Learning Steering Kernels for Guided Depth Completion

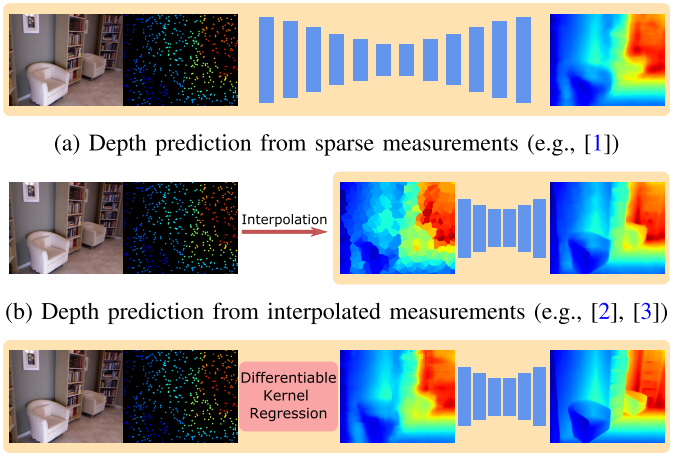
Lina Liu, Yiyi Liao, Yue Wang, Andreas Geiger, and Yong Liu

***Abstract*— This paper addresses the guided depth completion task in which the goal is to predict a dense depth map given a guidance RGB image and sparse depth measurements. Recent advances on this problem nurture hopes that one day we can acquire accurate and dense depth at a very low cost. A major challenge of guided depth completion is to effectively make use of extremely sparse measurements, e.g., measurements covering less than 1% of the image pixels. In this paper, we propose a fully differentiable model that avoids convolving on sparse tensors by jointly learning depth interpolation and refinement. More specifically, we propose a differentiable kernel regression layer that interpolates the sparse depth measurements via learned kernels. We further refine the interpolated depth map using a residual depth refinement layer which leads to improved performance compared to learning absolute depth prediction using a vanilla network. We provide experimental evidence that our differentiable kernel regression layer not only enables end-to-end training from very sparse measurements using standard convolutional network architectures, but also leads to better depth interpolation results compared to existing heuristically motivated methods. We demonstrate that our method outperforms many state-of-the-art guided depth completion techniques on both NYUv2 and KITTI. We further show the generalization ability of our method with respect to the density and spatial statistics of the sparse depth measurements.**

***Index Terms*— Depth completion, depth interpolation, kernel regression, steering kernels.**

I. Introduction

DEPTH perception plays an important role for humans, and is also a valuable cue in computer vision and robotics [4]–[6]. Acquiring dense and accurate depth at a low cost is particularly important for applications such as autonomous driving and virtual reality. While existing depth sensors have contributed significantly to these applications, the goal of obtaining accurate and dense depth at low cost is still not fulfilled. For example, 3D LiDAR provides accurate but sparse measurements and existing scanners are typically expensive. Though structured light sensors are more affordable, they often have a smaller field of view and are constrained to indoor environment. More recently, learning-based monocular depth estimation methods have attracted attention due to their low cost, as only a monocular RGB image is required to estimate the depth. However, estimating depth from monocular images is a highly ill-posed problem.



(a) Depth prediction from sparse measurements (e.g., [1])

(b) Depth prediction from interpolated measurements (e.g., [2], [3])

(c) End-to-end depth interpolation and refinement (ours)

Fig. 1. Illustration comparing different guided depth completion approaches. The task is to predict a dense depth map given a guidance RGB image and sparse depth measurements. Learned components are highlighted in yellow.

To acquire dense and accurate depth using affordable sensors, we aim for a learning-based solution of guided depth completion. More specifically, we predict the full resolution depth map based on a guidance RGB image and a set of very sparse depth measurements (e.g., less than 1% of the image pixels), which can be obtained from a low-cost LiDAR with a small number of laser beams or visual SLAM [7]. Compared to predicting depth from a monocular RGB image alone, our method is able to resolve spatial ambiguities using metric cues from the sparse depth measurements.

However, estimating dense depth maps from sparse measurements is not trivial. Focusing on this problem, there are two dominant lines of research as illustrated in Fig. 1. In the first class (Fig. 1a), the sparse depth measurements are represented as a sparse depth map. The pixels without valid depth are set to a constant number, e.g., 0 [1]. The resulting

Manuscript received March 26, 2020; revised December 28, 2020; accepted January 19, 2021. Date of publication February 4, 2021; date of current version February 12, 2021. The work of Lina Liu and Yong Liu was supported by the National Natural Science Foundation of China under Grant 61836015. The work of Yiyi Liao was supported by the German Federal Ministry of Education and Research (BMBF): Tübingen AI Center, under Grant FKZ: 01IS18039A. The work of Andreas Geiger was supported by the ERC Starting Grant LEGO-3D (850533). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Christophoros Nikou. *(Corresponding authors: Yiyi Liao; Yong Liu.)*

Lina Liu, Yue Wang, and Yong Liu are with the State Key Laboratory of Industrial Control Technology and Institute of Cyber-Systems and Control, Zhejiang University, Zhejiang 310027, China (e-mail: yongliu@iipc.zju.edu.cn).

Yiyi Liao and Andreas Geiger are with the Autonomous Vision Group, Max Planck Institute for Intelligent Systems and University of Tübingen, 72076 Tübingen, Germany (e-mail: yiyi.liao@tue.mpg.de).

Digital Object Identifier 10.1109/TIP.2021.3055629

sparse depth map is concatenated with the guidance image and fed into a neural network to predict the dense depth map. The advantage of this method is that the model can be trained in an end-to-end manner. However, in this case the network needs to resolve the ambiguity between valid and invalid depth measurements and vanilla convolutional networks with small convolution kernels are not suited for very sparse inputs as considered in this work [8]. Another line of works avoids this problem by addressing the guided depth completion task in two stages (Fig. 1b). The sparse depth measurements are first interpolated into an intermediate dense depth map, which is then utilized for depth refinement in a second stage [2], [3]. Though each pixel in the interpolated depth image is valid in this case, existing techniques for constructing the interpolated depth map are hand-crafted and thus cannot be trained from data.

In this paper, we propose a solution to guided depth completion by combining advantages of intermediate dense depth representations and end-to-end training as illustrated in Fig. 1c. Our key idea is to replace the hand-crafted interpolation with a differentiable kernel regression layer. Our learned kernel regression layer transforms the sparse measurements into a dense representation which is suitable for depth refinement using a standard convolutional network. More specifically, we first estimate the interpolated depth map using kernel regression, where the kernel shapes are learned via a neural network conditioning on the guidance image. Next, we transform the absolute depth estimation task into residual depth estimation by refining the interpolated depth map using a refinement network. Our contributions are summarized as follows:

- We propose a differentiable kernel regression layer which learns the kernel shapes conditioned on the guidance RGB image. We show that our learned kernels reduce the estimation bias comparing to heuristically designed kernels.
- We integrate our differentiable kernel regression layer into guided depth completion that enables end-to-end depth estimation while avoiding convolving directly on the sparse measurements. With the dense interpolated map obtained from our kernel regression layer, we are able to redefine the task of estimating the absolute depth map as estimating the residual between the real and the interpolated depth map.
- Our experiments demonstrate that the proposed method achieves superior performance compared to existing guided depth completion techniques. We further show that our model generalizes well wrt.varying number of points and different depth observations patterns.

The remainder of the paper is organized as follows: Section II gives a review of related work. We introduce our method in Section III and present the experimental results in Section IV. Finally, we conclude the paper in Section V.

II. Related Work

A. Monocular Depth Estimation

Many works have considered the task of estimating a depth map from a monocular image. Conventional methods tackle the problem based on hand-crafted features and graphical models [9], [10]. More recently, supervised learning methods achieve state-of-the-art performance on monocular depth estimation [11]–[18]. Unsupervised methods also demonstrate promising results on this task [19], [20].

Despite recent advances on monocular depth estimation, the task is inherently ill-posed, e.g., the global scale of the depth map is ambiguous [11]. In this work, we consider the task of guided depth completion in which scale ambiguities can be resolved using a set of sparse measurements.

B. Depth Upsampling

Depth upsampling methods aim at increasing the spatial resolution of a given depth map. While a few works directly upsample the depth map without any guidance information [21], [22], several approaches demonstrate that it is beneficial to exploit a color or intensity image at the target resolution as guidance [23]–[32]. Most approaches falling into this category either use a guided bilateral filter [23]–[26] or global energy minimization [27]–[32] for this purpose. More recently, deep learning methods became popular, demonstrating state-of-the-art performance on the depth upsampling task [33]–[35].

In contrast to depth upsampling methods that take a low-dimensional depth map defined on a regular pixel grid as input, we consider the depth completion task in which the depth observations are incomplete, sparse and irregularly distributed.

C. Depth Completion

Various methods have been proposed to *fill holes* in depth images [4], [36]–[41]. These methods typically assume that the depth is densely observed for most image areas but incomplete at specific regions, e.g., at object boundaries and in reflective areas where depth sensors struggle to reconstruct depth. Therefore, they are not directly applicable to sparse input observations which we consider in this paper.

Many works have attempted to estimate dense depth maps given only very sparse and irregular depth measurements. Non-guided methods [8], [42] provide compelling results given observations at a moderate sparsity level, e.g., 10% of pixels carry depth information [8]. However, their performance is limited when only very few depth measurements are available, e.g., less than 1% of the pixels being observed. Incorporating a guidance image can substantially boost depth completion performance in both moderate [41]–[45] and extremely sparse settings [1]–[3], [42], [46]–[48]. In this paper, we focus on the second, more challenging setting.

In this setting, a major challenge is to effectively incorporate the extremely sparse measurements into a deep neural network. A straightforward option is to represent the measurements as a sparse depth map where the depth value in unknown regions is set to 0, and then feed it into the deep networks along with the RGB image [1], [46], [47] or an inferred normal map [44]. However, as pointed out in [8], standard convolutional neural networks degenerate when applied directly on sparse inputs given a limited kernel size. Though this problem can be resolved by specifically designing the network architecture

for sparse inputs [8], [42], [49], we seek an alternative by differentiably transforming the sparse input into a dense representation that can be fed into standard convolutional neural networks. Following this idea, our earlier work [2] extrapolates a horizontal 2D laser scanline in gravity direction and takes this extrapolated depth map as input to a deep neural network. Similarly, Chen et al. [3] and Shivakumar et al. [48] interpolate the sparse depth map into a dense depth map as input to the convolutional neural network. All these methods pre-process the sparse depth measurements heuristically and are not trained end-to-end.

Instead of constructing a dense depth map using hand-crafted interpolation techniques, in this paper, we propose a differentiable kernel regression layer which allows us to exploit the image distribution for depth interpolation. Moreover, our model learns to predict the optimal kernel parameters conditioned on the input image end-to-end from data.

D. Kernel Regression

Kernel regression is a well-established regression method [50], [51]. Conventionally, the same kernel is applied over the entire image, i.e., the kernel is data and location independent. Takeda et al. [52] propose a data-adaptive kernel regression method that steers the kernel based on an extra input, e.g., an RGB image. While in [52] the kernel shapes are heuristically designed, we learn the image-conditional kernel shape end-to-end from data. We experimentally demonstrate that our learned kernels are able to reduce the estimation bias compared to heuristically designed ones.

A few metric learning methods also learn kernel shapes from data for kernel regression [53], [54]. Weinberger & Tesauro [53] optimize the kernel shape by minimizing a regression loss. Noh et al. [54] derive an analytical solution for estimating kernel shapes from Gaussian-distributed data. Note that none of these approaches uses a deep neural network as encoder for estimating the kernel shapes. In contrast, we exploit a deep neural network to directly infer the kernel shape at each pixel from our guidance image, effectively amortizing the inference process [55].

III. Method

Our goal is to estimate a dense depth map given sparse depth measurements and a guidance RGB image as input. Existing works for guided depth completion either directly infer a dense depth map from sparse measurements or base their predictions on intermediate dense depth maps that are obtained using hand-crafted interpolation techniques (e.g., nearest neighbor). In this work, we instead propose to split the guided depth completion problem into two differentiable end-to-end trainable stages as illustrated in Fig. 2: We first predict a dense interpolated depth map $\tilde{\mathbf{D}}$ from sparse depth measurements $\mathcal{D}$ using kernel regression where the kernel parameters $\mathbf{\Phi} = \{\mathbf{\Gamma}, \mathbf{\Theta}, \mathbf{\Sigma}\}$ are estimated based on the guidance image $\mathbf{I}$. Here, $\mathcal{D} = \{(\mathbf{x}_1, d_1), \cdots, (\mathbf{x}_P, d_P)\}$ denotes the set of sparse depth measurements where $\mathbf{x}_i \in \mathbb{R}^2$ is the pixel location and $d_i \in \mathbb{R}^+$ the depth value. In a second step, we refine $\tilde{\mathbf{D}}$ into the final depth map $\hat{\mathbf{D}}$ using a deep residual network. Our method avoids applying convolutions directly on extremely sparse inputs while being end-to-end trainable.

In this section, we first give a formal introduction to kernel regression and demonstrate that existing works for hand-crafted depth interpolation [2], [3] are special cases of this kernel regression framework. Next, we present our differentiable kernel regression module which uses a neural network for predicting the kernel parameters from the guidance image instead of heuristically designing the kernels. Finally, we introduce our residual estimation network, as well as the loss functions and training details.

A. Kernel Regression

Kernel regression is a non-parametric regression technique which uses a kernel as weighting function. In the context of depth completion, we aim at developing a regression model $\tilde{\mathbf{D}}(\mathbf{x})$ that estimates a depth value at an arbitrary pixel $\mathbf{x} \in \mathbb{R}^2$ from a set of sparse measurements $\mathcal{D}$. Following [52], the kernel regression model at a given pixel $\mathbf{x}$ can be formulated as

$$\tilde{\mathbf{D}}(\mathbf{x}) = \frac{\sum_{i=1\dots P} \mathbf{K_H}(\mathbf{x}_i - \mathbf{x}) d_i}{\sum_{i=1\dots P} \mathbf{K_H}(\mathbf{x}_i - \mathbf{x})} \tag{1}$$

which represents the predicted depth at $\mathbf{x}$ as a linear combination of known measurements d_i weighted by the kernel function $\mathbf{K_H}(\mathbf{x}_i - \mathbf{x})$. The kernel shape is crucial for the performance of the regression model. In practice, they are often heuristically designed and can be categorized into data-independent and data-dependent kernels.

1) Data-Independent Kernel: A data-independent kernel determines the weight between $\mathbf{x}_i$ and $\mathbf{x}$ by the relative distance $\mathbf{x}_i - \mathbf{x}$ independent of a guidance image $\mathbf{I}$. In this case, the kernel shape at any pixel is constant across the image domain. A common kernel choice is the Gaussian kernel:

$$\mathbf{K_H}(\mathbf{x}_i - \mathbf{x}) = \frac{1}{2\pi\sqrt{\det(\mathbf{H}^2)}} \exp\left\{-\frac{(\mathbf{x}_i - \mathbf{x})^T \mathbf{H}^{-2} (\mathbf{x}_i - \mathbf{x})}{2}\right\} \tag{2}$$

Here, $\mathbf{H} \in \mathbb{R}^{2\times 2}$ is a constant smoothing matrix which does not depend on the guidance image $\mathbf{I}$.

We remark that some of existing hand-crafted intermediate depth maps [2], [3] can be view as interpolated using kernel regression with data-independent kernels. In our previous work [2], we expand a single horizontal laser scanline in vertical direction to generate a dense depth map. We can achieve a similar effect using kernel regression by constructing a Gaussian kernel that is extremely elongated in vertical direction. In [3], an intermediate dense depth map is constructed using nearest neighbor interpolation. This operation is equivalent to applying a hard assignment in Eq. (1), which considers only the depth value d_i with the largest $\mathbf{K_H}(\mathbf{x}_i - \mathbf{x})$ for estimating $\tilde{\mathbf{D}}(\mathbf{x})$. The interpolated depth map of both techniques [2], [3] is constructed without considering a guidance image $\mathbf{I}$.

2) Data-Dependent Kernel: Takeda et al. [52] propose a non-constant *steering* kernel which depends on a guidance image $\mathbf{I}$. More specifically, they use the image gradient $\nabla\mathbf{I}$

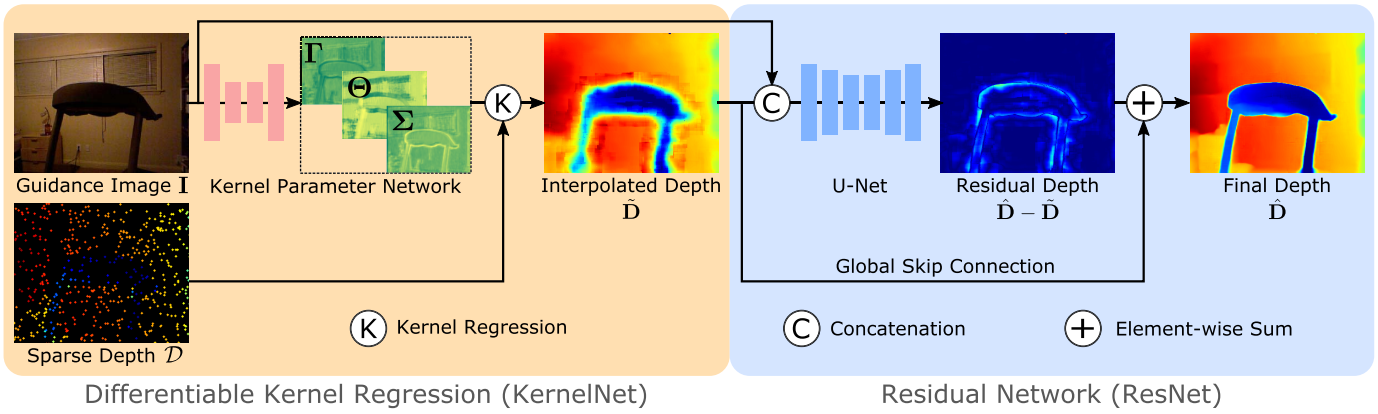


Fig. 2. **Method Overview.** Our differentiable kernel regression module (orange) predicts a dense interpolated depth map $\tilde{\mathbf{D}}$ from the guidance image $\mathbf{I}$ and the sparse measurements $\mathcal{D}$ (visually enhanced). The kernel parameters $\mathbf{\Phi} = \{\mathbf{\Gamma}, \mathbf{\Theta}, \mathbf{\Sigma}\}$ are learned from $\mathbf{I}$, and the interpolated depth map $\tilde{\mathbf{D}}$ is estimated using kernel regression given $\mathbf{\Phi}$ and $\mathcal{D}$. Given the interpolated depth map $\tilde{\mathbf{D}}$, our residual network (blue) predicts the final depth map $\hat{\mathbf{D}}$ conditioned on the guidance image $\mathbf{I}$. Instead of directly predicting $\hat{\mathbf{D}}$ using a deep neural network, we learn the residual depth $\hat{\mathbf{D}} - \tilde{\mathbf{D}}$ by adding a global skip connection. Our model avoids directly applying convolutions on the sparse measurements $\mathcal{D}$ while being end-to-end trainable.

to adaptively "steer" the kernel, resulting in elongated kernels spread along the image edges

$$\mathbf{K}_{\mathbf{H}i}(\mathbf{x}_i - \mathbf{x}) = \frac{1}{2\pi\sqrt{\det(\mathbf{H}_i^2)}} \exp\left\{-\frac{(\mathbf{x}_i - \mathbf{x})^T \mathbf{H}_i^{-2} (\mathbf{x}_i - \mathbf{x})}{2}\right\} \tag{3}$$

where $\mathbf{H}_i$ denotes the smoothing matrix which depends on the local content of guidance image $\mathbf{I}$ at pixel $\mathbf{x}_i$.

While $\mathbf{H}_i$ is constructed adaptively with respect to image features in [52], it is still heuristically designed and may not be optimal to reduce the estimation bias between the interpolated depth map $\tilde{\mathbf{D}}$ and the ground truth $\mathbf{D}$.

B. Differentiable Kernel Regression

To overcome the limitation of heuristic steering kernels, we propose a differentiable kernel regression layer that allows for learning the smoothing matrix $\mathbf{H}_i$ fully driven by the raw data. In contrast to the hand-crafted depth interpolation based on data-independent kernels, our learned steering kernels allow for adaptively integrating additional information of the guidance image $\mathbf{I}$. More importantly, we integrate the differentiable kernel regression layer into the guided depth completion framework, allowing for jointly optimizing depth interpolation and depth refinement end-to-end.

Specifically, we learn the kernel shapes from $\mathbf{I}$ to directly minimize the estimation bias between the interpolated depth map $\tilde{\mathbf{D}}$ and the ground truth $\mathbf{D}$. Note that our kernel shapes are learned without direct supervision, guided only by the estimation bias, i.e., we do not require ground truth for the kernel shapes. More formally, we follow the parametrization of [52] and represent $\mathbf{H}_i$ as

$$\mathbf{H}_i = h\mu_i \mathbf{C}_i^{-\frac{1}{2}} \tag{4}$$

where h is the global smoothing parameter and μ_i is a scalar that captures the local density of the depth samples. $\mathbf{C}_i$ is a covariance matrix that is specified using three parameters (i.e., $\boldsymbol{\gamma}_i$, $\boldsymbol{\theta}_i$ and $\boldsymbol{\sigma}_i$) which control the scaling, rotation and elongation of $\mathbf{C}_i$ respectively:

$$\begin{aligned} \mathbf{C}_i &= \boldsymbol{\gamma}_i \mathbf{U}_{\theta_i} \mathbf{\Lambda}_i \mathbf{U}_{\theta_i}^{\mathrm{T}} \\ \mathbf{U}_{\theta_i} &= \begin{bmatrix} \cos\theta_i & \sin\theta_i \\ -\sin\theta_i & \cos\theta_i \end{bmatrix} \\ \mathbf{\Lambda}_i &= \begin{bmatrix} \sigma_i & 0 \\ 0 & \sigma_i^{-1} \end{bmatrix} \end{aligned} \tag{5}$$

Intuitively, a circular kernel with isometric weighting is firstly elongated by the elongation matrix $\mathbf{\Lambda}_i$ whose axes lengths are scaled by σ_i. Next, the elongated kernel is rotated by the 2D rotation matrix $\mathbf{U}_{\theta_i}$. Finally, the kernel is scaled by $\boldsymbol{\gamma}_i$.

In our differentiable kernel regression layer, we propose to predict the kernel parameters $\mathbf{\Phi} = \{\mathbf{\Gamma}, \mathbf{\Theta}, \mathbf{\Sigma}\}$ from the guidance image $\mathbf{I}$ using a *kernel parameter network*, where $\gamma_i = \mathbf{\Gamma}(\mathbf{x}_i)$, $\theta_i = \mathbf{\Theta}(\mathbf{x}_i)$ and $\sigma_i = \mathbf{\Sigma}(\mathbf{x}_i)$. This is illustrated in the orange block of Fig. 2. In contrast to [52], we fix h and μ_i in our formulation as these parameters are redundant wrt. γ_i and thus mainly affect the initialization. We set $h = 5.0$ and $\mu_i = 1.0$ empirically where a relatively large h is used to compensate for the sparse observations. Note that each parameter map $(\mathbf{\Gamma}, \mathbf{\Theta}, \mathbf{\Sigma})$ has the size of the guidance image $\mathbf{I}$ and a data-dependent covariance matrix $\mathbf{C}_i$ is specified at every pixel location $\mathbf{x}_i$, determining the steering kernel $\mathbf{K}_{\mathbf{H}_i}(\mathbf{x}_i - \mathbf{x})$ in Eq. (3). Given the kernels centered at all sparse depth measurements $\mathcal{D}$, we then estimate the dense interpolated depth map $\tilde{\mathbf{D}}$ using kernel regression in Eq. (1). The kernel parameter network and the kernel regression operation are both differentiable and thus allow for end-to-end training.

We use a shallow version of [56] as our backbone for learning the kernel parameters. Note that we only use the guidance image $\mathbf{I}$ as input to the kernel parameter network, thus avoiding convolutions directly on the sparse depth measurements $\mathcal{D}$. Moreover, our kernel parameter network is independent of the sparse depth measurements. As evidenced by our experiments, this leads to high generalization performance wrt.the number of depth measurements as well as the (sparse) observation pattern. While only pixels with valid depth measurements

provide gradients to update the kernel parameter network, our approach is able to learn smooth kernel parameter maps as illustrated in Fig. 2 and Fig. 6.

We will refer to the entire kernel regression network (illustrated in orange in Fig. 2) as *KernelNet* in the following.

C. Residual Network

As illustrated in the blue block in Fig. 2, we refine the interpolated depth to obtain the final depth prediction. The dense interpolated depth map allows us to use standard convolutional networks to estimate the final depth map. We use a standard U-Net [56] which aggregates local and global feature information as our backbone.

As the output of the *KernelNet* is a dense depth map, the refinement step can be effectively implemented by estimating the residual depth $\hat{\mathbf{D}}-\tilde{\mathbf{D}}$ between the final depth map $\hat{\mathbf{D}}$ and the interpolated depth map $\tilde{\mathbf{D}}$. The residual depth estimation is implemented by introducing a *global skip connection* as illustrated in Fig. 2. Note that learning the residual mapping [57] via the global skip connection ensures a performance lower bound on the final depth, i.e., the final depth should perform equal or better than the interpolated depth. Another intuition of the residual formulation is normalization: while the absolute depth may differ in range across images, the residual depth is expected to follow a zero-mean distribution which helps during training [58].

We will refer to the entire residual network (illustrated in blue in Fig. 2) as *ResNet* in the following.

D. Loss Function

We define our loss function jointly on the interpolated depth map $\tilde{\mathbf{D}}$ and the final depth map $\hat{\mathbf{D}}$. For the interpolated depth map, the depth at each pixel is estimated using kernel regression as defined in Eq. (1). For the final depth map, we adopt a probabilistic estimation method by representing the depth value as the expectation over a predicted depth distribution at each pixel. Specifically, we estimate a discrete random variable at every pixel location $\mathbf{x}$, represented as a set of possible values $\bar{d}_1, \bar{d}_2, \ldots, \bar{d}_M$ occurring with probabilities $P(\mathbf{x}) = \{p_1(\mathbf{x}), p_2(\mathbf{x}), \ldots, p_M(\mathbf{x})\}$ at pixel $\mathbf{x}$, where M is the number of the discrete bins. The predicted depth at pixel location $\mathbf{x}$ is calculated as the expected depth value wrt. $P(\mathbf{x})$:

$$\hat{\mathbf{D}}(\mathbf{x}) = \sum_{k=1}^{M} \bar{d}_k \; p_k(\mathbf{x}) \tag{6}$$

In our implementation, $\bar{d}_1, \bar{d}_2, \ldots, \bar{d}_M$ are fixed and obtained by uniformly sampling on a provided range interval, while $P(\mathbf{x})$ is estimated using a softmax layer.

Given the data representation, we formulate our loss for training the full model as follows

$$\mathcal{L} = \sum_{\mathbf{x}} w_1 \mathcal{L}_{inter} + w_2 \mathcal{L}_{dirac} + w_3 \mathcal{L}_{mean} + w_4 \mathcal{L}_{grad} \tag{7}$$

with constant hyperparameters $w_1, \ldots, w_4$. The sum notation denotes the aggregation of the loss over all pixels. Note that we have dropped the dependency of the loss functions on the pixel location $\mathbf{x}$ for clarity. We now explain each term in detail.

1) Loss on Interpolated Depth: We minimize the ℓ_1 distance between the interpolated depth and the ground truth depth:

$$\mathcal{L}_{inter}(\mathbf{x}) = |\mathbf{D}(\mathbf{x}) - \tilde{\mathbf{D}}(\mathbf{x})| \tag{8}$$

This loss enables learning the kernel shape directly from data to reduce the estimation bias.

2) Loss on Final Depth Distribution: For the discrete random variable at every pixel location $\mathbf{x}$, we do not have access to the ground truth probability distribution. A reasonable assumption is a Dirac delta distribution based on which bin the ground truth depth falls into after discretization. Thus, a standard cross-entropy loss can be applied

$$\mathcal{L}_{dirac}(\mathbf{x}) = \sum_{k=1}^{M} \delta([\mathbf{D}(\mathbf{x})] - k) \log(p_k) \tag{9}$$

where $[\mathbf{D}(\mathbf{x})]$ returns the index of the bin $\mathbf{D}(\mathbf{x})$ belongs to. Note that existing works point out that the cross-entropy loss defined on discrete depth values outperforms naïve depth value regression [16].

3) Loss on Final Depth Mean: With only the assumption of the Dirac delta distribution, the estimated accuracy can be limited by the discretization. Therefore we also minimize the ℓ_1 distance between the expectation value in Eq. (6) and the ground truth depth:

$$\mathcal{L}_{mean}(\mathbf{x}) = |\mathbf{D}(\mathbf{x}) - \hat{\mathbf{D}}(\mathbf{x})| \tag{10}$$

In our experiments, we demonstrate that this additional constraint on the expected depth values boosts prediction accuracy.

4) Loss on Final Depth Gradient: To eliminate depth distortion and blurry predictions at object boundaries, we follow [59] and apply a constraint to the depth gradient. This loss penalizes the disagreement of depth edges between $\mathbf{D}$ and $\hat{\mathbf{D}}$, in both vertical and horizontal direction

$$\mathcal{L}_{grad}(\mathbf{x}) = F(|\nabla(\mathbf{D} - \hat{\mathbf{D}})(\mathbf{x})|) \tag{11}$$

where $\nabla(\mathbf{D} - \hat{\mathbf{D}})(\mathbf{x})$ is a 2D vector comprising the horizontal and vertical depth gradients at pixel location $\mathbf{x}$. $F(\xi)$ is a robust activation function formulated as

$$F(\xi) = \sum_{i} \log(\xi_i + 1) \tag{12}$$

where ξ is a vector with each entry denoted by ξ_i.

E. Training

We train the network in three steps. At the first step, we train the differentiable kernel regression module using only $\mathcal{L}_{inter}$, with a learning rate of 0.01 for 10 epochs, resulting in a first approximation of the interpolated depth. Next, we train the residual depth network to further refine the interpolated depth using $\mathcal{L}_{dirac}$, $\mathcal{L}_{mean}$ and $\mathcal{L}_{grad}$. Finally, we train the full model in an end-to-end manner with all loss functions.

IV. EXPERIMENTS

In this section, we first introduce the experimental setup and compare with state-of-the-art methods for guided depth completion. Next, we analyze the benefits of our differentiable kernel regression network and conduct ablation studies wrt.our architecture design and loss functions. Finally, we explore the generalization ability of our method in terms of the number of points and different depth observation patterns.

A. Experimental Setup

1) Datasets: We evaluate the proposed method on NYUv2 [4] and KITTI [60]. For the indoor dataset NYUv2, we follow the official split with 249 scenes for training and 215 scenes for testing. In particular, the test dataset with 654 images is used for evaluation, and 46k images are sampled from the raw training data for training, where missing depth values are inpainted with a cross-bilateral filter provided in the official toolbox. The original image of resolution 640×480 pixels is downsampled and cropped to 304×228 pixels as input following [1], [3], [46].

The outdoor dataset KITTI includes RGB images and depth images collected by Velodyne HDL64. Following [1], we use 46k images from the training sequences for training, and a random subset of 3200 images from the test sequences for evaluation. For fair comparison, we also use the bottom crop (912×228) and evaluate on valid pixels as in [1]. [1]

2) Sparse Measurements: Following [1], [3], [46], we use 500 sparse random depth samples per image (less than 1% of all pixels) on both NYUv2 and KITTI. Note that this setting is more challenging than the setting of the KITTI depth completion benchmark [8] which uses an input density of roughly 10%. We further validate the generalization ability of our method wrt.different sparse input modalities in the ablation study. To this end, we simulate laser scanners with different number of beams on KITTI, by sampling 1/4/8 beams from the original 64 beams.

3) Evaluation Metrics: We adopt the standard evaluation metrics. Let $d_i = \mathbf{D}(\mathbf{x}_i)$ denote the ground truth depth at a pixel location $\mathbf{x}_i$ and $\hat{d}_i = \hat{\mathbf{D}}(\mathbf{x}_i)$ denote the estimated depth. The evaluation metrics are specified as follows:

- Root Mean Squared Error (rms): $\sqrt{\frac{1}{N}\sum_i (\hat{d}_i - d_i)^2}$
- Mean Absolute Relative Error (rel): $\frac{1}{N}\sum_i \frac{|\hat{d}_i - d_i|}{d_i}$
- Threshold δ: percentage of $\hat{d}_i$, s.t. $\max(\frac{\hat{d}_i}{d_i}, \frac{d_i}{\hat{d}_i}) < \delta$, $\delta \in \{1.02, 1.05, 1.10, 1.25, 1.25^2, 1.25^3\}$, see [46].

Here, N denotes the total number of pixels.

4) Implementation Details: We adopt the original U-Net [56] as the backbone of our ResNet. Specifically, it contains 4 downsampling steps and 4 upsampling steps, each with a stride of 2. For KernelNet, we use a shallow version of U-Net by reducing both downsampling and upsampling steps to 1. We set the number of the discrete bins $M = 401$ for all experiments.

[1]Note that $\mathcal{L}_{grad}$ is not applicable to KITTI due to the discontinuity of the ground truth depth $\mathbf{D}$, therefore we only apply $\mathcal{L}_{grad}$ on NYUv2.

B. Comparison to the State-of-the-Art

1) Baselines: We compare to the following state-of-the-art methods on the guided depth completion task.

- Sparse-to-Dense [1] takes sparse measurements as input where non-observed regions are set to 0. The sparse depth map and the guidance image are concatenated to estimate the dense depth map end-to-end.
- Sparse-to-Dense (SS) [47] extends [1] by exploiting self-supervision (SS) via a photometric loss between neighboring frames.
- CSPN [46] stands for Convolutional Spatial Propagation Network and improves Sparse-to-Dense [1] by refining the depth estimation using a recurrent spatial propagation model.
- Nconv-CNN [49] proposes a normalized convolutional layer which takes as input sparse measurements and a confidence map for unguided depth completion. The output of the unguided network is further concatenated with the RGB image for guided depth completion.
- DeepLiDAR [44] also takes sparse measurements as input. It estimates surface normals as the intermediate representation to produce dense depth and thus additionally requires ground truth surface normals as supervision. While the original implementation uses a synthetic dataset with ground truth normals for pre-training, we omit this step for fair comparison. We compute the surface normal ground truth from the depth map following [44] where a plane fitting algorithm is first applied on KITTI to obtain a dense depth map [4].
- D^3-Random is an adapted version of Deep Depth Densification (D^3) [3] which constructs an intermediate dense depth map from the sparse measurements using nearest neighbor interpolation as input to a depth completion network. In the original implementation, more than 500 points are sampled on a regular grid as sparse measurements. This sampling approach can provide richer information due to the even distribution and the larger amount of measurements. For a fair comparison, we retrain and re-evaluate D^3 using the same sparse measurement pattern as ours, which we denote as D^3-Random.

We follow the official implementations of all baselines except for D^3-Random as no official implementation is released.[2]

2) NYUv2 Dataset: Table I (top) compares the performance of different methods on NYUv2. As can be seen, our method achieves superior performance quantitatively. Fig. 3 shows a qualitative comparison between D^3-Random, our interpolated depth map and our final depth map. Note that our differentiable kernel regression layer provides an interpolated depth map that captures the coarse structure of the scene while the residual network is able to further refine details.

3) KITTI Dataset: Table I (bottom) and Fig. 4 show our comparison on KITTI. Our method achieves comparable

[2]Sparse-to-Dense: https://github.com/fangchangma/sparse-to-dense.pytorch, CSPN: https://github.com/XinJCheng/CSPN/tree/master/cspn_pytorch, Nonv-CNN: https://github.com/abdo-eldesokey/nconv-nyu and https://github.com/abdo-eldesokey/nconv, DeepLiDAR: https://github.com/JiaxiongQ/DeepLiDAR, D^3-Random: https://github.com/Shiaoming/DensefromRGBS (third-party implementation).

TABLE I

GUIDED DEPTH COMPLETION ON NYUV2 AND KITTI. ALL METHODS TAKE 500 SAMPLED POINTS AS DEPTH MEASUREMENTS

Dataset	Method	Error (↓)		Accuracy (↑)					
		rms	rel	$\delta_{1.02}$	$\delta_{1.05}$	$\delta_{1.10}$	$\delta_{1.25}$	$\delta_{1.25^2}$	$\delta_{1.25^3}$
NYUv2	Sparse-to-Dense [1]	0.230	0.044	52.3	82.3	92.6	97.1	99.4	99.8
	Sparse-to-Dense (SS) [47]	0.132	0.027	64.3	87.4	95.0	99.0	**99.9**	**100.0**
	CSPN [46]	0.117	0.016	83.2	93.4	97.1	99.2	**99.9**	**100.0**
	Nconv-CNN [49]	0.125	0.017	82.2	92.9	96.7	99.1	99.8	**100.0**
	DeepLiDAR [44]	0.115	0.022	-	-	-	**99.3**	**99.9**	**100.0**
	D^3-Random [3]	0.157	0.025	69.3	88.6	95.3	99.0	99.8	99.9
	Ours	**0.111**	**0.015**	**84.8**	**94.1**	**97.4**	**99.3**	**99.9**	**100.0**
KITTI	Sparse-to-Dense [1]	3.378	0.073	30.0	65.8	85.2	93.5	97.6	98.9
	Sparse-to-Dense (SS) [47]	2.245	0.050	52.4	81.5	91.2	96.1	98.6	99.3
	CSPN [46]	2.977	0.044	70.2	85.7	91.4	95.7	98.0	99.1
	Nconv-CNN [49]	2.739	0.063	50.4	74.2	86.4	94.8	98.2	99.3
	DeepLiDAR [44]	**2.126**	0.044	54.1	82.3	92.0	96.2	**98.7**	**99.4**
	D^3-Random [3]	3.124	0.060	46.4	73.1	85.1	94.1	98.2	99.3
	Ours	2.708	**0.037**	**76.4**	**88.0**	**92.7**	**96.3**	98.3	99.2

TABLE II

KERNEL PARAMETERS. COMPARISON OF THE INTERPOLATED DEPTH $\tilde{\mathbf{D}}$ WRT. DIFFERENT KERNEL PARAMETERS

Kernel Regression	Error (↓)		Accuracy (↑)					
	rms	rel	$\delta_{1.02}$	$\delta_{1.05}$	$\delta_{1.10}$	$\delta_{1.25}$	$\delta_{1.25^2}$	$\delta_{1.25^3}$
Data-independent Kernel	0.230	0.045	54.6	75.6	87.6	96.7	99.5	99.8
Steering Kernel [52]	0.221	0.043	55.7	77.2	88.4	96.9	99.5	99.8
Learned Steering Kernel	**0.198**	**0.034**	**65.5**	**82.9**	**91.6**	**97.7**	**99.8**	**99.9**

performance with DeepLiDAR which requires additional supervision and outperforms the other baselines. As illustrated in Fig. 4, compared to the baseline, our method provides more reliable depth estimates for reflective surfaces and thin structures where fewer and more noisy depth inputs are available.

C. Analysis of Kernel Regression Network

1) Learned Steering Kernels: We validate the advantages of our learned steering kernels compared to hand-crafted kernels in Table II. As we focus on analyzing the advantage of the learned kernel shapes, for this experiment we directly report the performance on the interpolated depth map rather than the final depth map. Specifically, we compare to two baselines for determining the kernel parameters. The first baseline adopts data-independent kernels, i.e., the same scaling, rotation and elongation parameters are applied at all pixel locations. We perform grid search to determine these parameters. For the second baseline, we consider the original steering kernel regression method which heuristically constructs the steering kernels based on image gradients [52].

Our results suggest that the steering kernel [52] outperforms the data-independent kernel, while our learned steering kernel further improves performance compared to both heuristically designed kernels, demonstrating that our estimated kernel parameters conditioned on the guidance image are able to effectively reduce the estimation bias.

2) Learning Process: We further investigate the learning process of the kernel shapes qualitatively. Fig. 5 visualizes three Gaussian kernels over multiple training iterations, determined by the learned kernel parameters $(\gamma_i, \theta_i, \sigma_i)$ at corresponding pixel location $\mathbf{x}_i$. We also show the heuristically designed kernel shapes of [52] for comparison. We observe that the elongated axes converge to align with the edges of the guidance image. In contrast, the heuristic kernel shapes [52] are less aligned with the image edges, suggesting that it is hard to capture the local image properties by hand.

3) Qualitative Analysis: We also visualize the learned kernel maps in Fig. 6. We observe that the kernel shape is highly correlated with local statistics of the guidance image. For example, the scale parameter γ_i is larger at smooth regions (lighter color) while smaller around edges (darker color). The rotation parameter $\boldsymbol{\theta}_i$ clearly distinguishes vertical vs. horizontal edges. The elongation parameters $\boldsymbol{\sigma}_i$ is larger around the edges and smaller in smooth regions.

D. Ablation Study

1) Architecture and Loss Function: To discover the contribution of each part of the model, we conduct ablation studies on NYUv2 in Table III. Specifically, we first perform monocular depth estimation, taking only the guidance image as input ("RGB") and predicting the depth map using our residual network without the global skip connection ("U-Net"). Next, we add the sparse depth map as an additional channel

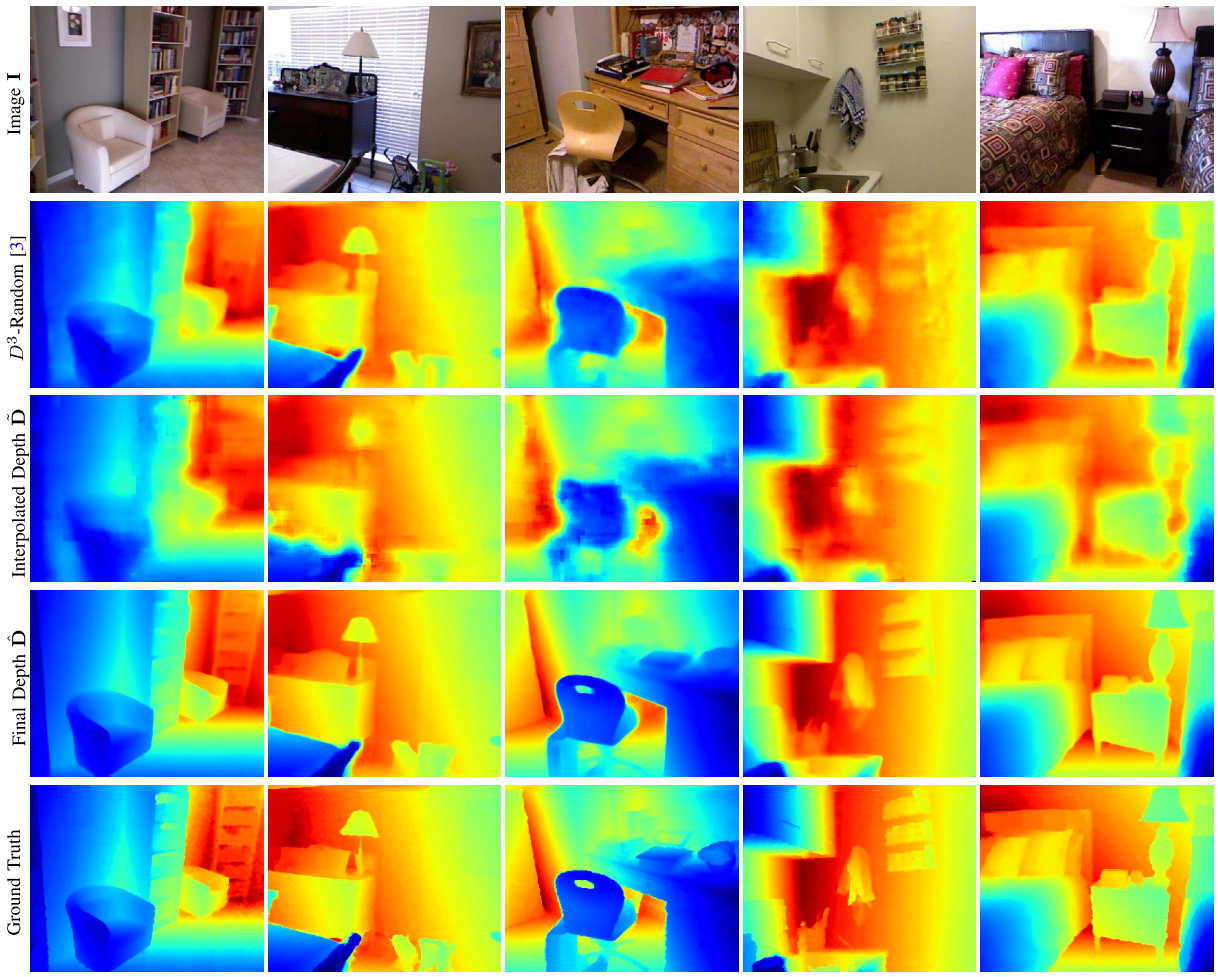


Fig. 3. **Qualitative Comparison on NYUv2.** From top-to-bottom: Guidance image **I**, depth completion results of D^3-Random [3], our interpolated depth map $\tilde{\mathbf{D}}$, our final depth map $\hat{\mathbf{D}}$ and the ground truth depth map.

TABLE III

ABLATION STUDY ON NYUV2. WE COMPARE THE RESULTS OF OUR METHOD WITH RESPECT TO DIFFERENT INPUT MODALITIES, LOSS FUNCTIONS AND NETWORK ARCHITECTURES. FOR THE LATTER, WE USE THE RGB IMAGE (WITH AND WITHOUT THE SPARSE DEPTH INPUT) DIRECTLY FOR OUR RESIDUAL NETWORK WITHOUT GLOBAL SKIP CONNECTION ("U-NET") AS WELL AS OUR FULL MODEL (FIG. 2) WITHOUT ("KERNELNET + U-NET") AND WITH ("KERNELNET + RESNET") GLOBAL SKIP CONNECTION

Input	Loss	Network Architecture	Error (↓)		Accuracy (↑)					
			rms	rel	$\delta_{1.02}$	$\delta_{1.05}$	$\delta_{1.10}$	$\delta_{1.25}$	$\delta_{1.25^2}$	$\delta_{1.25^3}$
RGB	$\mathcal{L}_{dirac}$	U-Net	0.804	0.220	7.29	17.6	32.7	63.0	87.4	95.9
RGB	$\mathcal{L}_{dirac}$+$\mathcal{L}_{mean}$	U-Net	0.819	0.223	7.31	17.6	33.0	64.2	87.0	95.3
RGB	$\mathcal{L}_{dirac}$+$\mathcal{L}_{mean}$+$\mathcal{L}_{grad}$	U-Net	0.791	0.219	7.51	18.2	33.7	64.3	87.7	95.8
RGB+Sparse	$\mathcal{L}_{dirac}$	U-Net	0.281	0.039	57.6	81.0	92.2	97.6	99.5	99.9
RGB+Sparse	$\mathcal{L}_{dirac}$+$\mathcal{L}_{mean}$	U-Net	0.198	0.027	69.9	87.9	94.4	98.1	99.6	99.9
RGB+Sparse	$\mathcal{L}_{dirac}$+$\mathcal{L}_{mean}$+$\mathcal{L}_{grad}$	U-Net	0.189	0.024	71.6	90.4	96.1	98.9	99.6	99.8
RGB+Sparse	$\mathcal{L}_{dirac}$	KernelNet + U-Net	0.176	0.034	58.5	81.6	92.8	98.3	99.7	99.9
RGB+Sparse	$\mathcal{L}_{dirac}$+$\mathcal{L}_{mean}$	KernelNet + U-Net	0.147	0.024	71.5	90.0	95.4	98.8	99.8	99.9
RGB+Sparse	$\mathcal{L}_{dirac}$+$\mathcal{L}_{mean}$+$\mathcal{L}_{grad}$	KernelNet + U-Net	0.134	0.022	72.7	90.5	96.2	99.0	99.8	**100.0**
RGB+Sparse	$\mathcal{L}_{dirac}$	KernelNet + ResNet	0.122	0.016	84.1	93.4	96.9	99.2	99.8	**100.0**
RGB+Sparse	$\mathcal{L}_{dirac}$+$\mathcal{L}_{mean}$	KernelNet + ResNet	0.115	0.015	84.6	94.0	97.3	**99.3**	**99.9**	**100.0**
RGB+Sparse	$\mathcal{L}_{dirac}$+$\mathcal{L}_{mean}$+$\mathcal{L}_{grad}$	KernelNet + ResNet	**0.111**	**0.015**	**84.8**	**94.1**	**97.4**	**99.3**	**99.9**	**100.0**

to the input ("RGB+Sparse") similar to [1], [43] in which the pixels with unknown depth are set to 0. We further add our differentiable kernel regression module and directly estimate the final depth without the global skip connection ("KernelNet + U-Net"). Finally, we use our full model by adding the global skip connection ("KernelNet + ResNet").

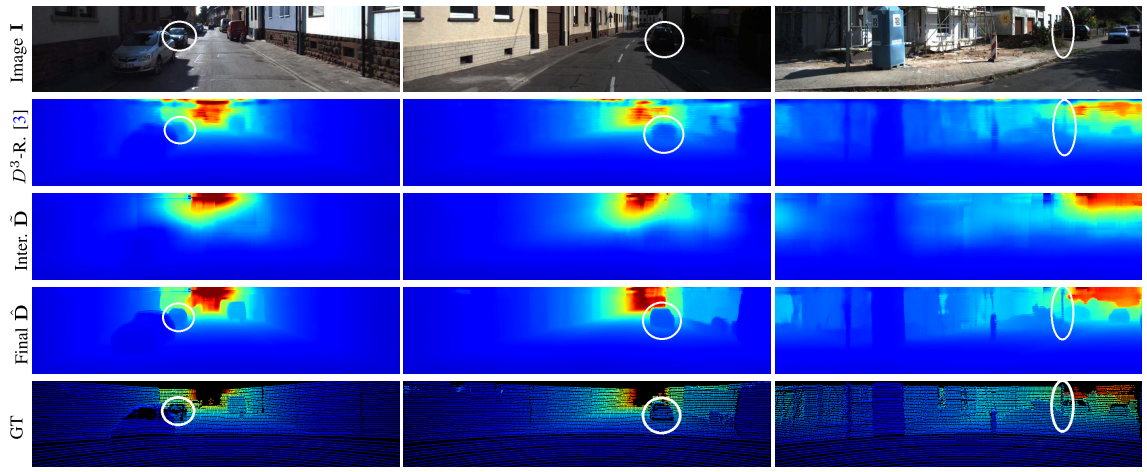


Fig. 4. **Qualitative Comparison on KITTI.** From top-to-bottom: Guidance image $\mathbf{I}$, depth completion results of D^3-Random [3], our interpolated depth map $\tilde{\mathbf{D}}$, our final depth map $\hat{\mathbf{D}}$ and the ground truth depth map (visually enhanced). White circles highlight reflective surfaces and thin structures where our method provides more reliable depth estimates compared to the baseline D^3-Random [3].

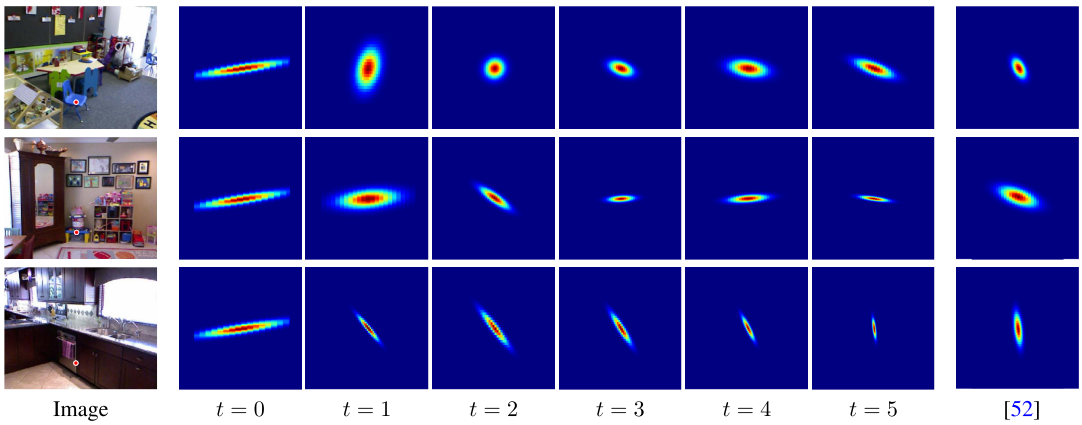


Fig. 5. **Kernel Shape Evolution across Learning Iterations.** t denotes the number of training iterations. The red dots in the left column denote the locations of the visualized kernels. All parameters of the kernel parameter network are randomly initialized, the kernel shapes are almost identical before training and change over iterations. We also show the heuristic kernels proposed by Takeda et al. [52] in the right most column for comparison.

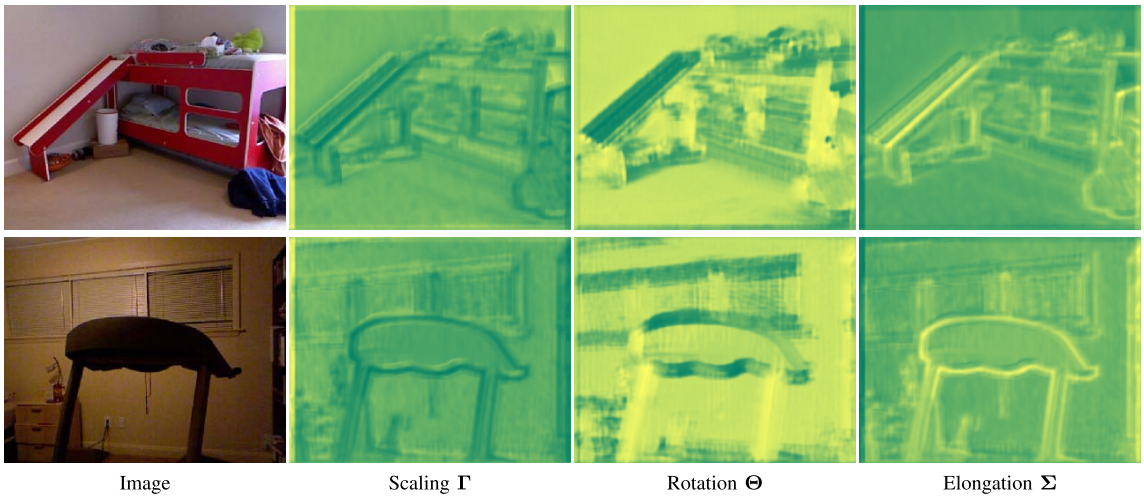


Fig. 6. **Learned Kernel Parameters on NYUv2.** Brighter colors denote larger values. Each kernel is represented with 3 parameters, including scaling $\mathbf{\Gamma}$, rotation $\mathbf{\Theta}$ and elongation $\mathbf{\Sigma}$. The scale parameters are larger in smooth regions and smaller around edges. The rotation parameters clearly distinguish vertical vs. horizontal edges. The elongation parameters are larger around the edges and smaller in smooth regions.

We observe that all variants of "RGB+Sparse" outperform "RGB", taking advantage of the sparse measurements. More importantly, we observe noticeable performance gains by replacing the sparse depth map with our interpolated depth map from the kernel regression network (KernelNet), demonstrating that standard convolutional networks suffer in

TABLE IV

GENERALIZATION. EVALUATION OF OUR MODEL IN TERMS OF GENERALIZATION WITH RESPECT TO THE NUMBER OF POINTS (TOP) AND DIFFERENT DEPTH OBSERVATION PATTERNS (BOTTOM). "1-BEAM [2]" DENOTES A BASELINE WITHOUT LEARNED KERNEL

Dataset	Input	Error (↓)		Accuracy (↑)					
		rms	rel	$\delta_{1.02}$	$\delta_{1.05}$	$\delta_{1.10}$	$\delta_{1.25}$	$\delta_{1.25^2}$	$\delta_{1.25^3}$
NYUv2	300 points	0.149	0.020	79.2	91.2	95.9	98.8	99.7	99.9
	400 points	0.123	0.016	83.0	93.3	97.0	99.2	99.8	100.0
	500 points	0.111	0.015	84.8	94.1	97.4	99.3	99.9	100.0
	600 points	0.106	0.014	85.9	94.6	97.6	99.4	99.9	100.0
	700 points	0.101	0.013	86.8	95.0	97.8	99.5	99.9	100.0
	1000 points	0.092	0.012	88.6	95.7	98.1	99.6	99.9	100.0
	1500 points	0.083	0.010	90.1	96.4	98.4	99.6	99.9	100.0
	2000 points	0.078	0.010	91.0	96.8	98.6	99.7	100.0	100.0
KITTI	1-beam [2]	4.500	0.113	–	–	–	87.4	96.0	98.4
KITTI	1-beam	3.686	0.094	29.9	52.0	72.7	90.1	96.3	98.3
	4-beam	3.593	0.085	33.8	61.6	78.5	91.1	96.9	98.6
	8-beam	3.105	0.048	68.1	84.0	90.4	95.2	97.9	99.0
	500 points	2.708	0.037	76.4	88.0	92.7	96.3	98.3	99.2

the presence of extremely sparse inputs. The comparison between "KernelNet + U-Net" and "KernelNet + ResNet" demonstrates the advantage of the residual formulation with the global skip connection.

In our ablation study on the loss functions, we compare the performance of $\mathcal{L}_{dirac}$ with a combination of $\mathcal{L}_{dirac}$ and $\mathcal{L}_{mean}$. We find that adding $\mathcal{L}_{mean}$ improves the performance in all model variants. The gradient loss $\mathcal{L}_{grad}$ further improves the performance on all metrics.

2) Generalization wrt.Number of Points: We investigate the generalization ability of our model wrt.the number of depth input points in Table IV (top). Specifically, we train the network taking 500 points as sparse measurements and change the number of observed points at inference time. Table IV shows that the performance is correlated with the number of observed points, indicating that the network generalizes well wrt.different sparsity levels. Note that our method is able to achieve better performance given more sparse measurements while it is trained on 500 measurements. This is particularly valuable in applications where the sparse measurements are obtained from visual SLAM, in which case the number of points might differ at every frame. We further compare to CSPN [46] and NConv-CNN [49] given different number of sparse measurements at inference time. Fig. 7 shows that our method consistently outperforms the baselines.

3) Generalization wrt.Observation Pattern: We further validate the generalization ability of our method wrt.various sparse measurement patterns such as laser scans with different number of beams. Specifically, we evaluate our guided depth completion performance given 1/4/8 beams of laser scans and 500 random sparse measurements on KITTI in Table IV (bottom). We retrain the network for each type of measurements in this case due to the large variation of these different patterns. We first compare with [2] that uses 1 beam of laser range data as a baseline. Note that given the same sparse observation, our method outperforms [2] exploiting the learned interpolation kernels. Furthermore, depth completion performance can be improved by increasing the beams of the laser range scanner, validating the generalization ability of our method wrt.different sensory information. Moreover, 500 random samples yield higher precision than 500 points located on a 1-beam laser scanline. This suggests that the observed points from a single beam of laser range data are correlated and thus not as informative as uniformly sampled points.

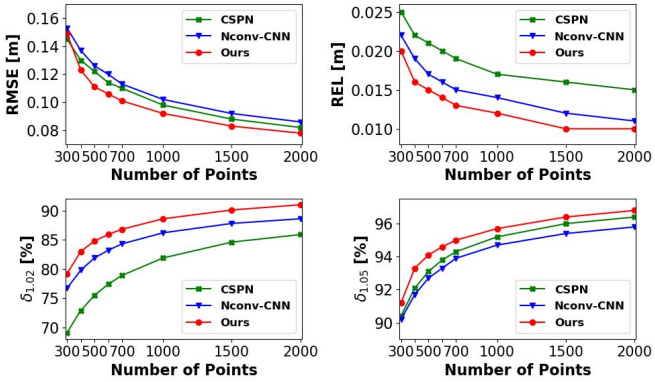


Fig. 7. **Generalization wrt. Number of Points on NYUv2.** All methods are trained on 500 sparse measurements and evaluated on different number of sparse measurements. Top row: lower is better; bottom row: higher is better.

TABLE V

COMPUTATIONAL COST ANALYSIS. INFERENCE TIME EVALUATED ON A SINGLE NVIDIA 1080 TI GPU

	KernelNet	ResNet
Average Inference Time (s)	0.085	0.029
Number of Parameters (M)	3.065	13.422

E. Computational Cost Analysis

Lastly, we evaluate the average inference time and the number of parameters of our proposed method in Table V. The

inference time is evaluated on a single NVIDIA 1080 Ti GPU at an input resolution of 304×228 pixels. Table V shows that our full model, including both KernelNet and ResNet, requires 0.114 seconds for inference. Despite having less parameters, KernelNet is more time-consuming due to the differentiable kernel regression operation (w/o the kernel parameter network) which takes 0.048 seconds for one forward pass.

V. Conclusion

This paper proposes a novel guided depth completion method. By integrating differentiable kernel regression into the guided depth completion formulation, our method avoids the application of convolutions on extremely sparse depth maps while still being end-to-end trainable. We conduct experiments on both indoor and outdoor datasets including NYUv2 and KITTI. We experimentally show that our method is able to achieve superior performance compared to methods which directly learn from sparse depth maps, as well as methods that use hand-crafted interpolated depth maps as input. Our ablation study demonstrates the advantages of our learned steering kernel and analyzes the effectiveness of our architecture design. We also show the generalization ability of our method with respect to various sparse measurement patterns, including randomly sampled sparse measurements and sparse LiDAR measurements. In future work, we plan to investigate if the differentiable kernel regression module can be applied to other image processing tasks such as image denoising.

References

[1] F. Ma and S. Karaman, "Sparse-to-dense: Depth prediction from sparse depth samples and a single image," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 1–8.

[2] Y. Liao, L. Huang, Y. Wang, S. Kodagoda, Y. Yu, and Y. Liu, "Parse geometry from a line: Monocular depth estimation with partial laser observation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 5059–5066.

[3] Z. Chen, V. Badrinarayanan, G. Drozdov, and A. Rabinovich, "Estimating depth from RGB and sparse sensing," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 176–192.

[4] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2012, pp. 746–760.

[5] Y. Liao, S. Kodagoda, Y. Wang, L. Shi, and Y. Liu, "Understand scene categories by objects: A semantic regularized scene classifier using convolutional neural networks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 2318–2325.

[6] Y. Wang, S. Huang, R. Xiong, and J. Wu, "A framework for multi-session RGBD SLAM in low dynamic workspace environment," *CAAI Trans. Intell. Technol.*, vol. 1, no. 1, pp. 90–103, Jan. 2016.

[7] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[8] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, "Sparsity invariant CNNs," in *Proc. Int. Conf. 3D Vis. (3DV)*, Oct. 2017, pp. 11–20.

[9] A. Saxena, S. H. Chung, and A. Y. Ng, "Learning depth from single monocular images," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2005, pp. 1161–1168.

[10] A. Saxena, M. Sun, and A. Y. Ng, "Make3D: Learning 3D scene structure from a single still image," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 824–840, May 2009.

[11] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2014, pp. 2366–2374.

[12] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2650–2658.

[13] F. Liu, C. Shen, G. Lin, and I. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2024–2039, Oct. 2016.

[14] M. Mancini, G. Costante, P. Valigi, and T. A. Ciarfuglia, "Fast robust monocular depth estimation for obstacle detection with fully convolutional networks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 4296–4303.

[15] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *Proc. 4th Int. Conf. 3D Vis. (3DV)*, Oct. 2016, pp. 239–248.

[16] Y. Cao, Z. Wu, and C. Shen, "Estimating depth from monocular images as classification using deep fully convolutional residual networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 11, pp. 3174–3182, Nov. 2018.

[17] B. Li, Y. Dai, and M. He, "Monocular depth estimation with hierarchical fusion of dilated CNNs and soft-weighted-sum inference," *Pattern Recognit.*, vol. 83, pp. 328–339, Nov. 2018.

[18] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," 2019, *arXiv:1907.01341*. [Online]. Available: http://arxiv.org/abs/1907.01341

[19] C. Godard, O. M. Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6602–6611.

[20] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6612–6619.

[21] O. Mac Aodha, N. D. F. Campbell, A. Nair, and G. J. Brostow, "Patch based synthesis for single depth image super-resolution," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2012, pp. 71–84.

[22] M. Hornacek, C. Rhemann, M. Gelautz, and C. Rother, "Depth super resolution by rigid body self-similarity in 3D," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 1123–1130.

[23] Q. Yang, R. Yang, J. Davis, and D. Nistér, "Spatial-depth super resolution for range images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2007, pp. 1–8.

[24] D. Chan, H. Buisman, C. Theobalt, and S. Thrun, "A noise-aware filter for real-time depth upsampling," in *Proc. Workshop Multi-Camera Multi-Modal Sensor Fusion Algorithms Appl.*, 2008.

[25] J. Dolson, J. Baek, C. Plagemann, and S. Thrun, "Upsampling range data in dynamic environments," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 1141–1148.

[26] M.-Y. Liu, O. Tuzel, and Y. Taguchi, "Joint geodesic upsampling of depth images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 169–176.

[27] J. Diebel and S. Thrun, "An application of Markov random fields to range sensing," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2005, pp. 291–298.

[28] A. Harrison and P. Newman, "Image and sparse laser fusion for dense scene reconstruction," in *Proc. Field Service Robot.*, 2009, pp. 219–228.

[29] J. Park, H. Kim, Y.-W. Tai, M. S. Brown, and I. Kweon, "High quality depth map upsampling for 3D-TOF cameras," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 1623–1630.

[30] D. Ferstl, C. Reinbacher, R. Ranftl, M. Rüther, and H. Bischof, "Image guided depth upsampling using anisotropic total generalized variation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 993–1000.

[31] P. Pinies, L. M. Paz, and P. Newman, "Too much TV is bad: Dense reconstruction from sparse laser with non-convex regularisation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 135–142.

[32] G. Riegler, D. Ferstl, M. Rüther, and H. Bischof, "A deep primal-dual network for guided depth super-resolution," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2016.

[33] V. Jampani, M. Kiefel, and P. V. Gehler, "Learning sparse high dimensional filters: Image filtering, dense CRFs and bilateral neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4452–4461.

[34] T. Hui, C. C. Loy, and X. Tang, "Depth map super-resolution by deep multi-scale guidance," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 353–369.

[35] Y. Wen, B. Sheng, P. Li, W. Lin, and D. D. Feng, "Deep color guided coarse-to-fine convolutional network cascade for depth image super-resolution," *IEEE Trans. Image Process.*, vol. 28, no. 2, pp. 994–1006, Feb. 2019.

[36] D. Doria and R. J. Radke, "Filling large holes in LiDAR data by inpainting depth gradients," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2012, pp. 65–72.

[37] J. Liu, X. Gong, and J. Liu, "Guided inpainting and filtering for Kinect depth maps," in *Proc. Int. Conf. Pattern Recognit. (ICPR)*, 2012, pp. 2055–2058.
[38] D. Herrera, J. Kannala, and J. Heikkilä, "Depth map inpainting under a second-order smoothness prior," in *Proc. Scandin. Conf. Image Anal.*, 2013, pp. 555–566.
[39] J. T. Barron and B. Poole, "The fast bilateral solver," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 617–632.
[40] J. T. Barron and J. Malik, "Intrinsic scene properties from a single RGB-D image," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 4, pp. 690–703, Apr. 2016.
[41] Y. Zhang and T. Funkhouser, "Deep depth completion of a single RGB-D image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 175–185.
[42] Z. Huang, J. Fan, S. Cheng, S. Yi, X. Wang, and H. Li, "HMS-net: Hierarchical multi-scale sparsity-invariant network for sparse depth completion," *IEEE Trans. Image Process.*, vol. 29, pp. 3429–3441, 2020.
[43] C. Cadena, A. R. Dick, and I. D. Reid, "Multi-modal auto-encoders as joint estimators for robotics scene understanding," in *Proc. Robot., Sci. Syst. (RSS)*, 2016, p. 1.
[44] J. Qiu *et al.*, "DeepLiDAR: Deep surface normal guided depth prediction for outdoor scene from sparse LiDAR data and single color image," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3313–3322.
[45] M. Jaritz, R. D. Charette, E. Wirbel, X. Perrotton, and F. Nashashibi, "Sparse and dense data with CNNs: Depth completion and semantic segmentation," in *Proc. Int. Conf. 3D Vis. (3DV)*, Sep. 2018, pp. 52–60.
[46] X. Cheng, P. Wang, and R. Yang, "Depth estimation via affinity learned with convolutional spatial propagation network," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 108–125.
[47] F. Ma, G. V. Cavalheiro, and S. Karaman, "Self-supervised sparse-to-dense: Self-supervised depth completion from LiDAR and monocular camera," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 3288–3295.
[48] S. S. Shivakumar, T. Nguyen, I. D. Miller, S. W. Chen, V. Kumar, and C. J. Taylor, "DFuseNet: Deep fusion of RGB and sparse depth information for image guided dense depth completion," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 13–20.
[49] A. Eldesokey, M. Felsberg, and F. S. Khan, "Confidence propagation through CNNs for guided sparse depth regression," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2423–2436, Oct. 2020.
[50] E. A. Nadaraya, "On estimating regression," *Theory Probab. Appl.*, vol. 9, no. 1, pp. 141–142, 1964.
[51] G. S. Watson, "Smooth regression analysis," *Sankhyā, Indian J. Statist. A*, pp. 359–372, Dec. 1964.
[52] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," *IEEE Trans. Image Process.*, vol. 16, no. 2, pp. 349–366, Feb. 2007.
[53] K. Q. Weinberger and G. Tesauro, "Metric learning for kernel regression," in *Proc. Conf. Artif. Intell. Statist. (AISTATS)*, 2007, pp. 612–619.
[54] Y.-K. Noh, M. Sugiyama, K.-E. Kim, F. Park, and D. D. Lee, "Generative local metric learning for kernel regression," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 2452–2462.
[55] S. Gershman and N. D. Goodman, "Amortized inference in probabilistic reasoning," in *Proc. Annu. Meeting Cognit. Sci. Soc.*, 2014, pp. 1–7.
[56] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Med. Image Comput. Comput.-Assist. Intervent. (MICCAI)*, 2015, pp. 234–241.
[57] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[58] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 448–456.
[59] J. Hu, M. Ozay, Y. Zhang, and T. Okatani, "Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2019, pp. 1043–1051.
[60] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 3354–3361.

Lina Liu received the B.S. degree in automation from Zhejiang University in 2018, where she is currently pursuing the Ph.D. degree with the Department of Control Science and Engineering, Institute of Cyber Systems and Control. Her research interests include computer vision and deep learning.

Yiyi Liao received the Ph.D. degree from the Department of Control Science and Engineering, Zhejiang University, China, in 2018. She is currently a Post-Doctoral Researcher with the Autonomous Vision Group, Max Planck Institute for Intelligent Systems and University of Tübingen, Germany. Her research interests include 3D vision and scene understanding.

Yue Wang received the Ph.D. degree from the Department of Control Science and Engineering, Zhejiang University, China, in 2016. He is currently an Associate Professor with the Department of Control Science and Engineering, Zhejiang University. His latest research interests include mobile robotics and robot perception.

Andreas Geiger received the Ph.D. degree in computer science from the Karlsruhe Institute of Technology (KIT), Germany, in 2013. He is currently a Professor in computer science with the University of Tübingen, Germany, and a group leader at the Max Planck Institute for Intelligent Systems, Tübingen, Germany.

Yong Liu received the B.S. degree in computer science and engineering and the Ph.D. degree in computer science from Zhejiang University in 2001 and 2007, respectively. He is currently a Professor with the Institute of Cyber Systems and Control, Department of Control Science and Engineering, Zhejiang University. He has published more than 30 research papers in machine learning, computer vision, information fusion, and robotics. His latest research interests include machine learning, robotics vision, information processing, and granular computing.