

# Monocular Depth Prediction through Continuous 3D Loss

Minghan Zhu<sup>1</sup>, Maani Ghaffari<sup>1</sup>, Yuanxin Zhong<sup>1</sup>, Pingping Lu<sup>1</sup>,  
Zhong Cao<sup>2</sup>, Ryan M. Eustice<sup>1</sup> and Hwei Peng<sup>1</sup>

**Abstract**—This paper reports a new continuous 3D loss function for learning depth from monocular images. The dense depth prediction from a monocular image is supervised using sparse LIDAR points, which enables us to leverage available open source datasets with camera-LIDAR sensor suites during training. Currently, accurate and affordable range sensor is not readily available. Stereo cameras and LIDARs measure depth either inaccurately or sparsely/costly. In contrast to the current point-to-point loss evaluation approach, the proposed 3D loss treats point clouds as continuous objects; therefore, it compensates for the lack of dense ground truth depth due to LIDAR’s sparsity measurements. We applied the proposed loss in three state-of-the-art monocular depth prediction approaches DORN, BTS, and Monodepth2. Experimental evaluation shows that the proposed loss improves the depth prediction accuracy and produces point-clouds with more consistent 3D geometric structures compared with all tested baselines, implying the benefit of the proposed loss on general depth prediction networks. A video demo of this work is available at <https://youtu.be/5HL8BjSAY4Y>.

## I. INTRODUCTION

Range measurement is vital for robots and autonomous vehicles. For ground vehicles, reliable and accurate range sensing is the key for Adaptive Cruise Control, Automatic Emergency Braking, and autonomous driving. With rapid development in deep learning techniques, image-based depth prediction gained much attention and progress, promising cost-effective and accessible range sensing using commercial monocular cameras. However, depth ground truth for an image is not always available for training a neural network. Today, in outdoor scenarios, we mainly rely on LIDAR sensors to provide accurate and detailed depth measurements, but the point clouds are too sparse compared with image pixels. Besides, LIDARs cannot get reliable reflection on some surfaces (e.g. dark, reflective, transparent [1]). Using stereo cameras is another way for range sensing, but it is less accurate for mid to far distance. Generating ground truth depth from an external visual SLAM module [2], [3] suffers similar problems, subject to noise and error.

Due to the lack of perfect ground truth, as discussed above, and the fact that monocular cameras are prevalent, much research effort has been devoted to unsupervised monocular depth learning, which requires only sequences

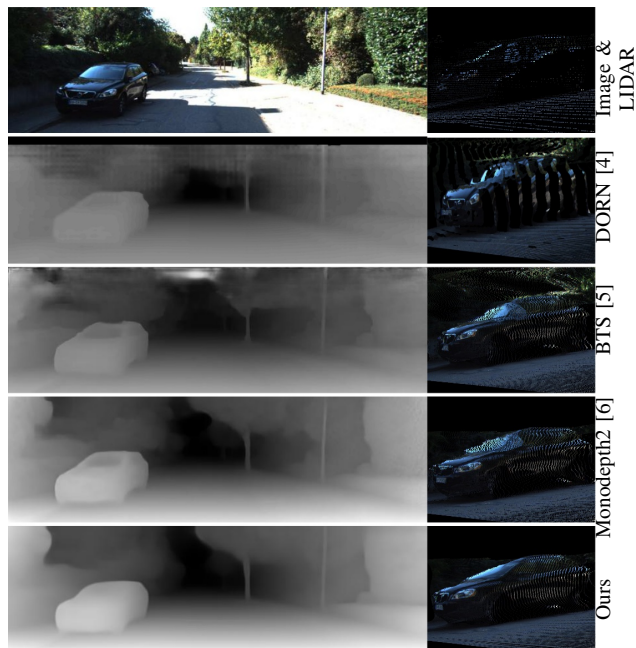


Fig. 1: Visualization of depth prediction. **1st row**: image and raw LIDAR scan of the vehicle colored by the image. **2nd - 4th rows**: depth predictions and point-clouds generated from image pixels with predicted depth using baseline methods. **5th row**: Our results. Our method can build on general depth prediction networks. We tested our method on the above three networks, but the figure only shows our result based on Monodepth2 [6] network for simplicity. This data sample is from KITTI dataset [7].

of monocular images as training data. These approaches have shown promising progress, but there is still a performance gap to supervised approaches (see Table I). Moreover, monocular unsupervised approaches are inherently scale-ambiguous. The depth prediction is relative and needs a scale factor to recover the true depth, meaning that there are real deployment limitations.

Despite that LIDAR sensors are still too expensive for large-scale deployment on vehicles, a number of driving datasets with sensor suites including cameras and LIDARs are already available [8]–[11]. Given such rich datasets, we improve monocular depth prediction by leveraging sparse LIDAR data as ground truth. As stated in BTS [5], currently ranking 1st in monocular depth prediction using the KITTI dataset [7] (Eigen’s split [1]), the high sparsity of ground truth data limits the depth prediction accuracy. Addressing the same issue, we propose a new continuous 3D loss that transforms discrete point clouds into continuous functions. The proposed loss better exploits data correlation in Euclidean and feature spaces, leading to improved performance

\*This work was partially supported by the Toyota Research Institute (TRI), partly under award number N021515.

<sup>1</sup>M. Zhu, M. Ghaffari, Y. Zhong, P. Lu, R. Eustice and H. Peng are with the University of Michigan, Ann Arbor, MI 48109, USA. {minghanz, maanigj, zyxin, pingpinl, eustice, hpeng}@umich.edu

<sup>2</sup>Z. Cao is with Tsinghua University, Beijing, 100084, China. caoc15@mails.tsinghua.edu.cn

of the current deep neural networks. An example is shown in Fig. 1. We note that the proposed 3D loss function is agnostic to the network architecture design, an active research area. The main contributions of this paper include:

- 1) We propose a novel continuous 3D loss function for monocular depth prediction.
- 2) By merely adding this loss to several state-of-the-art monocular depth prediction approaches [4]–[6], without modifying the network structures, we obtain more accurate and geometrically-plausible depth predictions compared with all these baseline methods on KITTI dataset under the supervision of raw LIDAR points.
- 3) Our work is open-sourced and software is available for download at <https://github.com/minghanz/c3d>.

The remainder of this paper is organized as follows. The literature review is given in Sec. II. The proposed new loss function, the theoretical foundation, and its application in monocular depth prediction are introduced in Sec. III. The experimental setup and results are presented in Sec. IV. Section V concludes the paper and provide future work ideas.

## II. RELATED WORK

Deep-learning-based 3D geometric understanding shares similar ideas with SfM/vSLAM approaches. For example, the application of reprojection loss in unsupervised depth prediction approaches [12] and direct methods in SfM/vSLAM [13] are tightly connected. However, they are fundamentally different since the back-propagation of neural networks only takes a small step along the gradient to learn the general prior from large amounts of data gradually. Learning correspondences among different views can assist with recovering the depth [14] if stereo or multi-view images are available as input. For single-view depth prediction, the network needs to learn from more general cues, including perspective, object size, and scenario layout. Although single-view depth prediction is an ill-posed problem in theory since infinite possibilities of 3D layout could result in the same 2D rendered image, this task is still viable since the plausible geometric layouts occur in the real world is limited and can be learned from data.

### A. Supervised single-view depth prediction

It is straight-forward to learn image depth by minimizing the point-wise difference between the predicted depth value and the ground truth depth value. The ground truth depth can come from LIDAR, but such measurements are sparse. One strategy is simply masking out pixels without ground truth depth values and only evaluating loss on valid points [1]. An alternative is to fill in invalid pixels in ground truth maps before evaluation [15], for example using “colorization” methods [16] included in NYU-v2 dataset [17]. While learning from the preprocessed dense depth maps is an easier task, it also limits the accuracy upper bound. The work of [18], [19] used synthetic datasets (e.g. [20], [21]) for training, in which perfect dense ground truth depth maps

are available. However, in practice, the domain difference between synthetic and real data poses a challenge.

### B. Unsupervised single-view depth prediction

The fact that an image’s ground truth depth is hard to obtain and usually sparse and noisy motivates some researchers to apply unsupervised approaches. Stereo cameras with known baseline provide self-supervision in that an image can be reconstructed from its stereo counterpart if the disparity is accurately estimated. Following this idea [22] proposed an end-to-end method to learn single-view depth from stereo data. Using consecutive image frames for self-supervision is similar, except that the camera motions between the consecutive time steps must be estimated and that scale ambiguity may arise. The work of [12] is one of the first proposing to use monocular videos only to learn pose and depth prediction through CNNs in an end-to-end manner. Researchers included an optical flow estimation module [23] and a motion segmentation module to deal with moving objects [24] so that rigid and non-rigid parts are treated separately.

### C. Loss functions in single-view depth prediction

Existing learning methods mainly rely on direct supervision of true depth and indirect supervision of view synthesis error. Most other loss functions are regularization terms. We summarize commonly used loss functions in the following. We omit loss functions from the adversarial learning framework [25], as they require dedicated network structures.

1) *Geometric losses*: Point-wise difference between predicted and ground truth depth values in the norms of L1 [26], L2 [27], Huber [28], *berHu* [15], and the same norms of inverse depth [2] have all been applied, with the consideration of emphasizing prediction error of near/far points. Cross-entropy loss [29] and ordinal loss [30] are applied when depth prediction is formulated as a classification or ordinal regression, instead of regression problems. The negative log-likelihood is adopted in approaches producing probabilistic outputs, e.g., in [31]. [1] introduced a scale-invariant loss to enable learning from data across scenarios with large scale variance. The surface normal difference is also a form of more structured geometric loss [27]. In contrast to the above loss terms which takes value difference in the image space, [32] directly measure geometric loss in the 3D space, minimizing point cloud distance by applying ICP (Iterative Closest Point) algorithms. [33] proposed non-local geometric losses to capture large scale structures.

2) *Non-geometric loss*: This class of loss functions is applied in unsupervised approaches. The most commonly used forms are intensity difference between warped and original pixels, and Structured Similarity (SSIM) [34], which also captures the higher-order statistics of pixels in a local area. In order to handle occlusion and non-rigid scenarios, various adjustments to the photometric errors are proposed. For example, using weight or masking to ignore a subset of pixels that are likely not recovered correctly from view synthesis [12], [27], and [6] used the minimum between forward and backward re-projection error to handle occlusion.

### 3) Regularization losses:

a) *Cross-frame consistency*: It is applied to fully exploit available connections in data between stereo pairs and sequential frames and improve generalizability by enforcing the network to learn view synthesis in different directions. For example, [35] performed view synthesis on a view-synthesized image from the stereo's view, aiming to recover the original image from this loop.

b) *Cross-task consistency*: It is applied to regularize the depth prediction by exploiting the correlation with other tasks, e.g., surface normal prediction [36], optical flow prediction [23], [24], and semantic segmentation [37].

c) *Self-regularization*: These are loss terms that suppress high-order variations in depth predictions. Edge-aware depth smoothness loss [38] is one of the most common example [35], [39]. They are widespread because, in unsupervised approaches, view-synthesis losses rely on image gradients, which are heavily non-convex and only valid in a local region. In supervised approaches, sparse ground truth leaves a subset of points uncovered. Such a regularization term can smooth out the prediction and broadcast supervision signal to a larger region.

Supervision signals in the literature are mostly from pixel-wise values (e.g., depth/reprojection error) and simple statistics in a local region (e.g., surface normal, SSIM), with heuristic regularization terms addressing the locality of such supervision signals. In contrast, we are introducing a new loss term that is smooth and continuous, overcoming such locality with embedded regularization effect.

## III. PROPOSED METHOD

Information captured by LIDAR and camera sensors is a discretized sampling of the real environment in points and pixels. The discretization of the two sensors are different, and a common approach of associating them is to project LIDAR points onto the image frame. This approach has two drawbacks. First, it is an approximation to allocate a pixel location for LIDAR points, subject to rounding error and foreground-background mixture error [40]. Secondly, LIDAR points are much sparser than image pixels, meaning that the supervision signal is propagated from only a small fraction of the image, and surfaces with certain characteristics (e.g., reflective, dark, transparent) are constantly missed due to the limitations of the LIDAR.

To handle the first problem, we evaluate the proposed loss function in the 3D space instead of the image frame. Specifically, we measure the difference between the LIDAR point cloud and the point cloud of image pixels back-projected using the predicted depth. This approach is similar to that of [32], which applied the distance metric of ICP for depth learning. However, since ICP needs the association of point pairs, this approach still suffers from the discretization problem. This problem may not be prominent when both point clouds are from image pixels [32] but is important when using the sparse LIDAR point cloud.

We propose to transform the point cloud into a continuous function, and thus the learning problem becomes aligning

two functions induced by the LIDAR point cloud and the image depth (point cloud). Our approach alleviates the discretization problem, as shown in Sec. IV-D and IV-C in more details.

### A. Function construction from a point cloud

Consider a collection of points,  $X = \{(x_i, \ell_X(x_i))\}_{i=1}^n$ , with each point  $x_i \in \mathbb{R}^3$  and its associated feature vector  $\ell_X(x_i) \in \mathcal{I}$ , where  $(\mathcal{I}, \langle \cdot, \cdot \rangle_{\mathcal{I}})$  is the inner product space of features. To construct a function from a point cloud such as  $X$ , we follow the approach of [41], [42]. That is

$$f = \sum_{i=1}^n \ell_X(x_i) k(\cdot, x_i), \quad (1)$$

where  $k : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}$  is the kernel of a Reproducing Kernel Hilbert Space (RKHS) [43]. Then the inner product with function  $g$  of point cloud  $Z = \{(z_j, \ell_Z(z_j))\}_{j=1}^m$  is given by

$$\langle f, g \rangle = \sum_{i=1}^n \sum_{j=1}^m \langle \ell_X(x_i), \ell_Z(z_j) \rangle_{\mathcal{I}} k(x_i, z_j). \quad (2)$$

For simplicity, let  $c_{ij} := \langle \ell_X(x_i), \ell_Z(z_j) \rangle_{\mathcal{I}}$ . We model the geometric kernel,  $k$ , using the exponential kernel [44, Chapter 4] as

$$k(x, z) = \sigma \exp\left(\frac{-\|x - z\|}{s}\right), \quad (3)$$

where  $\sigma$  and  $s$  are tuneable hyperparameters controlling the size and scale, and  $\|\cdot\|$  is the usual Euclidean norm. While there is no specific restrictions on what kernel to use, we found this kernel providing satisfactory result in practice.

### B. Continuous 3D loss

Let  $Z$  be the LIDAR point cloud that we use as the ground truth, and  $X$  the point cloud from image pixels with depth. We then formulate our continuous 3D loss function as:

$$L_{C3D}(X, Z) = - \sum_{i=1}^n \sum_{j=1}^m c_{ij} k(x_i, z_j), \quad (4)$$

i.e. to maximize the inner product. Different from [41] which aims to find the optimal transformation in the Lie group to align two functions, we operate on points in  $X$ . The gradient of  $L_{C3D}$  w.r.t. a point  $x_i \in X$  is:

$$\frac{\partial L_{C3D}}{\partial x_i} = - \sum_{j=1}^m (c_{ij} \frac{\partial k(x_i, z_j)}{\partial x_i} + \frac{\partial c_{ij}}{\partial x_i} k(x_i, z_j)). \quad (5)$$

For the exponential kernel we have:

$$\frac{\partial k(x_i, z_j)}{\partial x_i} = k(x_i, z_j) \frac{z_j - x_i}{s\|x_i - z_j\|}, \quad (6)$$

and for  $\frac{\partial c_{ij}}{\partial x_i}$  it depends on the specific form of the inner product of the feature space.

In our experiments we design two set of features, i.e.,  $c_{ij} := c_{ij}^v \cdot c_{ij}^n$ . The first one is the color in the HSV space denoted as  $\ell^v$ . We define the inner product in the HSV vector space using the same exponential kernel form and treat  $\ell^v(x)$



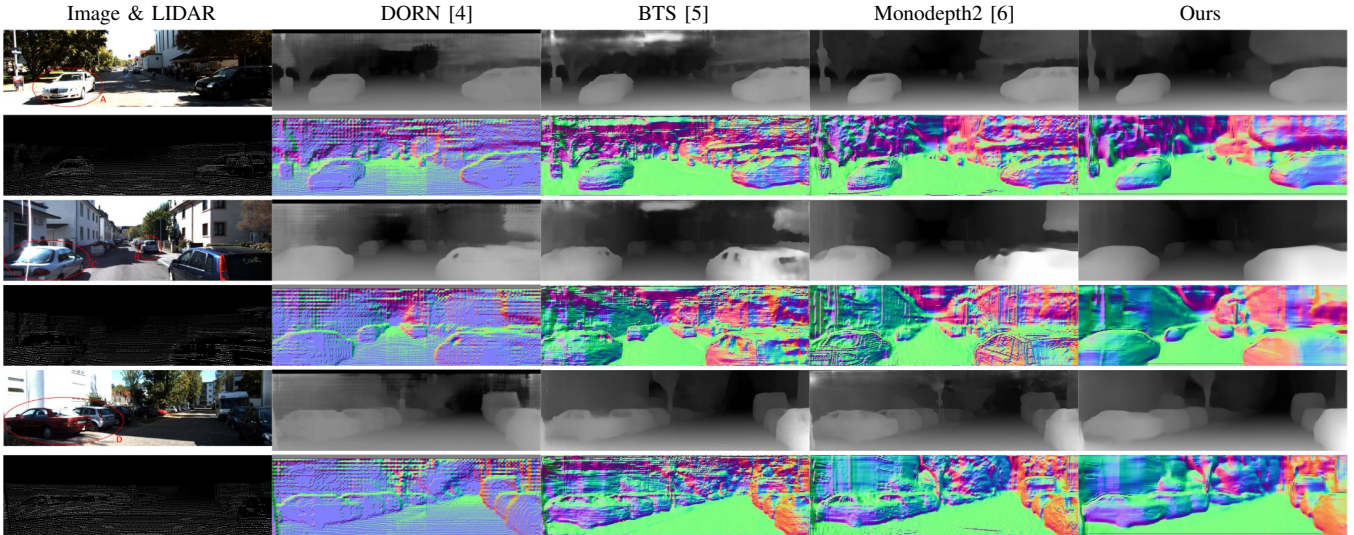


Fig. 2: Qualitative result on KITTI dataset. Three samples are shown. Each corresponds to two rows, showing depth prediction and surface normal directions calculated from predicted depth respectively (except the 1st column showing images and LIDAR point-clouds projected on image frame). Regions highlighted in circles, numbered A, B, C, D, are zoomed in with point-cloud view in Fig. 3.

as a constant. Since the pixel color is invariant w.r.t. its depth,  $\frac{\partial c_{ij}^n}{\partial x_i} = 0$ .

The second feature is the surface normal, denoted  $\ell^n$ , and we use a weighted dot product as the inner product of normal features, i.e.

$$c_{ij}^n := \frac{\ell_X^n(x_i)^T \ell_Z^n(z_j)}{r_X^n(x_i) + r_Z^n(z_j) + \epsilon}, \quad (7)$$

where  $\epsilon$  is to avoid numerical instability, and  $r^n(x)$  denotes the residual, embedding the smoothness of local surface at  $x$ , which is further explained in the following.

Given a point  $x_i$  with normal vector  $\ell_X^n(x_i)$ , the plane defined by the normal is given as:

$$N_{x_i} = \{x : x^T \ell_X^n(x_i) - x_i^T \ell_X^n(x_i) = 0\}. \quad (8)$$

Accordingly, the residual of an arbitrary point  $x'$  w. r. t. this local surface is defined as:

$$r_X^n(x'; x_i) = \frac{\|x'^T \ell_X^n(x_i) - x_i^T \ell_X^n(x_i)\|}{\|x' - x_i\|} \in [0, 1], \quad (9)$$

which equals to the cosine angle between the line  $\overline{x_i x'}$  and the local surface normal. Then the residual of the local surface is defined as:

$$r_X^n(x_i) = \frac{1}{|U(x_i)|} \sum_{x' \in U(x_i)} r_X^n(x'; x_i) \in [0, 1], \quad (10)$$

where  $U(x_i)$  is the set of points in the neighborhood of  $x_i$ , and  $|U(x_i)|$  denotes the number of elements in the set (its cardinality). This term equals to the average of the residual using a neighborhood of the local plane.

The derivative of this kernel w.r.t. the local surface normal vector  $\ell_Z^n(x_i)$  is then given by

$$\frac{\partial c_{ij}^n}{\partial \ell_Z^n(x_i)} = \frac{\ell_Z^n(z_j)}{r_X^n(x_i) + r_Z^n(z_j) + \epsilon}. \quad (11)$$

From the above analysis, we can see that the continuous 3D loss function produces a gradient that combines position difference and normal-direction differences between ground truth points and predicted points weighted by their closeness in the geometric and the feature space. The proposed method avoids point-to-point correspondences that are not always available in data and provides an inherent regularization that can be adjusted with understandable physical meanings.

The exponential operations in  $L_{C3D}$  result in very large numbers compared with other kinds of losses. For numerical stability, we use logarithm of the 3D loss in practice, i.e.

$$L'_{C3D}(X, Z) = \log(L_{C3D}(X, Z)). \quad (12)$$

The continuous 3D loss can be used for cross-frame supervision, in which case relative camera poses also come into play. For example, we can denote:

$$L'_{C3D_{i,j}}(X_i, Z_j) = L'_{C3D_{i,i}}(X_i, T_j^i Z_j). \quad (13)$$

where  $X_i, Z_i$  denotes point-clouds from camera and from LIDAR at frame  $i$ , and  $T_j^i \in SE(3)$  transforms points in coordinate  $j$  to coordinate  $i$ .

### C. Network architecture

To evaluate the effect of the continuous 3D loss function, we modified three state-of-the-art monocular depth prediction approaches: Monodepth2 [6], DORN [4], and BTS [5], by simply including the proposed loss function as an extra loss term. DORN and BTS are supervised depth prediction approaches, which are closely related to our work. Monodepth2 is originally an unsupervised approach, which we included to show that our proposed loss is still effective in the presence of photometric losses. For a fair comparison, we added an L1 depth loss to Monodepth2 so that all baselines are supervised.

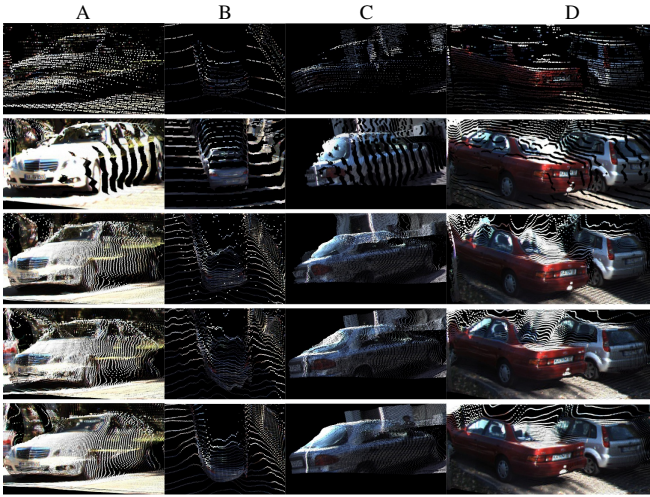


Fig. 3: Point-cloud visualization of vehicles circled in Fig. 2. **From top to bottom:** raw LIDAR colored by image, point clouds generated by DORN [4], BTS [5], Monodepth2 [6], and **by our approach**. The four columns correspond to A, B, C, D in Fig. 2 respectively.

## IV. EXPERIMENTS

### A. Implementation details

The model is implemented in Pytorch, and the training settings are consistent with the three baseline methods, except that the batch size is set to 3 in all methods. Specifically, the backbone feature extraction networks are ResNet-50 [45], ResNet-101, and DenseNet-161 [46], and the training epochs are 20, 40, and 50 for Monodepth2, DORN, and BTS respectively.

We implemented a customized operation in Pytorch to efficiently calculate the inner product on GPU, taking advantage of the sparsity of LIDAR point clouds and the double sum. Such computation only induces a small (5%) time overhead in each iteration.

In practice, LIDAR point clouds are cropped to only keep the front section in the camera view before calculating  $L_{C3D}$ . Besides, we can see from (4) that the calculation of the inner product involves a double sum over all point-pairs in two point clouds. To alleviate the computation burden, we discard point-pairs that are far away from each other in image space, of which the geometric kernel value is likely to be very small, hardly contributing to the loss.

The parameters in  $L_{C3D}$  mainly involves  $\sigma$  and  $s$  in the exponential kernel (3). For the HSV feature kernel, we use  $\sigma_v = 1$ ,  $s_v = 0.2$ . For the geometric kernel, we use  $\sigma_g = 1$ ,  $s_g = s_0 d$ , where  $d$  is the maximum depth in the pair of points involved in the kernel, so that the support of the kernel grows larger for further points. We do not specifically tune the value of  $s_0$ . Instead we sample it in each iteration of training as  $s_0 = 0.01 + 0.02|\alpha|$ ,  $\alpha \sim N(0, 1)$ .

### B. KITTI Dataset

As is common in literature (e.g. [4]–[6]), our experiment is conducted using the KITTI dataset [7], [47]. All three baselines follow Eigen’s data split [48], except that Monodepth2

also used Zhou’s [12] preprocessing to remove static frames in order to avoid degeneration of photometric losses.

We note that there are two versions of "depth ground truth" in the KITTI dataset. The first is the projection from raw LIDAR point cloud [7], whereas the second one is preprocessed in KITTI 2015 depth benchmark [47]. The latter is denser with fewer errors than raw LIDAR projection. This is due to the accumulation of 11 adjacent LIDAR scans and outlier removal by comparing them with stereo estimations. However, this densified depth ground truth is still semi-dense, i.e., not covering all pixels in images. To highlight our purpose of better leveraging LIDAR point clouds and to make the approach more generalizable to other datasets where such preprocessing is unavailable, we use the raw LIDAR point clouds for training and use the refined and denser depth images for evaluation. This results in 652 images in the test set, which are the frames with refined depth images in Eigen’s test split. This setup is different from what is in DORN and BTS; therefore, the baselines’ quantitative result is generated by us and not the same as in the original papers.

### C. Quantitative results and analysis

Consistent with the literature [35], depth is truncated at 80m maximum. We also crop a portion of the images as done in [22] before evaluation. The same setup can also be found in Monodepth2, DORN, and BTS. The definition of all metrics is consistent with those of [1]. In Table I, the quantitative comparison of our method with the baselines and other state-of-the-art approaches is reported. Improvement is achieved by simply adding our proposed continuous 3D loss function to all three baseline methods.

**Remark 1.** *We note that our approach does not outperform BTS results reported in the literature, as shown in Table I. The reason is that BTS is trained using refined and densified KITTI depth. When supervised by raw LIDAR depth, Our experiment shows that the proposed method can improve BTS, DORN, and Monodepth2. Our accuracy lies between baselines trained using raw LIDAR depth and those trained using densified depth. It implies that the ideal case is to have dense supervision. Our method acts as a surrogate to dense supervision when we only have access to sparse supervision.*

### D. Qualitative results and analysis

In order to show the effect of the new continuous 3D loss intuitively, in Fig. 2 we listed a few samples from the KITTI dataset. Each sample includes the RGB image, the raw LIDAR scan, and the predicted depth and corresponding surface normal directions from the baselines and our method. We only show our results based on Monodepth2 network and omit our result based on DORN and BTS due to page limit.

1) *Depth view:* We observe that both Monodepth2 and BTS predict incorrectly at the vehicle-window area from the depth prediction images. It creates “holes” in the depth map and fails to recover the full object contours, as in the second

TABLE I: The quantitative comparison using Eigen’s test split with improved ground truth.

• **Bold** numbers are the best. The rows of “Improvement” are w.r.t. the baselines.

- The “Train” column, “U”: unsupervised, “SS”: supervised by stereo disparity, “LS”: supervised by LIDAR depth, “DS”: supervised by densified KITTI depth. Our experiments are focused on “LS” cases, while results from other supervisions are given for reference.
- The “Source” column shows where the numbers are from. “O”: generated by us based on the official open-source implementation. “U”: generated by us based on unofficial implementation, where we made our best effort to align with the original paper.
- Gray results are with the supervision as in the original papers. They are a better reference than the numbers in “from literature” section because our experiments are conducted with training setups as similar as possible. In contrast, for example, the numbers of DORN and Monodepth2 in “from literature” section are of different backbones from those in our experiments.

|                      |                       | lower is better |        |              |              | higher is better |              |                 |                   |                   |
|----------------------|-----------------------|-----------------|--------|--------------|--------------|------------------|--------------|-----------------|-------------------|-------------------|
| Method               |                       | Train           | Source | Abs Rel      | Sq Rel       | RMSE             | RMSE log     | $\delta < 1.25$ | $\delta < 1.25^2$ | $\delta < 1.25^3$ |
| from literature      | DDVO [3]              | U               | [6]    | 0.126        | 0.866        | 4.932            | 0.185        | 0.851           | 0.958             | 0.986             |
|                      | 3net [49]             | U               | [6]    | 0.102        | 0.675        | 4.293            | 0.159        | 0.881           | 0.969             | 0.991             |
|                      | SuperDepth [50]       | U               | [6]    | 0.090        | 0.542        | 3.967            | 0.144        | 0.901           | 0.976             | 0.993             |
|                      | Monodepth2 [6]        | U               | [6]    | 0.090        | 0.545        | 3.942            | 0.137        | 0.914           | 0.983             | 0.995             |
|                      | SVSM FT [51]          | U+SS            | [52]   | 0.077        | 0.392        | 3.569            | 0.127        | 0.919           | 0.983             | 0.995             |
|                      | semiDepth [52]        | U+DS            | [52]   | 0.078        | 0.417        | 3.464            | 0.126        | 0.923           | 0.984             | 0.995             |
|                      | DORN [4]              | DS              | [52]   | 0.080        | 0.332        | 2.888            | 0.120        | 0.938           | 0.986             | 0.995             |
|                      | BTS [5]               | DS              | [5]    | <b>0.060</b> | <b>0.249</b> | <b>2.798</b>     | <b>0.096</b> | <b>0.955</b>    | <b>0.993</b>      | <b>0.998</b>      |
| from our experiments | Monodepth2 (Baseline) | U+LS            | O      | 0.077        | 0.444        | 3.568            | 0.118        | 0.934           | <b>0.988</b>      | <b>0.997</b>      |
|                      | Monodepth2+C3D (Ours) | U+LS            | O      | 0.072        | 0.370        | 3.371            | 0.116        | 0.937           | <b>0.988</b>      | <b>0.997</b>      |
|                      | Monodepth2            | U               | O      | 0.087        | 0.509        | 3.812            | 0.126        | 0.922           | 0.984             | 0.995             |
|                      | <i>Improvement</i>    | U+LS            | O      | 6.5%         | 16.6%        | 5.5%             | 1.7%         | 0.3%            | 0.0%              | 0.0%              |
|                      | BTS (Baseline)        | LS              | O      | 0.071        | 0.342        | 3.341            | <b>0.115</b> | 0.936           | 0.987             | <b>0.997</b>      |
|                      | BTS+C3D (Ours)        | LS              | O      | <b>0.068</b> | <b>0.326</b> | 3.231            | <b>0.115</b> | <b>0.937</b>    | 0.987             | <b>0.997</b>      |
|                      | BTS                   | DS              | O      | 0.063        | 0.268        | 2.896            | 0.101        | 0.949           | 0.991             | 0.998             |
|                      | <i>Improvement</i>    | LS              | O      | 4.2%         | 4.7%         | 3.3%             | 0.0%         | 0.1%            | 0.0%              | 0.0%              |
| from our experiments | DORN (Baseline)       | LS              | U      | 0.127        | 0.474        | 3.420            | 0.153        | 0.900           | 0.985             | 0.996             |
|                      | DORN+C3D (Ours)       | LS              | U      | 0.117        | 0.409        | <b>3.155</b>     | 0.142        | 0.916           | <b>0.988</b>      | <b>0.997</b>      |
|                      | DORN                  | DS              | U      | 0.110        | 0.358        | 3.064            | 0.133        | 0.927           | 0.991             | 0.998             |
|                      | <i>Improvement</i>    | LS              | U      | 7.9%         | 13.7%        | 7.8%             | 7.2%         | 1.8%            | 0.3%              | 0.1%              |

TABLE II: Quantitative comparison for ablation study on the effect of surface normal kernel.

|         |                       | lower is better |              |              |              | higher is better |                   |                   |  |
|---------|-----------------------|-----------------|--------------|--------------|--------------|------------------|-------------------|-------------------|--|
| Dataset | Method                | Abs Rel         | Sq Rel       | RMSE         | RMSE log     | $\delta < 1.25$  | $\delta < 1.25^2$ | $\delta < 1.25^3$ |  |
| KITTI   | Monodepth2 (Baseline) | 0.077           | 0.444        | 3.568        | 0.118        | 0.934            | <b>0.988</b>      | <b>0.997</b>      |  |
|         | $L_{C3D}^v$           | 0.075           | 0.404        | 3.481        | 0.117        | 0.935            | <b>0.988</b>      | <b>0.997</b>      |  |
|         | $L_{C3D}^{nv}$        | <b>0.072</b>    | <b>0.370</b> | <b>3.371</b> | <b>0.116</b> | <b>0.937</b>     | <b>0.988</b>      | <b>0.997</b>      |  |

and third examples. This area is not handled well by previous methods because

- The window area is a non-Lambertian surface with an inconsistent appearance at different viewing angles; therefore, photometric losses do not work.
- LIDAR does not receive a good reflection from glasses, as can be seen from Fig. 2; therefore, no supervision from the ground truth is available.
- The window area’s color is usually not consistent with other parts of the vehicle body, further failing appearance-based depth smoothness terms.

In contrast, our continuous 3D loss function provides supervision from all nearby points, thus overcoming the problem and providing inherent smoothness. The window-area is predicted correctly with full object shape preserved from our predictions.

DORN presents fewer “holes” and irregular contours in the depth images than the other two baselines. Its classification formulation restricts the possible distortions. It has a side effect that the predicted depths are all from a predefined discrete set of values, which can be observed more intuitively in surface-normal view in Sec. IV-D.2 and point-cloud view

in Sec. IV-D.3.

2) *Surface-normal view*: The surface-normal view provides a better visualization of 3D structures and local smoothness. The second row of each sample in Fig. 2 shows the surface normal direction calculated from the predicted depth. Despite the existence of regularizing smoothness term, the baseline method, Monodepth2, still produces many textures inherited from the color space. This is because the edge-aware smoothness loss is down-weighted at high-gradient pixels. BTS shows less, but still visible, texture and artifacts from color space in the normal map, and the inconsistency in window-area is apparent in surface-normal view. In contrast, our method does not produce such textures while still preserving the 3D structures with a clear shift between different surfaces. DORN produces almost-uniform normal images because the predicted depth value range is discrete.

3) *Point-cloud view*: By back-projecting image pixels using predicted depth to 3D space, we can recover the scene’s point cloud. This view allows us to inspect how well the depth prediction recovers the 3D geometry in the real world. This is important as the pixel-clouds could provide



a denser alternative to accurate-but-sparse LIDAR point clouds, benefiting 3D object detection, as indicated in [53].

Fig. 3 shows four examples. They cover both near and far objects and cases of over-exposure and color-blend-in with the background. We can see that the raw LIDAR scans are quite sparse on dark vehicle bodies and glass surfaces, posing challenges on using such data as ground truth for depth learning. “Holes” in predicted depth map transform to unregulated noise points in 3D view. Compared with the Monodepth2 and BTS baselines, our method produces point clouds with higher quality in both glass and non-glass areas, with a smooth surface and geometric structure consistent with the real vehicles. The shape distortion of DORN-produced point-clouds is also small, but the points all lie on some common vertical planes due to the discrete depth prediction, making the point-clouds unrealistic. Our method is capable of producing well-shaped point-clouds while not bound to this limitation.

4) *Summary*: In the qualitative comparison, we show that our method predicts depth with a smoother shape and less distortion, especially in reflective and transparent areas. This improvement is not fully presented in the quantitative analysis because the ground truth depth in those areas is generally missing.

#### E. Ablation study

We now take a closer look at different configurations in the function constructed from point clouds. Here we mainly investigate the effect of the surface normal kernel. We denote the continuous 3D loss without surface normal kernel as

$$L_{C3D}^v(X, Z) = - \sum_{i=1}^n \sum_{j=1}^m c_{ij}^v k(x_i, z_j) \quad (14)$$

and the one with surface normal kernel as:

$$L_{C3D}^{nv}(X, Z) = - \sum_{i=1}^n \sum_{j=1}^m c_{ij}^n c_{ij}^v k(x_i, z_j) \quad (15)$$

The quantitative comparison is in Table II, following the same setup as Fig. I and a data sample is shown in Fig. 4 for visualization. We only show results using the Monodepth2

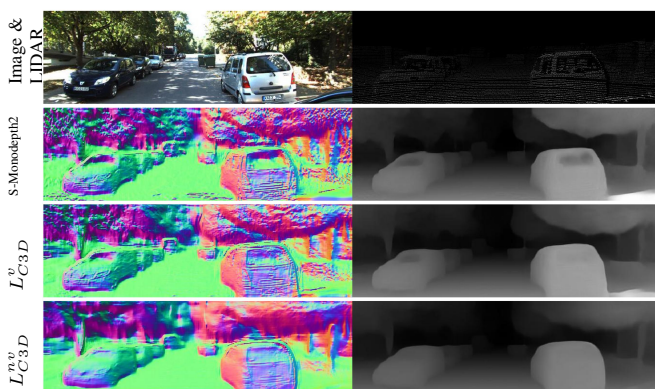


Fig. 4: Visualization of the effect of surface normal kernel. Except for the 1st column, left images are surface normals, and on the right side, corresponding predicted depth images are shown.

baseline due to space limitation. While the continuous loss  $L_{C3D}^v$  improved upon the baseline by exploiting the correlation among points, the prediction still produces artifacts caused by textures in color space. The surface normal kernel is sensitive to local noises and distinguishes between different parts of the 3D geometry, producing more geometrically plausible predictions.

## V. CONCLUSION

We proposed a new continuous 3D loss function for monocular single-view depth prediction. The proposed loss function addresses the gap between dense image prediction and sparse LIDAR supervision. We achieved this by transforming point clouds into continuous functions and aligning them via the function space’s inner product structure. By simply adding this new loss function to existing network architectures, the accuracy and geometric consistency of depth predictions are improved significantly on all three state-of-the-art baseline networks that we tested. The evaluation shows that our contribution is orthogonal to the progress in depth prediction network designs and that our work can benefit general depth prediction networks by applying the continuous 3D loss as a plug-in module.

Future work includes representation learning for features used in the proposed loss function to bring further improvements. Finally, exploring the benefits of the improved depth prediction for 3D object detection is another interesting research direction.

## ACKNOWLEDGMENT

This article solely reflects the opinions and conclusions of its authors and not TRI or any other Toyota entity.

## REFERENCES

- [1] D. Eigen, C. Puhrsch, and R. Fergus, “Depth map prediction from a single image using a multi-scale deep network,” in *Proc. Advances Neural Inform. Process. Syst. Conf.*, 2014, pp. 2366–2374.
- [2] N. Yang, R. Wang, J. Stuckler, and D. Cremers, “Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry,” in *Proc. European Conf. Comput. Vis.*, 2018, pp. 817–833.
- [3] C. Wang, J. Miguel Buenaposada, R. Zhu, and S. Lucey, “Learning depth from monocular videos using direct methods,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 2022–2030.
- [4] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, “Deep ordinal regression network for monocular depth estimation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 2002–2011.
- [5] J. H. Lee, M.-K. Han, D. W. Ko, and I. H. Suh, “From big to small: Multi-scale local planar guidance for monocular depth estimation,” *arXiv preprint arXiv:1907.10326*, 2019.
- [6] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, “Digging into self-supervised monocular depth estimation,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 3828–3838.
- [7] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The kitti dataset,” *Int. J. Robot. Res.*, 2013.
- [8] “Waymo open dataset: An autonomous driving dataset,” 2019.
- [9] R. Kesten, M. Usman, J. Houston, T. Pandya, K. Nadhamuni, A. Ferreira, M. Yuan, B. Low, A. Jain, P. Ondruska, S. Omari, S. Shah, A. Kulkarni, A. Kazakova, C. Tao, L. Platinsky, W. Jiang, and V. Shet, “Lyft level 5 av dataset 2019,” [urlhttps://level5.lyft.com/dataset/](https://level5.lyft.com/dataset/), 2019.
- [10] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, “nuscenes: A multimodal dataset for autonomous driving,” *arXiv preprint arXiv:1903.11027*, 2019.

- [11] Y. Dong, Y. Zhong, W. Yu, M. Zhu, P. Lu, Y. Fang, J. Hong, and H. Peng, "Mcity data collection for automated vehicles study," *arXiv preprint arXiv:1912.06258*, 2019.
- [12] 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 Recog.*, 2017, pp. 1851–1858.
- [13] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "Dtam: Dense tracking and mapping in real-time," in *Proc. IEEE Int. Conf. Comput. Vis.* IEEE, 2011, pp. 2320–2327.
- [14] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox, "Demon: Depth and motion network for learning monocular stereo," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 5038–5047.
- [15] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *Int. Conf. 3D Vis. (3DV)*. IEEE, 2016, pp. 239–248.
- [16] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," in *ACM SIGGRAPH*, 2004, pp. 689–694.
- [17] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgb-d images," in *Proc. European Conf. Comput. Vis.* Springer, 2012, pp. 746–760.
- [18] B. Bozorgtabar, M. S. Rad, D. Mahapatra, and J.-P. Thiran, "Syndemo: Synergistic deep feature alignment for joint learning of depth and ego-motion," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 4210–4219.
- [19] Y. Di, H. Morimitsu, S. Gao, and X. Ji, "Monocular piecewise depth estimation in dynamic scenes by exploiting superpixel relations," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 4363–4372.
- [20] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "Virtual worlds as proxy for multi-object tracking analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 4340–4349.
- [21] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 3234–3243.
- [22] R. Garg, V. K. BG, G. Carneiro, and I. Reid, "Unsupervised cnn for single view depth estimation: Geometry to the rescue," in *Proc. European Conf. Comput. Vis.* Springer, 2016, pp. 740–756.
- [23] Z. Yin and J. Shi, "Geonet: Unsupervised learning of dense depth, optical flow and camera pose," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 1983–1992.
- [24] A. Ranjan, V. Jampani, L. Balles, K. Kim, D. Sun, J. Wulff, and M. J. Black, "Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 12240–12249.
- [25] F. Aleotti, F. Tosi, M. Poggi, and S. Mattoccia, "Generative adversarial networks for unsupervised monocular depth prediction," in *Proc. European Conf. Comput. Vis.*, 2018, pp. 0–0.
- [26] Z. Wu, X. Wu, X. Zhang, S. Wang, and L. Ju, "Spatial correspondence with generative adversarial network: Learning depth from monocular videos," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 7494–7504.
- [27] X. Qi, R. Liao, Z. Liu, R. Urtasun, and J. Jia, "Geonet: Geometric neural network for joint depth and surface normal estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 283–291.
- [28] J. Ye, Y. Ji, X. Wang, K. Ou, D. Tao, and M. Song, "Student becoming the master: Knowledge amalgamation for joint scene parsing, depth estimation, and more," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 2829–2838.
- [29] D. Shin, Z. Ren, E. B. Sudderth, and C. C. Fowlkes, "3d scene reconstruction with multi-layer depth and epipolar transformers," in *Proc. IEEE Int. Conf. Comput. Vis.*, October 2019.
- [30] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 2002–2011.
- [31] F. Brickwedde, S. Abraham, and R. Mester, "Mono-sf: Multi-view geometry meets single-view depth for monocular scene flow estimation of dynamic traffic scenes," in *Proc. IEEE Int. Conf. Comput. Vis.*, October 2019.
- [32] R. Mahjourian, M. Wicke, and A. Angelova, "Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 5667–5675.
- [33] W. Yin, Y. Liu, C. Shen, and Y. Yan, "Enforcing geometric constraints of virtual normal for depth prediction," in *Proc. IEEE Int. Conf. Comput. Vis.*, October 2019.
- [34] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, April 2004.
- [35] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 270–279.
- [36] T. Dharmasiri, A. Spek, and T. Drummond, "Joint prediction of depths, normals and surface curvature from rgb images using cnns," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.* IEEE, 2017, pp. 1505–1512.
- [37] P. Z. Ramirez, M. Poggi, F. Tosi, S. Mattoccia, and L. Di Stefano, "Geometry meets semantics for semi-supervised monocular depth estimation," in *Asian Conference on Computer Vision*. Springer, 2018, pp. 298–313.
- [38] P. Heise, S. Klose, B. Jensen, and A. Knoll, "Pm-huber: Patchmatch with huber regularization for stereo matching," in *Proc. IEEE Int. Conf. Comput. Vis.*, December 2013.
- [39] Y. Kuznetsov, J. Stuckler, and B. Leibe, "Semi-supervised deep learning for monocular depth map prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 6647–6655.
- [40] J. Qiu, Z. Cui, Y. Zhang, X. Zhang, S. Liu, B. Zeng, and M. Pollefeys, "DeepLidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, June 2019.
- [41] M. Ghaffari, W. Clark, A. Bloch, R. M. Eustice, and J. W. Grizzle, "Continuous direct sparse visual odometry from RGB-D images," in *Proc. Robot.: Sci. Syst. Conf.*, Freiburg, Germany, June 2019.
- [42] W. Clark, M. Ghaffari, and A. Bloch, "Nonparametric continuous sensor registration," *arXiv preprint arXiv:2001.04286*, 2020.
- [43] A. Berlines and C. Thomas-Agnan, *Reproducing kernel Hilbert spaces in probability and statistics*. Kluwer Academic, 2004.
- [44] C. Rasmussen and C. Williams, *Gaussian processes for machine learning*. MIT press, 2006, vol. 1.
- [45] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 770–778.
- [46] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 4700–4708.
- [47] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, "Sparsity invariant cnns," in *Int. Conf. 3D Vis. (3DV)*, Oct 2017, pp. 11–20.
- [48] 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.*, December 2015.
- [49] M. Poggi, F. Tosi, and S. Mattoccia, "Learning monocular depth estimation with unsupervised trinocular assumptions," in *Int. Conf. 3D Vis. (3DV)*. IEEE, 2018, pp. 324–333.
- [50] S. Pillai, R. Amrűş, and A. Gaidon, "Superdepth: Self-supervised, super-resolved monocular depth estimation," in *Proc. IEEE Int. Conf. Robot. and Automation*. IEEE, 2019, pp. 9250–9256.
- [51] Y. Luo, J. Ren, M. Lin, J. Pang, W. Sun, H. Li, and L. Lin, "Single view stereo matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 155–163.
- [52] A. J. Amiri, S. Y. Loo, and H. Zhang, "Semi-supervised monocular depth estimation with left-right consistency using deep neural network," *arXiv preprint arXiv:1905.07542*, 2019.
- [53] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger, "Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, June 2019.