [1/km]	#P
SparseConv [32]	481.27	1601.33	1.78	4.94	25k
ADNN [4]	439.48	1325.37	3.19	59.39	1.7k
NCNN [7]	360.28	1268.22	1.52	4.67	0.5k
S2D [24]	288.64	954.36	1.35	3.21	26M
HMS-Net [14]	258.48	*937.48*	1.14	2.93	-
SDC [33]	249.11	**922.93**	1.07	2.80	2.5M
Spade [17]	*248.32*	1035.29	**0.98**	**2.60**	5.3M
NCNN-Conf-L1	**228.53**	988.57	*1.00*	*2.71*	330k
NCNN-Conf-L2	258.68	954.34	1.17	3.40	330k
pNCNN-Exp	251.77	960.05	1.05	3.37	670k

Table 1. Quantitative results on the *test* set of the KITTI-Depth for *unguided* depth completion. #P is the number of parameters.

$\sim 45\%$ by providing more accurate input confidences. Our probabilistic model trained using a Gaussian error model and a Laplace error model, *pNCNN-Exp* trained with the modified loss performs equally good to the *NCNN-Conf-L2*, but additionally providing proper output uncertainties.

5.4. Ablation Study

First we show the impact of each component of our proposed network on a qualitative example from the KITTI-Depth dataset. Figure 4 shows an example where the input measurements do not coincide with the groundtruth. The standard NCNN assigns 1-confidences to all measurements, which results in a corrupted prediction (first row). When we apply our input confidence estimation, the disturbed measurements are successfully identified and assigned zero confidence (second row). However, the output confidence is almost identical to the input confidence and shows no strong correlation with the accuracy. When we apply our full pipeline, the disturbed measurements are identified and the output uncertainty becomes highly correlated with the prediction error (third row).

Next, we show in Table 2 the impact of modifying different components of our pipeline. When the confidence estimation is discarded in *w/o conf-est* and binary input confidence is used, the RMSE is degraded, while the network still manages to achieve good AUSE. Similarly, when the noise variance estimation network is discarded in *w/o var-est*, the RMSE is severely degraded as the input confidence estimation network tries to make up for the absence of the variance estimation network. When the final prediction from the NCNN is fed along with the output confidence to the noise variance estimation network in *w depth-pred*, no improvement is gained in terms of AUSE. This demonstrates that our uncertainty measure efficiently encode the uncertainty information in the NCNN confidence stream

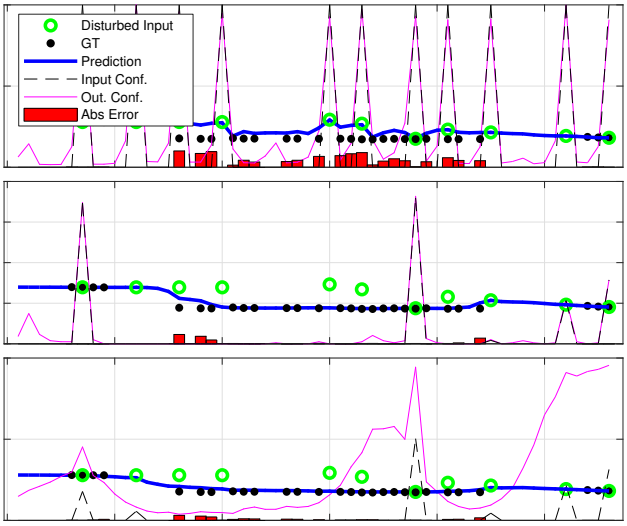

Figure 4. A qualitative example from the KITTI-Depth dataset showing the impact of each component of our proposed approach. First row is the standard *NCNN*, the second is *NCNN-Conf-L2*, and the third is *pNCNN*.

	RMSE	MAE	AUSE
pNCNN	1237.65	283.41	0.055
- w/o conf-est	1540.00	405.00	0.058
- w/o var-est	1703.50	604.10	0.123
- w depth-pred	1215.64	292.68	0.055
- w Laplace-loss	1272.32	248.26	0.089

Table 2. The results for the ablation study when trained on a subset of the training set evaluated on the selected validation set of the KITTI-Depth dataset.

without looking at the prediction. Finally, when we employ a Laplace error model for the loss in *w Laplace-loss*, *i.e.*, the L1 norm for residuals, the MAE improves, while AUSE is degraded since it is calculated based on the RMSE.

5.5. Ensemble of pNCNN

To examine whether our probabilistic approach can be extended to a fully Bayesian approach, we form an ensemble of four *pNCNN* network that were initialized randomly and trained on random subset of the KITTI-Depth dataset. We evaluate multiple fusion approaches which are summarized in Table 3. Fusion by selecting the most confident pixel from each network, *maxConf*, achieves the best results, outperforming taking the mean, which is commonly used. Taking a weighted mean using confidences, *wMean*, or a maximum likelihood estimation, *MLE*, also gives better results than the standard mean. This demonstrated the potential of using the proposed output confidences in more sophisticated fusion schemes.

	RMSE	MAE	**Fusion**	RMSE	MAE
Net-1	1337.5	290.5	Mean	1287.3	290.5
Net-2	1325.1	303.1	wMean	1261.3	285.9
Net-3	1315.1	296.9	maxConf	**1260.7**	283.8
Net-4	1321.1	288.3	MLE	1264.1	**282.4**

Table 3. Fusion schemes for an ensemble of *pNCNN* trained on a subset of the KITTI-Depth and evaluated on the selected validation set. *MLE* refers to Maximum Likelihood Estimation.

5.6. Mutli-Path Interference (MPI) Correction

To demonstrate the generalization capabilities of our approach on other kinds of noise, we evaluate it on depth data from a Time-of-Flight (ToF) camera, *i.e*. Kinect2, that suffers from MPI. We use the FLAT dataset [11] for this purpose which provides raw measurements for three different frequencies and phases. We use the libfreenect2 [35] to calculate the depth from the measurements and we compare against applying the bilateral filtering on the noisy depth.

Table 4 summarizes the results, where we outperform the Bilateral filtering with a significant margin in terms of RMSE error when evaluated both on noisy and clean data with no MPI. Bilateral filtering on the other hand performs worse than doing no processing as it assigns zeros to pixels close to edges. When edges are not considered for evaluation, bilateral filtering improves the results slightly, but is outperformed by our approach.

5.7. Sparse Optical Flow Rectification

We generate the input flow by applying the Lucas-Kanade method [23] to pairs of images from driving sequences. The groundtruth is produced by geometrical verification over several frames under a multiple rigid body assumption [27]. Figure 5 shows an example for rectifying the corrupted measurement and densifying the flow field. More results are given in the supplementary materials.

5.8. What happens if the input is undisturbed?

An essential question is how our confidence estimation network will perform if the input data is not disturbed? To answer this question, we train our network *NCNN-Conf* and *pNCNN* on the NYU dataset [26], where the input is sampled from the groundtruth depth. We use 1000 depth points sampled uniformly with a sparsity level of 0.6%. Figure 6 and Table 7 show that both our methods surprisingly improves the results compared to the standalone NCNN [7]. This is a result of allowing the confidence estimation network to assign proper confidences to points based on their proximity to edges similar to non-linear filtering. This leads to sharper edges and better reconstruction of objects.

RMSE [mm]	Ours	Biateral	No-Proc
No-MPI	**231**	444	415
MPI	**283**	429	449
No-MPI-Masked	**175**	263	288
MPI-Masked	**205**	282	299

Table 4. The RMSE error in millimeters for Multi-Path Interference (MPI) correction on the FLAT dataset [11]. *No-Proc* refers to evaluating the depth without any processing. The masked version disregards edges from the evaluation.

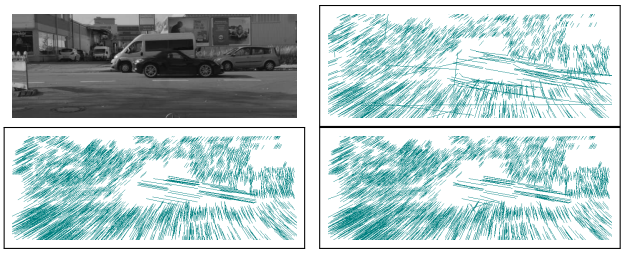

Figure 5. Qualitative example for optical flow outliers rejection. In right-bottom order, RGB frame, raw flow input, groundtruth flow, and estimated flow.

Figure 6. A qualitative example from the NYU dataset [26]. Top-to-bottom: groundtruth, NCNN [7], NCNN-Conf.

	RMSE	MAE
NCNN [7]	0.165	0.07
NCNN-Conf	**0.135**	**0.05**
pNCNN	0.144	0.06

Figure 7. Quantitative results on the NYU dataset [26] in meters.

6. Conclusion

We proposed a self-supervised approach for estimating the input confidence for sparse data based on the NCNNs. We also introduced a probabilistic version of NCNNs that enable the to output meaningful uncertainty measures. Experiments on the KITTI dataset for unguided depth completion showed that our small network with 670k parameters achieves state-of-the-art results in terms of prediction accuracy and it provides an accurate uncertainty measure. When compared against the existing probabilistic method for dense problems, our proposed approach outperforms all of them in terms of the prediction accuracy, the quality of the uncertainty measure, and the computational efficiency. Moreover, we showed that our approach can be applied to other sparse problems as well. These results demonstrate the gains from adhering to the signal/uncertainty philosophy compared to conventional black-box models.

Acknowledgments: This work was supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) and Swedish Research Council grant 2018-04673.

References

[1] Alexander C Aitken. Iv.on least squares and linear combination of observations. *Proceedings of the Royal Society of Edinburgh*, 55:42–48, 1936.

[2] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. *arXiv preprint arXiv:1904.11492*, 2019.

[3] Yun Chen, Bin Yang, Ming Liang, and Raquel Urtasun. Learning joint 2d-3d representations for depth completion. In *ICCV*, 2019.

[4] Nathaniel Chodosh, Chaoyang Wang, and Simon Lucey. Deep Convolutional Compressed Sensing for LiDAR Depth Completion. mar 2018.

[5] Jiwoong Choi, Dayoung Chun, Hyun Kim, and Hyuk-Jae Lee. Gaussian yolov3: An accurate and fast object detector using localization uncertainty for autonomous driving. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[6] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Atom: Accurate tracking by overlap maximization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4660–4669, 2019.

[7] Abdelrahman Eldesokey, Michael Felsberg, and Fahad Shahbaz Khan. Propagating confidences through cnns for sparse data regression. In *The British Machine Vision Conference (BMVC), Northumbria University, Newcastle upon Tyne, England, UK, 3-6 September, 2018*, 2018.

[8] Abdelrahman Eldesokey, Michael Felsberg, and Fahad Shahbaz Khan. Confidence propagation through cnns for guided sparse depth regression. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[9] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

[10] Jochen Gast and Stefan Roth. Lightweight probabilistic deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3369–3378, 2018.

[11] Qi Guo, Iuri Frosio, Orazio Gallo, Todd Zickler, and Jan Kautz. Tackling 3d tof artifacts through learning and the flat dataset. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 368–383, 2018.

[12] Fredrik K Gustafsson, Martin Danelljan, and Thomas B Schön. Evaluating scalable bayesian deep learning methods for robust computer vision. *arXiv preprint arXiv:1906.01620*, 2019.

[13] Lifeng Huang, Chengying Gao, Yuyin Zhou, Changqing Zou, Cihang Xie, Alan Yuille, and Ning Liu. Upc: Learning universal physical camouflage attacks on object detectors, 2019.

[14] Z. Huang, J. Fan, S. Yi, X. Wang, and H. Li. HMS-Net: Hierarchical Multi-scale Sparsity-invariant Network for Sparse Depth Completion. *ArXiv e-prints*, Aug. 2018.

[15] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 652–667, 2018.

[16] Maximilian Jaritz, Raoul De Charette, Emilie Wirbel, Xavier Perrotton, and Fawzi Nashashibi. Sparse and dense data with cnns: Depth completion and semantic segmentation. In *2018 International Conference on 3D Vision (3DV)*, pages 52–60. IEEE, 2018.

[17] Maximilian Jaritz, Raoul de Charette, Emilie Wirbel, Xavier Perrotton, and Fawzi Nashashibi. Sparse and dense data with cnns: Depth completion and semantic segmentation. *arXiv preprint arXiv:1808.00769*, 2018.

[18] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in neural information processing systems*, pages 5574–5584, 2017.

[19] Hans Knutsson and Carl-Fredrik Westin. Normalized and differential convolution. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 515–523. IEEE, 1993.

[20] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pages 6402–6413, 2017.

[21] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 734–750, 2018.

[22] Yanghao Li, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Scale-aware trident networks for object detection. *arXiv preprint arXiv:1901.01892*, 2019.

[23] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. 1981.

[24] Fangchang Ma, Guilherme Venturelli Cavalheiro, and Sertac Karaman. Self-supervised sparse-to-dense: self-supervised depth completion from lidar and monocular camera. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3288–3295. IEEE, 2019.

[25] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 181–196, 2018.

[26] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[27] Mikael Persson, Tommaso Piccini, Michael Felsberg, and Rudolf Mester. Robust stereo visual odometry from monocular techniques. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 686–691. IEEE, 2015.

[28] Jiaxiong Qiu, Zhaopeng Cui, Yinda Zhang, Xingdi Zhang, Shuaicheng Liu, Bing Zeng, and Marc Pollefeys. Deeplidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3313–3322, 2019.

[29] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[30] Towaki Takikawa, David Acuna, Varun Jampani, and Sanja Fidler. Gated-scnn: Gated shape cnns for semantic segmentation. *arXiv preprint arXiv:1907.05740*, 2019.

[31] Hugo Touvron, Andrea Vedaldi, Matthijs Douze, and Hervé Jégou. Fixing the train-test resolution discrepancy. *arXiv preprint arXiv:1906.06423*, 2019.

[32] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *2017 International Conference on 3D Vision (3DV)*, pages 11–20. IEEE, 2017.

[33] Wouter Van Gansbeke, Davy Neven, Bert De Brabandere, and Luc Van Gool. Sparse and noisy lidar completion with rgb guidance and uncertainty. In *2019 16th International Conference on Machine Vision Applications (MVA)*, pages 1–6. IEEE, 2019.

[34] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1328–1338, 2019.

[35] Lingzhu Xiang, Florian Echtler, Christian Kerl, Thiemo Wiedemeyer, Lars, hanyazou, Ryan Gordon, Francisco Facioni, laborer2008, Rich Wareham, and et al. libfreenect2: Release 0.2. Apr 2016.

[36] Yan Xu, Xinge Zhu, Jianping Shi, Guofeng Zhang, Hujun Bao, and Hongsheng Li. Depth completion from sparse lidar data with depth-normal constraints. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[37] Yi Zhu, Karan Sapra, Fitsum A Reda, Kevin J Shih, Shawn Newsam, Andrew Tao, and Bryan Catanzaro. Improving semantic segmentation via video propagation and label relaxation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8856–8865, 2019.