NDDepth: Normal-Distance Assisted Monocular Depth Estimation and Completion

Shuwei Shao, Zhongcai Pei, Weihai Chen*, Peter C. Y. Chen and Zhengguo Li, *Fellow, IEEE*

***Abstract*—Over the past few years, monocular depth estimation and completion have been paid more and more attention from the computer vision community because of their widespread applications. In this paper, we introduce novel physics (geometry)-driven deep learning frameworks for these two tasks by assuming that 3D scenes are constituted with piece-wise planes. Instead of directly estimating the depth map or completing the sparse depth map, we propose to estimate the surface normal and plane-to-origin distance maps or complete the sparse surface normal and distance maps as intermediate outputs. To this end, we develop a normal-distance head that outputs pixel-level surface normal and distance. Meanwhile, the surface normal and distance maps are regularized by a developed plane-aware consistency constraint, which are then transformed into depth maps. Furthermore, we integrate an additional depth head to strengthen the robustness of the proposed frameworks. Extensive experiments on the NYU-Depth-v2, KITTI and SUN RGB-D datasets demonstrate that our method exceeds in performance prior state-of-the-art monocular depth estimation and completion competitors.**

***Index Terms*—Monocular depth estimation, Depth completion, Surface normal, Plane-to-origin distance, Piece-wise planar constraint**

I. Introduction

Active depth sensing has made substantial strides in performance and proven its practicality across various applications, including robotics [1], [2], scene understanding [3] and augmented reality [4]. Although specialist hardware sensors, for example, Microsoft Kinect and LiDAR, are capable of capturing accurate depth range, they tend to exhibit difficulties in collecting dense depth maps due to the sensor noise, transparent and reflective surfaces or the limited number of scanning lines. Hence, monocular depth estimation and completion able to generate dense depth maps are developed.

Monocular depth estimation aims to predict the depth map from a single RGB image. Considering that a 2D image can be projected from an infinite number of 3D scenes, such a task is indeed ill-posed and inherently ambiguous. Hence, solving it reliably demonstrates a formidable challenge for traditional methods [5], [6] because of their inherent limitations, typically involving low-dimensional and sparse distances or known and fixed objects. Recently, much progress has been made in this field benefiting from the explosion of deep learning [7]–[11]. Most efforts focus on designing increasingly complicated and powerful networks, which renders the depth estimation a difficult fitting problem without the help of additional guidance.

The mainstream of depth completion is to leverage the RGB image as guidance to complete the sparse depth map. A typical approach is to utilize multiple branches for extracting features from the sparse depth map and its corresponding RGB image, respectively, and later merge them at different scales [12]–[15]. In pursuit of advancing the frontiers, spatial propagation networks (SPNs) [16]–[18] and residual depth learning [19]–[21] have been incorporated into the frameworks. Recently, Rho *et al.* [22] and Zhang *et al.* [23] made use of a pure Transformer or the combination of Transformer and convolutional neural networks (CNNs) to further boost completion performance.

We argue that real-world 3D scenes, *e.g.*, indoor scenarios, typically exhibit a high degree of regularity and proper scene priors should be incorporated into the framework to improve the nature of the solution. Planes are a common representation for modeling geometric prior knowledge of 3D scenes [24]–[26]. Patil *et al.* [27] recently introduced a piece-wise planarity prior and adopted the offset vector field to borrow information from co-planar pixels. However, there is no direct constraint imposed on the offset vector field to help it learn about planar regions in their method. In addition, the planarity prior tends to fail in the high-curvature regions, *e.g.*, bushes, trees and other clutter from the outdoor scenarios, inevitably deteriorating the depth accuracy.

In this paper, we propose novel physics (geometry)-driven deep learning frameworks for monocular depth estimation and completion by assuming that 3D scenes are constituted with piece-wise planes. More specifically, we parametrize the plane representation via surface normal and plane-to-origin distance (the distance from the corresponding plane to the origin, i.e., camera center in our case) and develop a normal-distance head to output pixel-level surface normal and distance. Furthermore, a plane-aware consistency constraint is devised to enforce the surface normal and distance maps to be piece-wise constant. In order to obtain planar regions, we adopt the Felzenszwalb segmentation algorithm [28] for online plane detection utilizing the geometric dissimilarity calculated from surface normal and distance maps. The regularized surface normal and distance maps are then transformed into depth maps. We note that the acquired depth maps are prone to make severe errors in the high-curvature regions owing to the invalidity of the planarity assumption. To account for such failure cases, we integrate a second depth head that is designed in accordance with the regular paradigms. The depth uncertainty is modeled to fully

This work was supported in part by the National Natural Science Foundation of China under grant 61620106012 and in part by the A*STAR-MTC Programmatic Funds under grant M23L7b0021.

Shuwei Shao, Zhongcai Pei, Weihai Chen are with the School of Automation Science and Electrical Engineering, Beihang University, Beijing, China. (email: swshao@buaa.edu.cn, peizc@buaa.edu.cn, whchen@buaa.edu.cn)

Peter C. Y. Chen is with the Department of Mechanical Engineering, National University of Singapore, Singapore. (e-mail: mpechenp@nus.edu.sg)

Zhengguo Li is with the SRO department, Institute for Infocomm Research, A*STAR, Singapore. (ezgli@i2r.a-star.edu.sg)

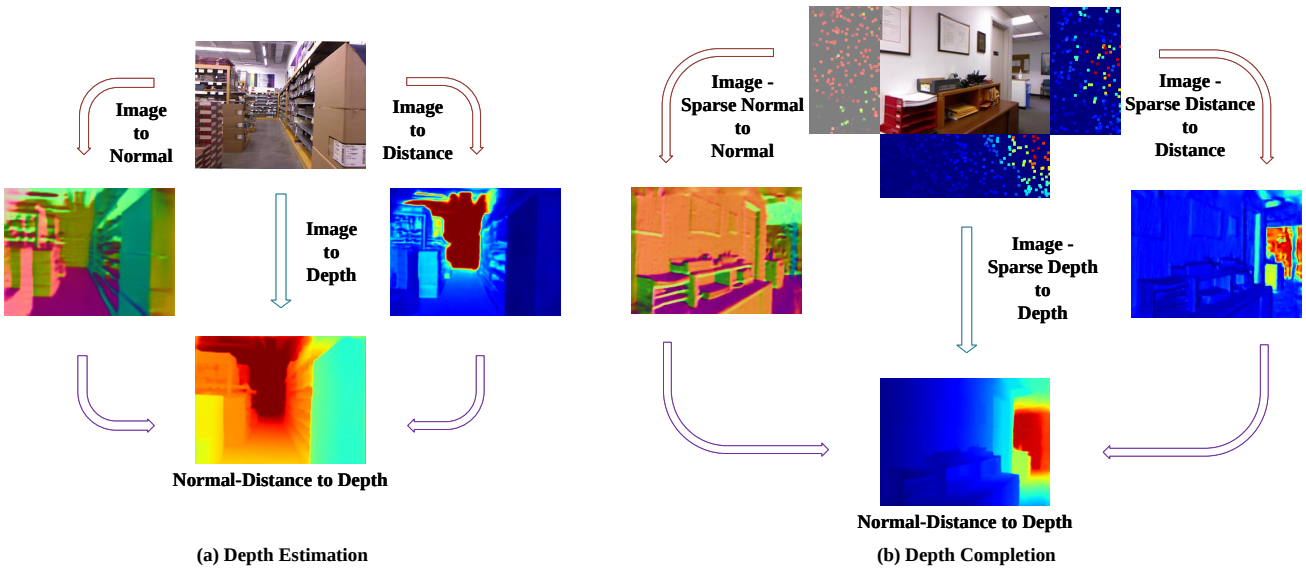

Fig. 1. **A brief illustration of the relationship among RGB image, surface normal map/sparse surface normal map, plane-to-origin distance map/sparse plane-to-origin distance map and depth map/sparse depth map**.

exploit the strengths of these two heads. For depth estimation, the depth maps and uncertainty maps are fed into a developed contrastive iterative refinement module for depth refinement in a complementary manner. In terms of depth completion, the depth maps from these two heads are first fused utilizing the uncertainty maps. Then, the fused depth map and uncertainty map are input into the dedicated non-local spatial propagation network [17] for later depth refinement. The whole pipelines for monocular depth estimation and completion are presented in Figs. 2 and 4, respectively.

The NDDepth approach was first proposed in our previous ICCV 2023 conference paper (Oral presentation) [29]. In this full version, we further extend NDDepth to depth completion. Different from previous approaches that directly complete the sparse depth map, we propose to complete the sparse surface normal and distance maps as intermediate outputs to exploit the geometry of the real-world 3D scenes. Besides, we find that only using the intermediate normal-distance representation can already achieve an impressive performance improvement. It is worth mentioning that while the state-of-the-art performance on the NYU-Depth-v2 [30] has been nearly saturated for quite some time, such as from NLSPN [17] (0.092 RMSE) to CompletionFormer [23] (CFormer, 0.090 RMSE), our NDDepth is capable of improving the CFormer from 0.090 RMSE to 0.081 RMSE, establishing a new record.

To summarize, the contributions of this work are listed as follows:

- We introduce novel physics-driven deep learning frameworks for tasks of monocular depth estimation and completion, which consist of two heads built with asymmetric paradigms. The normal-distance head acquires piece-wise planar depth, while the regular depth head strengthens the framework resilience.
- We develop a plane-aware consistency constraint to encourage the piece-wise constant property of surface normal and plane-to-origin distance maps, and a contrastive iterative refinement module to refine depth in a complementary fashion.
- Extensive experiments verify the efficacy of our designed components. The proposed method outperforms previous state-of-the-art monocular depth estimation and completion competitors on the NYU-Depth-v2 [30], KITTI [31] and SUN RGB-D [32] datasets.

II. Related Work

A. Monocular Depth Estimation

Monocular depth estimation involves the task of predicting the depth map from a single RGB image, which has witnessed dramatic progress over the years, with various methods being developed to tackle this challenge. In the early stages, Saxena *et al.* [33] considered both local and global image features and harnessed the power of a Markov Random Field to regress depth. Eigen *et al.* [7], on the other hand, took a substantial leap forward by introducing CNN into depth estimation, using multi-scale networks to extract vital depth information. As the field evolved, Laina *et al.* [34] employed a fully convolutional network based on residual learning [35] and a reverse Huber loss for optimization. Later, Cao *et al.* [36] and Fu *et al.* [8] took a creative turn by reframing the depth regression problem as classification, making it more tractable. The classification-based depth estimation has inspired extensive follow-up works, for example, Adabins [10] and BinsFormer [37]. [10] observed that the depth distribution varies significantly between different images and proposed to generate adaptive bins in light of the image content. [37] further revisited [10] by disentangling bins and probabilistic representations learning to prevent the global and fine-grained information from defacing each other. Besides, Agarwal *et al.* [38] devised a bin center predictor by utilizing pixel queries at the coarsest level to predict adaptive bins. There are also a series of subsequent studies along the line of depth regression. Lee *et al.* [9] developed several multi-scale guidance layers to connect the intermediate layer features

with the final layer depth prediction. Yang *et al.* [39] pioneered one of the endeavors by making use of the Vision Transformer (ViT) [40] to capture the critical long-distance correlation in depth estimation. Yuan *et al.* [11] selected the conventional path of Conditional Random Fields (CRFs) optimization and designed neural window fully-connected CRFs to reduce the computation complexity. However, the majority of these studies are data-driven approaches while the introduced estimation framework marries deep learning with the fundamental physics of the real-world 3D scenes.

B. Depth Completion

Image-guided depth completion aims to predict the dense depth map from inputs of different modalities that include a sparse depth map and an RGB image. Following the advance of deep learning, depth completion has made significant strides recently. As one of the pioneering works, Ma *et al.* [41], [42] adopted an encoder-decoder architecture to predict the dense output in either a supervised or self-supervised framework. To retain the precise measurements from the sparse depth input while further improving the final depth map quality, Cheng *et al.* [16] introduced the spatial propagation network (SPN) [43] into depth completion. They developed a convolutional spatial propagation network (CSPN) and placed the CSPN behind the encoder-decoder network end to refine its prediction. Building upon CSPN, many follow-up works have sprung up. Cheng *et al.* [44] used learnable convolutional kernel sizes and number of iterations to improve the efficiency. Park *et al.* [17] utilized deformable kernels [45] for propagation to alleviate the mixed-depth problem at object boundaries. Lin *et al.* [18] leveraged independent affinity matrices in each propagation to relax the representation limitation. Zhang *et al.* [23] integrated Transformer and CNN to strengthen the capacity of the encoder-decoder network for better SPN refinement.

In addition to depending only on a single branch, the multi-branch networks have been employed to facilitate multi-modal fusion [12], [20], [46]–[48]. The conventional techniques for fusing multi-modal information involve feature concatenation or elementwise summation operations. Thereafter, more advanced fusion strategies have been introduced. Tang *et al.* [12] applied the guided image filtering [49] while utilizing dynamically generated spatially-variant kernels. Zhong *et al.* [50] used channel-wise canonical correlation analysis. Zhang *et al.* [51] developed a neighbor attention mechanism to merge features at multiple scales. Zhao *et al.* [14] leveraged symmetric gated fusion in the decoder. More recently, Rho *et al.* [22] designed separate Transformer branches to embed sparse depth map and RGB image, and a guided attention mechanism was proposed to capture inter-modal dependencies. Differently, we decouple depth completion into surface normal completion and plane-to-origin distance completion, to leverage the geometry of the real-world 3D scenes.

C. Geometric Constraints for Depth

Traditional methods, such as in multi-view stereo [52] and 3D reconstruction [24], [25], made use of the planarity prior to enable faster optimization and tackle poorly textured surfaces. Recently, Qi *et al.* [53] and Qiu *et al.* [54] introduced surface normal to guide the depth prediction. Xu *et al.* [55] leveraged the depth-normal constraints to refine coarse depth prediction in the plane-to-origin distance subspace. Kusupati *et al.* [56] designed a consistency constraint between spatial depth gradients derived from depth and surface normal. Long *et al.* [57] devised adaptive surface normal to determine the reliable local geometry. Huynh *et al.* [58] and Yin *et al.* [59] introduced non-local coplanarity constraints through either the depth-attention module or the virtual surface normal. Patil *et al.* [27] proposed a piece-wise planarity prior by predicting the offset vector field to deliver information from co-planar pixels. Nevertheless, the offset vector field is not given any direct constraints to aid in its comprehension of planar regions. Moreover, the planarity prior is not always valid, leading to erroneous depth predictions in the high-curvature regions. By contrast, our method explicitly enforces the planar constraint inside each planar region. This effectively avoids the interference of pixels from other planes. Notably, the depth head in our framework allows such failure cases in [27] to be mitigated.

III. Methodology

Real-world 3D scenes typically exhibit a significant degree of regularity, for example, indoor scenarios. As a result, it is reasonable to make the assumption that 3D scenes are constituted with piece-wise planes. In this section, we elaborate on the following parts, which include depth from normal-distance constraint, plane-aware consistency, normal-distance assisted depth estimation and completion.

A. Depth from Normal-Distance Constraint

Let $\mathbf{P} = [X, Y, Z]^{\mathrm{T}}$ denote a 3D point, and $\mathbf{p} = [u, v]^{\mathrm{T}}$ be its 2D projection on the image plane within a planar region of the 3D scenes. The surface normal is the vector originating from $\mathbf{P}$ and perpendicular to the corresponding plane, denoted as $\mathbf{N}(\mathbf{p})$. The distance from the origin (in our case, the camera center) to the corresponding plane is referred to as the plane-to-origin distance, denoted as $\mathcal{D}(\mathbf{p})$. Then, the normal-distance constraint is expressed as

$$\mathbf{N}(\mathbf{p})\,\mathbf{P} = \mathcal{D}(\mathbf{p}). \tag{1}$$

In accordance with the fundamental principles of a pinhole camera, the mathematical representation of the projection from the 3D point $\mathbf{P}$ to the 2D point $\mathbf{p}$ is given by

$$\mathbf{D}(\mathbf{p})\,\widetilde{\mathbf{p}} = \mathbf{K}\mathbf{P}, \tag{2}$$

where $\mathbf{D}(\mathbf{p})$ stands for the depth at $\mathbf{p}$, $\widetilde{\mathbf{p}}$ denotes the homogeneous coordinate of $\mathbf{p}$, and $\mathbf{K}$ denotes the intrinsic matrix.

By further substituting Eq. 2 into Eq. 1, the depth is derived via

$$\mathbf{D}(\mathbf{p}) = \frac{\mathcal{D}(\mathbf{p})}{\mathbf{N}(\mathbf{p})\,\mathbf{K}^{-1}\widetilde{\mathbf{p}}}. \tag{3}$$

Rather than directly estimating the depth map or completing the sparse depth map, we aim to estimate the surface normal and distance maps or complete the sparse surface normal and distance maps as intermediate outputs, and then apply Eq. 3

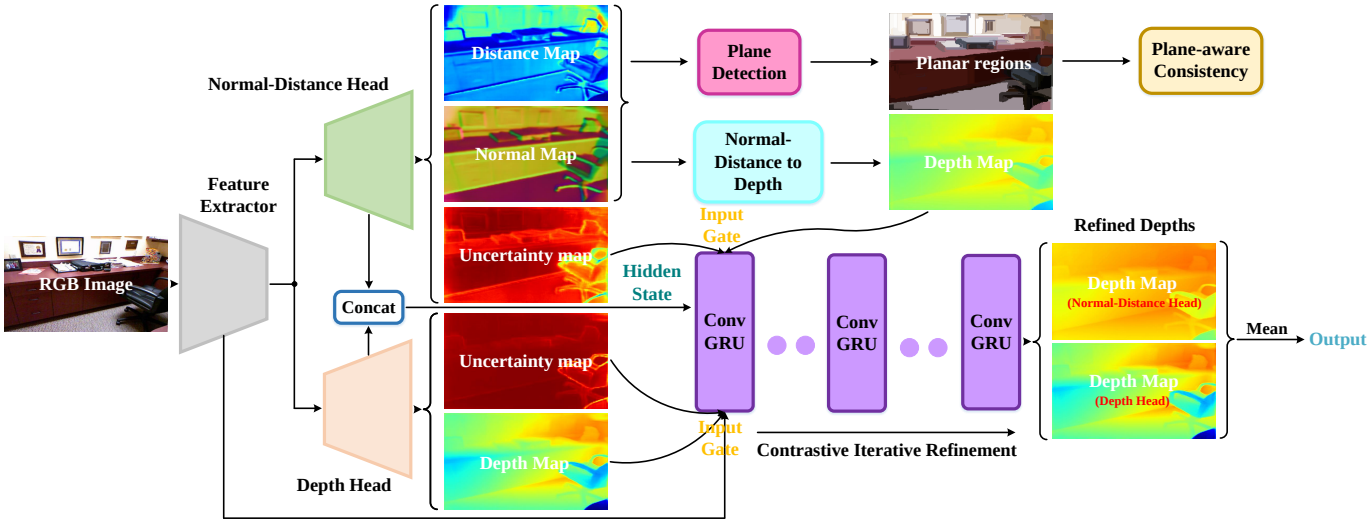

Fig. 2. **Overview of the whole pipeline for monocular depth estimation**. After the contrastive iterative refinement, we acquire two refined depth maps. The upper one is originally from the normal-distance head, and the lower one is originally from the depth head. By taking the mean operation on these two refined depth maps, we obtain the final output. We note that the contrastive iterative refinement occurs at a reduced 1/4 resolution and subsequently, the refined depth maps are upsampled to the full resolution using bilinear interpolation. For the sake of simplicity, we have omitted this particular detail in the flowchart.

to derive the depth map. In contrast to the direct prediction of depth, the intermediate normal-distance representation enjoys the benefit of being piece-wise constant, which is not suitable for depth. This nature allows for the incorporation of a plane-aware consistency constraint that facilitates interactions among pixels, ultimately leading to an enhanced depth prediction.

B. Plane-aware Consistency

Planar region detection. To effectively enforce the plane-aware consistency constraint, it is imperative to identify planar regions. In line with previous works [60]–[63], we leverage the Felzenszwalb segmentation algorithm [28] in our approach.

Let $\mathbf{q}$ be an adjacent pixel to $\mathbf{p}$. We define the dissimilarity in surface normal map between these two pixels as

$$dis_{\mathbf{N}}(\mathbf{p},\mathbf{q}) = \|\mathbf{N}(\mathbf{q}) - \mathbf{N}(\mathbf{p})\|, \tag{4}$$

where $\|\cdot\|$ denotes the Euclidean distance. Suppose $dis_{\mathbf{N}}^{\max}$ and $dis_{\mathbf{N}}^{\min}$ correspond to the maximum and minimum dissimilarities, respectively, among all adjacent pixels, which we use for normalizing the dissimilarity via

$$\overline{dis}_{\mathbf{N}} = \frac{dis_{\mathbf{N}}(\mathbf{p},\mathbf{q}) - dis_{\mathbf{N}}^{\min}}{dis_{\mathbf{N}}^{\max} - dis_{\mathbf{N}}^{\min}}. \tag{5}$$

We define the dissimilarity in plane-to-origin distance map as

$$dis_{\mathcal{D}}(\mathbf{p},\mathbf{q}) = |\mathcal{D}(\mathbf{q}) - \mathcal{D}(\mathbf{p})|. \tag{6}$$

Similarly, we normalize the distance dissimilarity via

$$\overline{dis}_{\mathcal{D}} = \frac{dis_{\mathcal{D}}(\mathbf{p},\mathbf{q}) - dis_{\mathcal{D}}^{\min}}{dis_{\mathcal{D}}^{\max} - dis_{\mathcal{D}}^{\min}}. \tag{7}$$

The normalized surface normal and distance dissimilarities are combined to quantify the geometric dissimilarity,

$$\overline{dis}_{g} = \overline{dis}_{\mathbf{N}} + \overline{dis}_{\mathcal{D}}, \tag{8}$$

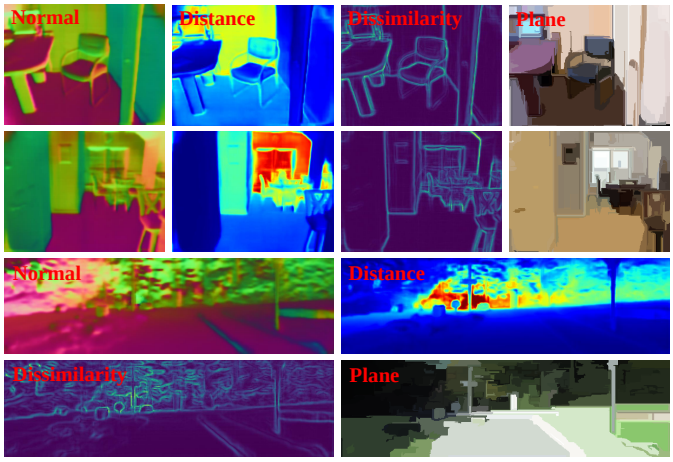

Fig. 3. **Illustration of surface normal, plane-to-origin distance, geometric dissimilarity and extracted planes**. The geometric dissimilarity is calculated from surface normal and plane-to-origin distance. The extracted planes are detected based on geometric dissimilarity using the Felzenszwalb segmentation algorithm [28].

In light of the geometric dissimilarity, we adopt the Felzenszwalb segmentation algorithm to conduct online plane detection. Intuitively, planar regions, *e.g.*, indoor floors and walls, as well as outdoor roads, tend to occupy larger areas. Hence, we focus exclusively on regions exceeding a 200-pixel threshold for selection. In Fig. 3, we provide some qualitative results of surface normal, distance, geometric dissimilarity and detected planes.

Plane-aware consistency loss. After identifying the planar regions, we encourage the plane-aware consistency by imposing penalties on the first-order gradients of surface normal and distance maps inside these planar regions, denoted as $\mathcal{M}(\mathbf{p})$,

$$\mathcal{L}_{pc} = \sum_{\mathbf{p}} \mathcal{M}(\mathbf{p}) |\nabla \mathbf{N}(\mathbf{p})| + \sum_{\mathbf{p}} \mathcal{M}(\mathbf{p}) |\nabla \mathcal{D}(\mathbf{p})|. \tag{9}$$

The initial surface normal and distance predictions may not be accurate. Fortunately, they will gradually improve with every training epoch, resulting in a better detection, and vice versa.

C. Normal-Distance Assisted Depth Estimation

Given an RGB image $\mathbf{R}$, previous methods train a network to directly map the RGB image into a depth prediction $\mathbf{D}$,

$$N_\theta : \mathbf{R} \rightarrow \mathbf{D}. \tag{10}$$

Different from previous studies, our approach uses a network that is equipped with a normal-distance head to predict surface normal and distance maps,

$$N_\theta : \mathbf{R} \rightarrow \mathbf{N}, \mathcal{D}, \tag{11}$$

which are subject to regularization by the plane-aware consistency constraint. The regularized surface normal and distance maps are converted into depth predictions via Eq. 3. However, the converted depth maps do not consistently produce accurate results and are susceptible to large errors in the high curvature regions. We observe that regular estimation paradigms tend to achieve smaller errors in these regions. Therefore, we integrate a second depth head built upon [11] into our framework. To fully capitalize on the strengths of both heads, we incorporate the depth uncertainty modeling. Further, a contrastive iterative refinement module is developed to enhance depth maps in a complementary fashion.

Uncertainty modeling. Identifying regions with large errors is critical before initiating the depth refinement. We model the depth uncertainty in the form of a probability density function of Laplace distribution [64],

$$\begin{aligned} \mathbf{U}_1^{gt}(\mathbf{p}) &= 1 - \exp\left(-\frac{\left|\mathbf{D}_1(\mathbf{p}) - \mathbf{D}^{gt}(\mathbf{p})\right|}{b}\right) \\ \mathbf{U}_2^{gt}(\mathbf{p}) &= 1 - \exp\left(-\frac{\left|\mathbf{D}_2(\mathbf{p}) - \mathbf{D}^{gt}(\mathbf{p})\right|}{b}\right), \end{aligned} \tag{12}$$

where $\mathbf{D}_1(\mathbf{p})$ and $\mathbf{D}_2(\mathbf{p})$ are the depth predictions of normal-distance and depth heads, respectively, $\mathbf{D}^{gt}$ is the ground-truth depth map, b stands for the error tolerance coefficient and is set to 0.2. As $\mathbf{D}^{gt}(\mathbf{p})$ is not accessible at application phase, these two heads also predict uncertainty maps to approximate $\mathbf{U}_1^{gt}$ and $\mathbf{U}_2^{gt}$. We denote the uncertainty predictions as $\mathbf{U}_1(\mathbf{p})$ and $\mathbf{U}_2(\mathbf{p})$, supervised by

$$\mathcal{L}_{\mathbf{U}} = \sum_{\mathbf{p}} \left|\mathbf{U}_1(\mathbf{p}) - \mathbf{U}_1^{gt}(\mathbf{p})\right| + \sum_{\mathbf{p}} \left|\mathbf{U}_2(\mathbf{p}) - \mathbf{U}_2^{gt}(\mathbf{p})\right|. \tag{13}$$

In addition to the uncertainty maps, we determine the complementary regions within these two depth maps by calculating the absolute difference between $\mathbf{D}_1(\mathbf{p})$ and $\mathbf{D}_2(\mathbf{p})$,

$$dif_{\mathbf{D}}(\mathbf{p}) = \left|\mathbf{D}_1(\mathbf{p}) - \mathbf{D}_2(\mathbf{p})\right|, \tag{14}$$

where $dif_{\mathbf{D}}$ is referred to as the complementary map.

Contrastive iterative update. Grounded on the uncertainty maps and complementary map, we refine these two depth maps iteratively. During each iteration, we update $\mathbf{D}_1(\mathbf{p})$ and $\mathbf{D}_2(\mathbf{p})$ as

$$\mathbf{D}_1^{t+1}(\mathbf{p}) \leftarrow \mathbf{D}_1^t(\mathbf{p}) + \Delta\mathbf{D}_1^t(\mathbf{p}), \mathbf{D}_2^{t+1}(\mathbf{p}) \leftarrow \mathbf{D}_2^t(\mathbf{p}) + \Delta\mathbf{D}_2^t(\mathbf{p}), \tag{15}$$

Algorithm 1 Contrastive iterative update

Input: Depth maps $\mathbf{D}_1$, $\mathbf{D}_2$; Uncertainty maps $\mathbf{U}_1$, $\mathbf{U}_2$; Complementary map $dif_{\mathbf{D}}$; Contextual features.
Output: Depth maps $\mathbf{D}_1^{final}$, $\mathbf{D}_2^{final}$.

```
for t = 1, ..., maximum iteration step m do
    Projected features = Conv(D_1^t, D_2^t, U_1, U_2, dif_D);
    Aggregated features = Conv(Projected features);
    I^t = Concat(Aggregated features, Contextual features);
    h^{t+1} = ConvGRU(h^t, I^t);
    ΔD_1^t, ΔD_2^t = Conv(h^{t+1});
    D_1^{t+1} = D_1^t + ΔD_1^t; D_2^{t+1} = D_2^t + ΔD_2^t;
end for
D_1^final = D_1^{m+1}; D_2^final = D_2^{m+1}.
```

where $\Delta\mathbf{D}_1^t(\mathbf{p})$ and $\Delta\mathbf{D}_2^t(\mathbf{p})$ are the depth updates at iteration t. Drawing inspiration from [65], we leverage a convolutional gated recurrent unit (ConvGRU) [66] to produce these updates. The ConvGRU is capable of preserving past states and effectively leveraging the temporal context information during the refinement process.

To begin, we employ two convolutional layers to transform $\mathbf{D}_1^t$, $\mathbf{D}_2^t$, $\mathbf{U}_1$, $\mathbf{U}_2$ and $dif_{\mathbf{D}}$ into the feature space, respectively, and employ one convolutional layer to aggregate the projected features. Then, we concatenate these aggregated features with the contextual features (1/4 resolution output from the feature extractor) as input $\mathbf{I}^t$. The structure inside ConvGRU is shown as follows,

$$\mathbf{z}^{t+1} = \sigma\left(Conv_{5\times5}\left(\left[\mathbf{h}^t, \mathbf{I}^t\right]\right)\right), \tag{16}$$

$$\mathbf{r}^{t+1} = \sigma\left(Conv_{5\times5}\left(\left[\mathbf{h}^t, \mathbf{I}^t\right]\right)\right), \tag{17}$$

$$\widehat{\mathbf{h}}^{t+1} = \tanh\left(Conv_{5\times5}\left(\left[\mathbf{r}^{t+1} \odot \mathbf{h}^t, \mathbf{I}^t\right]\right)\right), \tag{18}$$

$$\mathbf{h}^{t+1} = \left(1 - \mathbf{z}^{t+1}\right) \odot \mathbf{h}^t + \mathbf{z}^{t+1} \odot \widehat{\mathbf{h}}^{t+1}, \tag{19}$$

where $\mathbf{z}$ is the update gate, $\mathbf{r}$ is the reset gate, $Conv_{5\times5}$ is the separable 5×5 convolution, $\odot$ denotes the element-wise multiplication, σ and tanh denote the sigmoid and tanh activation functions, respectively. The hidden state $\mathbf{h}^t$ is initialized using the penultimate layer feature maps from both normal-distance and depth heads, concatenated together and activated with the tanh function. Finally, we employ two convolutional layers to estimate depth updates $\Delta\mathbf{D}_1^t$ and $\Delta\mathbf{D}_2^t$ from $\mathbf{h}^{t+1}$, and utilize Eq.15 to acquire updated depth maps $\mathbf{D}_1^{t+1}$ and $\mathbf{D}_2^{t+1}$.

Utilizing this refinement module, the depth maps undergo an iterative enhancement until they ultimately reach convergence,

$$\mathbf{D}_1^{final}(\mathbf{p}) \leftarrow \mathbf{D}_1^t(\mathbf{p}), \mathbf{D}_2^{final}(\mathbf{p}) \leftarrow \mathbf{D}_2^t(\mathbf{p}). \tag{20}$$

In Algorithm 1, we present a summarization of the above contrastive iterative update process.

We further upsample $\mathbf{D}_1^{final}$ and $\mathbf{D}_2^{final}$ to the full resolution, followed by the mean operation to obtain the final output,

$$\mathbf{D}^{final}(\mathbf{p}) = 0.5(\mathbf{D}_1^{final}(\mathbf{p}) + \mathbf{D}_2^{final}(\mathbf{p})). \tag{21}$$

Notably, there is little difference in performance utilizing simple averaging or uncertainty-based fusion for $\mathbf{D}_1^{final}(\mathbf{p})$ and

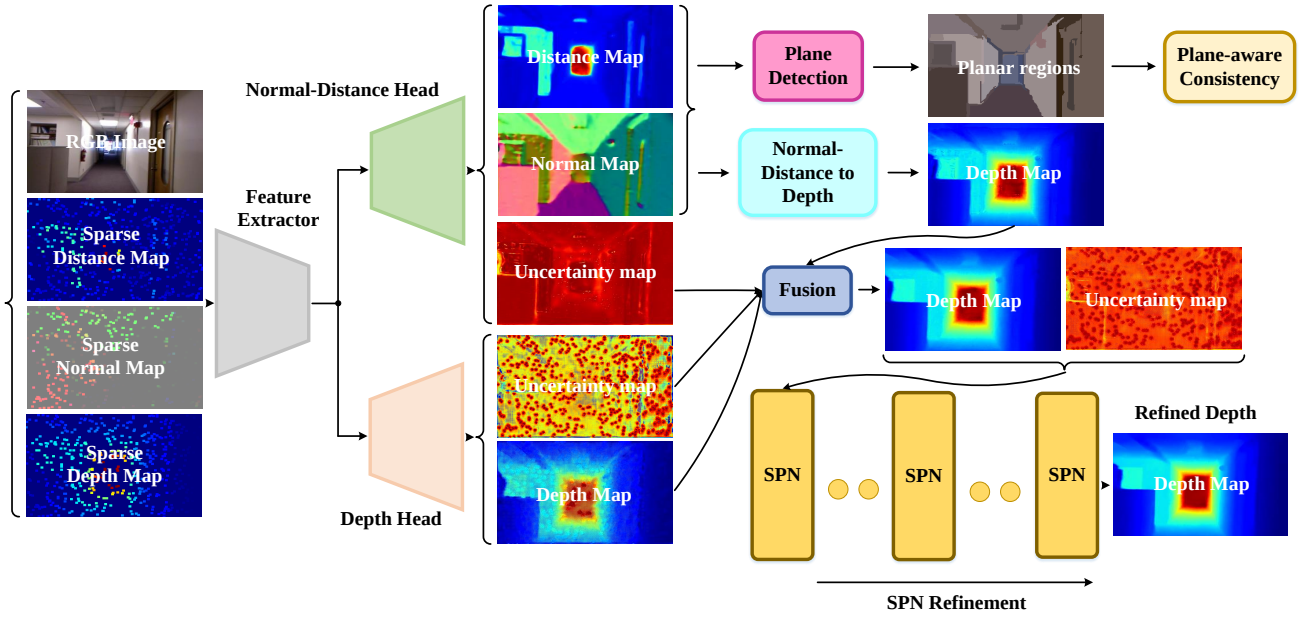

Fig. 4. **Overview of the whole pipeline for depth completion**. Note that apart from the depth and uncertainty maps, the fused affinity map is also fed into the SPN to assist refinement. For simplicity, we have omitted this detail in the flowchart.

$\mathbf{D}_2^{final}(\mathbf{p})$. This is reasonable since $\mathbf{D}_1^{final}(\mathbf{p})$ and $\mathbf{D}_2^{final}(\mathbf{p})$ are the final refined results according to the uncertainty maps.

The training optimization objectives are detailed as follows:

Depth loss. We adopt a scaled Scale-Invariant loss for depth supervision [9],

$$\mathcal{L}_{\mathbf{D}} = \sum_{t=1}^{m+1} \gamma^{m+1-t} \begin{pmatrix} \kappa \sqrt{\frac{1}{|\mathbf{T}|}\sum_{\mathbf{p}} \left(\mathbf{g}_1^t(\mathbf{p})\right)^2 - \frac{\eta}{|\mathbf{T}|^2}\left(\sum_{\mathbf{p}} \mathbf{g}_1^t(\mathbf{p})\right)^2} + \\ \kappa \sqrt{\frac{1}{|\mathbf{T}|}\sum_{\mathbf{p}} \left(\mathbf{g}_2^t(\mathbf{p})\right)^2 - \frac{\eta}{|\mathbf{T}|^2}\left(\sum_{\mathbf{p}} \mathbf{g}_2^t(\mathbf{p})\right)^2} \end{pmatrix}, \quad (22)$$

where γ is the decay factor and m denotes the maximum iteration step, set to 0.85 and 3, respectively, $\mathbf{g}(\mathbf{p}) = \log \mathbf{D}(\mathbf{p}) - \log \mathbf{D}^{gt}(\mathbf{p})$, $\mathbf{T}$ denotes a set of pixels containing valid values, κ and η are set to 10 and 0.85 following [9].

Normal loss. We adopt a negative cosine loss to supervise surface normal [67],

$$\mathcal{L}_{\mathbf{N}} = \frac{1}{|\mathbf{T}|}\sum_{\mathbf{p}} 1 - \mathbf{N}(\mathbf{p})\,\mathbf{N}^{gt\mathrm{T}}(\mathbf{p}), \quad (23)$$

Distance loss. We adopt an L1 loss to supervise plane-to-origin distance,

$$\mathcal{L}_{\mathcal{D}} = \frac{1}{|\mathbf{T}|}\sum_{\mathbf{p}} \left|\mathcal{D}(\mathbf{p}) - \mathcal{D}^{gt}(\mathbf{p})\right|. \quad (24)$$

Due to the lack of surface normal and distance ground-truths in the NYU-Depth-v2 and KITTI datasets, we follow [54] to acquire the ground-truth of surface normal from depth ground-truth. Then, the distance ground-truth is obtained by $\mathcal{D}^{gt}(\mathbf{p}) = \mathbf{D}(\mathbf{p})^{gt}\,\mathbf{N}(\mathbf{p})^{gt}\,\mathbf{K}^{-1}\tilde{\mathbf{p}}$.

The **overall loss** is the combination of $\mathcal{L}_{\mathbf{D}}$, $\mathcal{L}_{\mathbf{N}}$, $\mathcal{L}_{\mathcal{D}}$, $\mathcal{L}_{\mathbf{U}}$ and $\mathcal{L}_{pc}$,

$$\mathcal{L}_{overall=}\lambda_1\mathcal{L}_{\mathbf{D}} + \lambda_2\mathcal{L}_{\mathbf{N}} + \lambda_3\mathcal{L}_{\mathcal{D}} + \lambda_4\mathcal{L}_{\mathbf{U}} + \lambda_5\mathcal{L}_{pc}, \quad (25)$$

where λ_1, λ_2, λ_3, λ_4 and λ_5 are empirically set to 1, 5, 0.25, 1 and 0.01, respectively.

Network architecture. We adopt Swin-L [68] as the feature extractor, unless specified otherwise. As for the KITTI official split with more training data, we choose SwinV2-L [69] with a larger window size as the feature extractor to capture strong representations. The normal-distance head and the depth head share the same architecture except for the final prediction layer. These parts with the same architecture are designed according to NeWCRFs [11]. The normal-distance head predicts surface normal, distance and uncertainty maps, while the depth head predicts depth and uncertainty maps.

D. Normal-Distance Assisted Depth Completion

Given a sparse depth map $\mathbf{D}^{sp}$ and an RGB image $\mathbf{R}$, prior methods train a network to directly complete the sparse depth map into a dense one $\mathbf{D}$ under the guidance of RGB image,

$$N_\theta : \mathbf{D}^{sp}, \mathbf{R} \rightarrow \mathbf{D}. \quad (26)$$

Unlike previous methods, our approach completes the sparse surface normal $\mathbf{N}^{sp}$ and distance $\mathcal{D}^{sp}$,

$$N_\theta : \mathbf{N}^{sp}, \mathcal{D}^{sp}, \mathbf{R} \rightarrow \mathbf{N}, \mathcal{D}. \quad (27)$$

The surface normal and distance maps are regularized through the plane-aware consistency constraint and then converted into depth predictions using Eq. 3. Similar to the estimation framework, we integrate a regular depth head based on CFormer [23] to improve the robustness of completion framework. Then, we fuse the depth maps from the normal-distance head and depth head in light of the uncertainty maps. The depth uncertainty is modeled in the same fashion as the estimation framework. Here we leverage a different fusion manner from monocular depth estimation. This is due to the fact that the depth maps

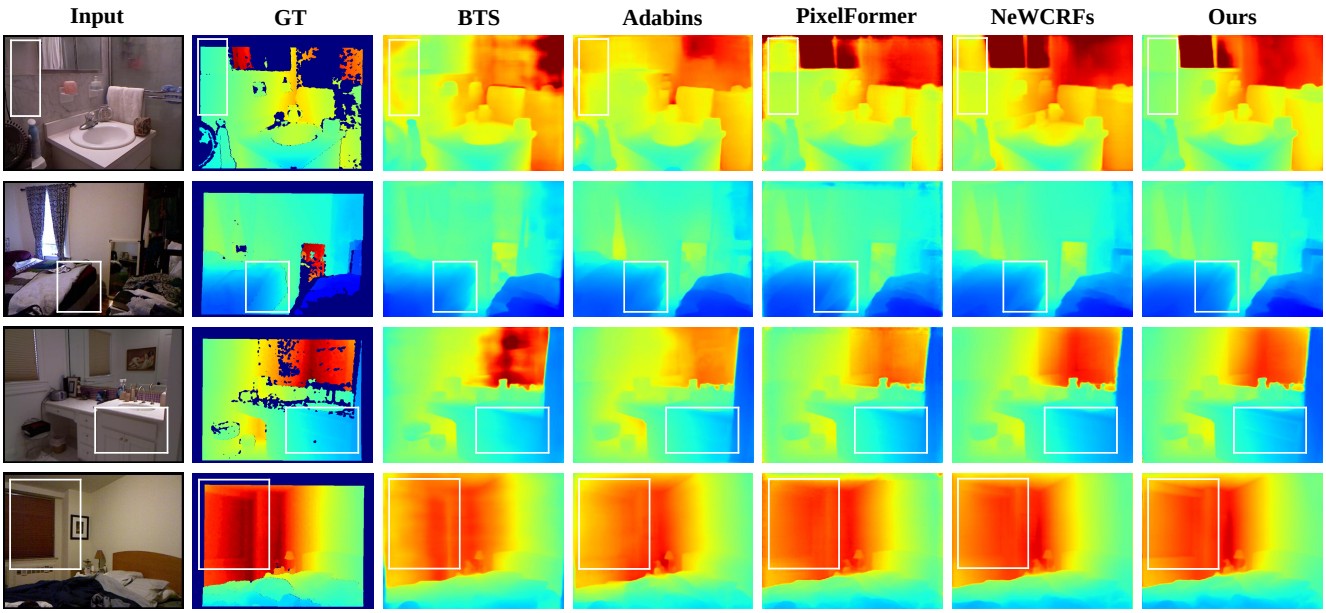

Fig. 5. **Qualitative depth estimation results on the NYU-Depth-v2 dataset**. The white boxes show the regions to focus on.

employed for fusion in depth completion are initial predictions without refinement, while the depth maps employed for fusion in monocular depth estimation undergo refinement according to uncertainty maps. The uncertainty-based fusion process is mathematically described as

$$\mathbf{W}_1(\mathbf{p}) = \frac{1-\mathbf{U}_1(\mathbf{p})}{2-\mathbf{U}_1(\mathbf{p})-\mathbf{U}_2(\mathbf{p})}, \tag{28}$$

$$\mathbf{W}_2(\mathbf{p}) = \frac{1-\mathbf{U}_2(\mathbf{p})}{2-\mathbf{U}_1(\mathbf{p})-\mathbf{U}_2(\mathbf{p})}, \tag{29}$$

$$\mathbf{D}(\mathbf{p}) = \mathbf{W}_1(\mathbf{p})\,\mathbf{D}_1(\mathbf{p}) + \mathbf{W}_2(\mathbf{p})\,\mathbf{D}_2(\mathbf{p}), \tag{30}$$

$$\mathbf{U}(\mathbf{p}) = \mathbf{W}_1(\mathbf{p})\,\mathbf{U}_1(\mathbf{p}) + \mathbf{W}_2(\mathbf{p})\,\mathbf{U}_2(\mathbf{p}). \tag{31}$$

The fused depth and uncertainty maps are then input into the non-local spatial propagation network (NLSPN) [17] for depth refinement.

SPN refinement. The spatial propagation of $\mathbf{D}^t(\mathbf{p})$ at iteration t using its non-local neighbors $\Omega(\mathbf{p})$ is defined as

$$\mathbf{D}^{t+1}(\mathbf{p}) = \mathbf{W}_{\mathbf{p}}(\mathbf{p})\,\mathbf{D}^t(\mathbf{p}) + \sum_{\mathbf{p}_{nei}\in\Omega(\mathbf{p})} \mathbf{W}_{\mathbf{p}_{nei}}(\mathbf{p})\,\mathbf{D}^t(\mathbf{p}_{nei}), \tag{32}$$

where $\mathbf{W}_{\mathbf{p}_{nei}}(\mathbf{p})$ denotes the affinity weight between the target pixel $\mathbf{p}$ and its neighbor pixel $\mathbf{p}_{nei}$, with value inside $(-1, 1)$, $\mathbf{W}_{\mathbf{p}}(\mathbf{p}) = 1 - \sum_{\mathbf{p}_{nei}\in\Omega(\mathbf{p})} \mathbf{W}_{\mathbf{p}_{nei}}(\mathbf{p})$ and signifies the degree to which the original depth will be conserved. Here, our normal-distance head and depth head also predict the affinity maps, respectively, which are fused in the same fusion manner as the depth and uncertainty maps for SPN refinement. Additionally, the affinity map is modulated by the uncertainty map to reduce the negative impact of erroneous depth regions. Following m iterations of spatial propagation, we obtain the final prediction $\mathbf{D}^{final}(\mathbf{p})$, where m is set to 6 based on [23].

The training optimization objectives are detailed as follows:

Depth loss. We adopt a combined L1 and L2 loss for depth supervision [17],

$$\mathcal{L}_{\mathbf{D}} = \frac{1}{|\mathbf{T}|}\sum_{\mathbf{p}} \left(\left|\mathbf{D}^{final}(\mathbf{p}) - \mathbf{D}^{gt}(\mathbf{p})\right| + \left|\mathbf{D}^{final}(\mathbf{p}) - \mathbf{D}^{gt}(\mathbf{p})\right|^2 \right). \tag{33}$$

We note that only the final output is supervised following [17], [23].

The normal loss, distance loss, uncertainty loss and plane-aware consistency loss are same to the estimation framework.

The **overall loss** is thus summarized as

$$\mathcal{L}_{overall=}\lambda_1\mathcal{L}_{\mathbf{D}} + \lambda_2\mathcal{L}_{\mathbf{N}} + \lambda_3\mathcal{L}_{\mathcal{D}} + \lambda_4\mathcal{L}_{\mathbf{U}} + \lambda_5\mathcal{L}_{pc}, \tag{34}$$

where λ_1, λ_2, λ_3, λ_4 and λ_5 are empirically set to 1, 5, 0.25, 1 and 0.01, respectively.

Network architecture. We make slight modifications to the encoder developed in CFormer [23] as the feature extractor. It receives a sparse surface normal map, a sparse plane-to-origin distance map, a sparse depth map and the corresponding RGB image. The design of normal-distance and depth heads mainly follows the decoder of CFormer. The former predicts surface normal, distance, uncertainty and affinity maps, while the latter predicts depth, uncertainty and affinity maps.

IV. Experiment

A. Datasets and Evaluation Metrics

NYU-Depth-v2 dataset comprises data from indoor scenes, providing images and ground-truth depth maps with a resolution of 640×480 pixels. Regarding monocular depth estimation, we leverage 36253 images for training and 654 images for testing in line with [11], [38]. The training and test resolution is 640×480. For depth completion, we employ 50000 images for training and 654 images for testing following [17], [23]. Moreover, the images and ground-truth depth maps are half-downsampled and center-cropped to a resolution of 304×288 during training and testing. The sparse depth, surface normal

Method	Cap	Abs Rel ↓	Sq Rel ↓	RMSE ↓	$\log_{10}$ ↓	$\delta_{1.25}$ ↑	$\delta_{1.25^2}$ ↑	$\delta_{1.25^3}$ ↑
Eigen *et al.* [7]	0-10m	0.158	-	0.641	-	0.769	0.950	0.988
Fu *et al.* [8]	0-10m	0.115	-	0.509	0.051	0.828	0.965	0.992
Qi et al. [53]	0-10m	0.128	-	0.569	0.057	0.834	0.960	0.990
VNL [59]	0-10m	0.108	-	0.416	0.048	0.875	0.976	0.994
BTS [9]	0-10m	0.113	0.066	0.407	0.049	0.871	0.977	0.995
Zhang *et al.* [70]	0-10m	0.112	-	0.447	0.048	0.881	0.979	0.996
DAV [58]	0-10m	0.108	-	0.412	-	0.882	0.980	0.996
PWA [71]	0-10m	0.105	-	0.374	0.045	0.892	0.985	0.997
Long *et al.* [57]	0-10m	0.101	-	0.377	0.044	0.890	0.982	0.996
TransDepth [39]	0-10m	0.106	-	0.365	0.045	0.900	0.983	0.996
DPT [72]	0-10m	0.110	-	0.367	0.045	0.904	0.988	**0.998**
Adabins [10]	0-10m	0.103	-	0.364	0.044	0.903	0.984	0.997
P3Depth [27]	0-10m	0.104	-	0.356	0.043	0.898	0.981	0.996
Localbins [73]	0-10m	0.099	-	0.357	0.042	0.907	0.987	**0.998**
DepthFormer [74]	0-10m	0.096	-	0.339	0.041	0.921	0.989	**0.998**
NeWCRFs [11]	0-10m	0.095	0.045	0.334	0.041	0.922	**0.992**	**0.998**
PixelFormer [38]	0-10m	0.090	-	0.322	0.039	0.929	0.991	**0.998**
Ours	0-10m	**0.087**	**0.041**	**0.311**	**0.038**	**0.936**	0.991	**0.998**

TABLE I
QUANTITATIVE DEPTH ESTIMATION COMPARISON ON THE NYU-DEPTH-V2 DATASET. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD.

and distance maps are generated by randomly sampling from the corresponding ground-truths, same to the sampling manner in [17], [23].

KITTI dataset consists of data from outdoor scenes, providing stereo images, sparse and ground-truth depth maps. The resolution is around 1241×376 pixels. As for monocular depth estimation, we employ two prevalent data splits, the Eigen split with 23488 training images and 697 test images [7], as well as the official split with 85898 training images, 1000 validation images and 500 test images. The resolutions for training and testing are 1120×352 and 1216×352, respectively. Regarding depth completion, we use the official split with 85898 training images, 1000 validation images and 1000 test images. Due to the absence of LiDAR returns at the top of the depth map, the inputs undergo a bottom center-cropping operation for training based on [17], [23], resulting in a resolution of 1216×240. The sparse surface normal and distance maps are calculated from sparse depth maps in the same way as ground-truth acquisition. The ground-truth depth maps associated with the official test set are not available and the results are generated by the online server.

SUN RGB-D dataset contains approximately 10K images, which are captured by four sensors in indoor scenes. We adopt this dataset to validate the generalization ability of monocular depth estimation in a challenging zero-shot setup, and evaluate pre-trained models on the official 5050 test images.

Evaluation metrics. Following [11], [38], we adopt metrics SILog, Abs Rel, Sq Rel, RMSE, iRMSE, RMSE log, $\log_{10}$, $\delta_{1.25}$, $\delta_{1.25^2}$ and $\delta_{1.25^3}$ for monocular depth estimation. In line with [17], we adopt metrics REL, RMSE, iRMSE, MAE, iMAE, $\delta_{1.25}$, $\delta_{1.25^2}$ and $\delta_{1.25^3}$ for depth completion. Besides, we adopt metrics Mean, 11.25°, 22.5° and 30° [62] for surface normal comparison.

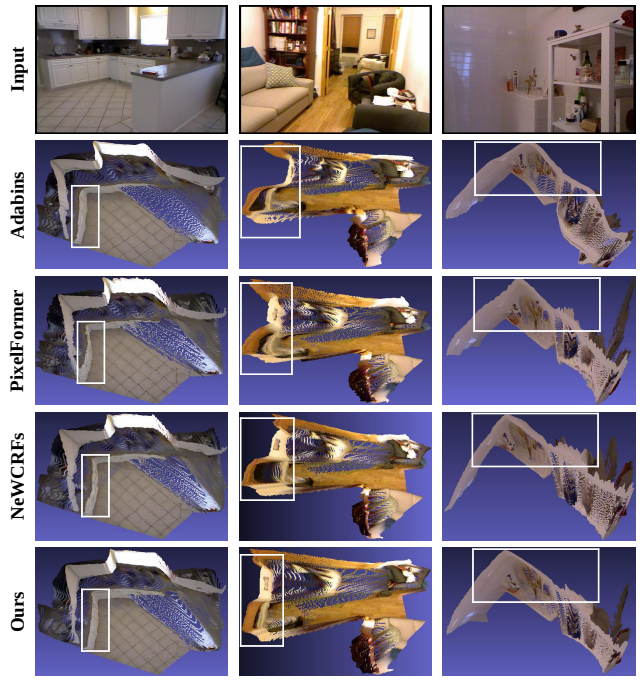

Fig. 6. **Qualitative results of point cloud for monocular depth estimation on the NYU-Depth-v2 dataset**. The point clouds are captured from the top view.

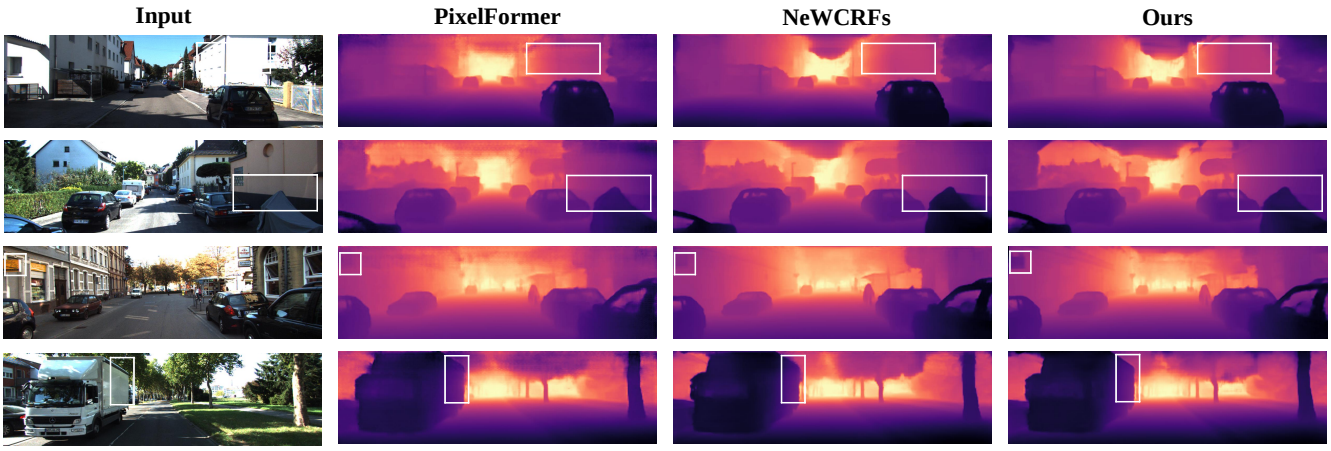

Fig. 7. **Qualitative depth estimation results on the Eigen split of KITTI dataset**.

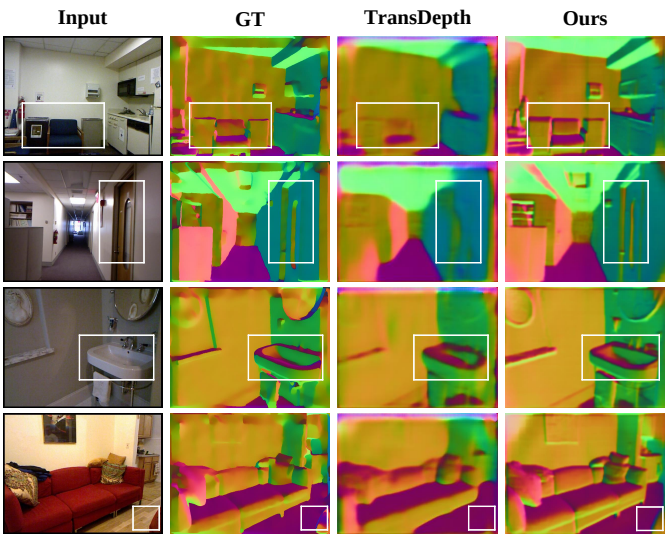

Fig. 8. **Qualitative surface normal results on the NYU-Depth-v2 dataset**.

Method	Mean ↓	11.25° ↑	22.5° ↑	30° ↑
Surface normal predicted from the network				
3DP [78]	33.0	18.8	40.7	52.4
Ladicky *et al.* [79]	35.5	24.0	45.6	55.9
Wang *et al.* [80]	28.8	35.2	57.1	65.5
Eigen *et al.* [67]	23.7	39.2	62.0	71.1
GeoNet [53]	19.0	48.4	71.5	79.5
GeoNet++ [81]	18.5	50.2	73.2	80.7
TransDepth [39]	20.8	48.2	70.0	77.2
Ours	**17.0**	**53.6**	**76.7**	**83.9**
Surface normal derived from the depth				
DORN [8]	36.6	15.7	36.5	49.4
GeoNet [53]	36.8	15.0	34.5	46.7
P^2Net [62]	36.1	15.6	37.7	50.0
GeoNet++ [81]	28.3	30.1	54.7	65.5
ASN‡ [57]	**20.0**	**43.5**	**69.1**	**78.6**
PixelFormer [38]	35.2	17.6	40.4	52.8
NeWCRFs [11]	30.3	24.0	50.7	62.9
Ours	26.9	32.0	58.3	68.9

TABLE II
QUANTITATIVE SURFACE NORMAL COMPARISON ON THE NYU-DEPTH-V2 DATASET. ‡ INDICATES THAT THE METHOD USES SURFACE NORMAL GROUND-TRUTH TO DIRECTLY SUPERVISE THE NORMAL DERIVED FROM DEPTH.

B. Implementation Details

We implement our frameworks in PyTorch [75] on NVIDIA RTX A5000 GPUs. For monocular depth estimation, we adopt the Adam optimizer [76] where β_1 and β_2 are set to 0.9 and 0.999, respectively, along with a batch size of 8. The learning rate is scheduled using polynomial decay, starting from a base value of 2e-5 and decreasing to 2e-6. The total epochs are 25. For depth completion, we use the AdamW optimizer [77] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The batch size and initial learning rate are set to 12 and 0.001, respectively. The model is trained for 72 epochs on the NYU-Depth-v2 dataset, with the learning rate being reduced by half at epochs 36, 48, and 60. As for the KITTI dataset, the model undergoes training for 100 epochs and we apply the learning rate decay by a factor of 0.5 at epochs 50, 60, 70, 80, and 90.

C. Monocular Depth Estimation

Comparison to previous competitors. We first evaluate the proposed method on the NYU-Depth-v2 dataset. The results are summarized in Table I. As we can see, our method exceeds previous competing methods on most metrics, with particularly notable results in the Abs Rel and RMSE. Fig. 5 provides qualitative depth comparison results, where our method excels in delineating planar regions while preserving local details, *e.g.*, boundaries. Besides, we present qualitative point cloud results in Fig. 6, which are obtained using depth maps. In particular, our point clouds preserve prominent geometric features such as planes and have fewer distortions compared to those generated by alternative methods.

To further evaluate the geometric properties, we summarize the surface normal results in Table II, divided into two cases: surface normal predicted from the network and surface normal derived from the depth. For the latter case, we generate normal

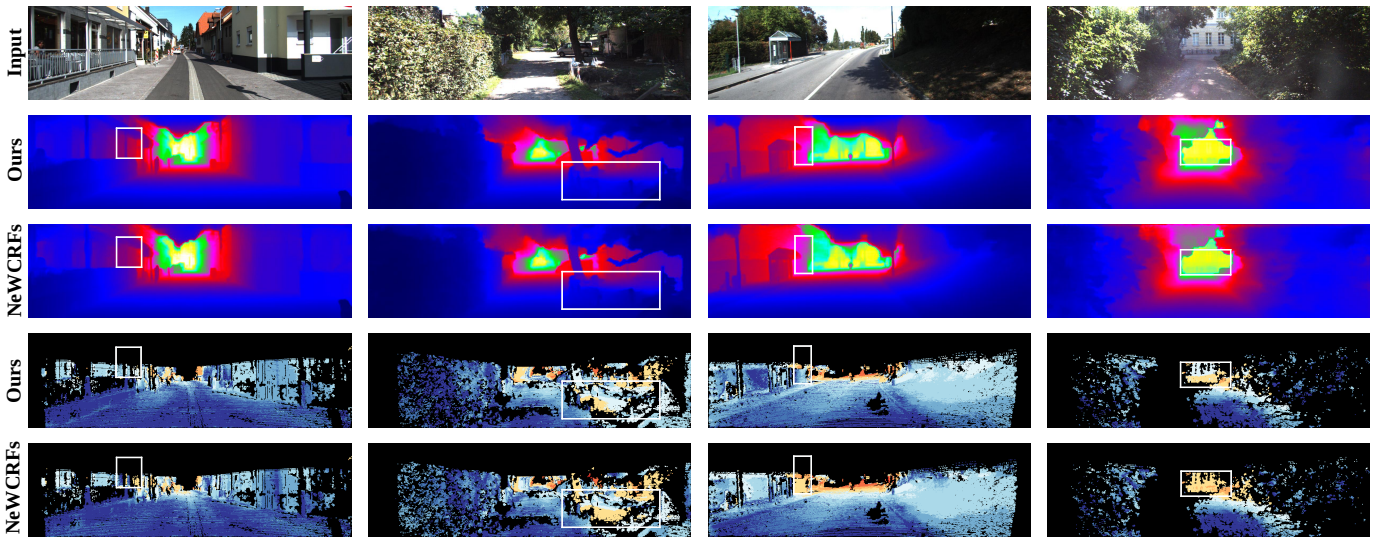

Fig. 9. **Qualitative depth estimation results on the official split of KITTI dataset**. The second and third rows correspond to the depth maps, while the fourth and fifth rows represent the error maps.

Method	Cap	Abs Rel ↓	Sq Rel ↓	RMSE ↓	RMSE log ↓	$\delta_{1.25}$ ↑	$\delta_{1.25^2}$ ↑	$\delta_{1.25^3}$ ↑
Eigen *et al.* [67]	0-80m	0.203	1.548	6.307	0.282	0.702	0.898	0.967
Gan *et al.* [82]	0-80m	0.098	0.666	3.933	0.173	0.890	0.984	0.985
Fu *et al.* [8]	0-80m	0.072	0.307	2.727	0.120	0.932	0.984	0.994
VNL [59]	0-80m	0.072	-	3.258	0.117	0.938	0.990	0.998
BTS [9]	0-80m	0.061	0.261	2.834	0.099	0.954	0.992	0.998
Zhang *et al.* [70]	0-80m	0.064	0.265	3.084	0.106	0.952	0.993	0.998
PWA [71]	0-80m	0.060	0.221	2.604	0.093	0.958	0.994	**0.999**
PackNet-SAN [83]	0-80m	0.062	-	2.888	-	0.955	-	-
TransDepth [39]	0-80m	0.064	0.252	2.755	0.098	0.956	0.994	**0.999**
DPT [72]	0-80m	0.060	-	2.573	0.092	0.959	0.995	0.996
Adabins [10]	0-80m	0.058	0.190	2.360	0.088	0.964	0.995	**0.999**
P3Depth [27]	0-80m	0.071	0.270	2.842	0.103	0.953	0.993	0.998
DepthFormer [74]	0-80m	0.052	0.158	2.143	0.079	0.975	0.997	**0.999**
NeWCRFs [11]	0-80m	0.052	0.155	2.129	0.079	0.974	0.997	**0.999**
PixelFormer [38]	0-80m	0.051	0.149	2.081	0.077	0.976	0.997	**0.999**
Ours	0-80m	**0.050**	**0.141**	**2.025**	**0.075**	**0.978**	**0.998**	**0.999**
Fu *et al.* [8]	0-50m	0.071	0.268	2.271	0.116	0.936	0.985	0.995
BTS [9]	0-50m	0.058	0.183	1.995	0.090	0.962	0.994	0.999
Zhang *et al.* [70]	0-50m	0.061	0.200	2.283	0.099	0.960	0.995	0.999
PWA [71]	0-50m	0.057	0.161	1.872	0.087	0.965	0.995	0.999
P3Depth [27]	0-50m	0.055	0.130	1.651	0.081	0.974	0.997	0.999
Ours	0-50m	**0.048**	**0.107**	**1.513**	**0.071**	**0.981**	**0.998**	**1.000**

TABLE III
QUANTITATIVE DEPTH ESTIMATION COMPARISON ON THE EIGEN SPLIT OF KITTI DATASET. "-" MEANS NOT APPLICABLE.

from depth leveraging a point-to-normal layer [84], [85]. It can be seen that our method exceeds the compared approaches in the former case and demonstrates comparable performance to the ASN in the latter case, even though ASN employs surface normal ground-truth to directly supervise the normal generated from depth. Fig. 8 presents qualitative surface normal results. In contrast to the TransDepth, the proposed approach captures fine-grained local details and its surface normal maps are better aligned to the ground-truth surface normal maps.

We then evaluate the proposed method on the KITTI dataset.

Method	SILog ↓	Sq Rel ↓	Abs Rel ↓	iRMSE↓
PAP [86]	13.08	10.27	2.72	13.95
P3Depth [27]	12.82	9.92	2.53	13.71
VNL [59]	12.65	10.15	2.46	13.02
Fu *et al.* [8]	11.77	8.78	2.23	12.98
BTS [9]	11.67	9.04	2.21	12.23
BA-Full [87]	11.61	9.38	2.29	12.23
PackNet-SAN [83]	11.54	9.12	2.35	12.38
PWA [71]	11.45	9.05	2.30	12.32
DepthFormer [74]	10.69	8.68	1.84	11.39
NeWCRFs [11]	10.39	8.37	1.83	11.03
PixelFormer [38]	10.28	8.16	1.82	10.84
Ours	**9.62**	**7.75**	**1.59**	**10.62**

TABLE IV
QUANTITATIVE DEPTH ESTIMATION COMPARISON ON THE OFFICIAL SPLIT OF KITTI DATASET. THE SILOG IS THE MAIN RANKING METRIC. OUR METHOD RANKS 1ST AMONG ALL SUBMISSIONS ON THE KITTI DEPTH PREDICTION ONLINE BENCHMARK AT THE SUBMISSION TIME.

Method	Abs Rel ↓	RMSE ↓	$\log_{10}$ ↓	$\delta_{1.25}$ ↑
Chen *et al.* [88]	0.166	0.494	0.071	0.757
VNL [59]	0.183	0.541	0.082	0.696
BTS [9]	0.172	0.515	0.075	0.740
Adabins [10]	0.159	0.476	0.068	0.771
P3Depth [27]	0.178	0.541	-	0.698
Localbins [73]	0.156	0.470	0.067	0.777
NeWCRFs [11]	0.151	0.424	0.064	0.798
PixelFormer [38]	0.144	0.441	0.062	0.802
Ours	**0.137**	**0.411**	**0.060**	**0.820**

TABLE V
GENERALIZATION COMPARISON ON THE SUN RGB-D DATASET.

Setting	PC	CIR	Abs Rel ↓	RMSE ↓	$\log_{10}$ ↓	$\delta_{1.25}$ ↑
D			0.095	0.334	0.041	0.922
N$\mathcal{D}$			0.092	0.324	0.039	0.926
N$\mathcal{D}$	✓		0.089	0.318	0.039	0.929
D&N$\mathcal{D}$	✓		0.088	0.315	0.038	0.931
D&N$\mathcal{D}$	✓	✓	**0.087**	**0.311**	**0.038**	**0.936**
D (planar)			0.094	0.316	0.040	0.925
N$\mathcal{D}$ (planar)	✓		0.088	0.301	0.038	0.934
D (non-planar)			0.102	0.390	0.043	0.909
N$\mathcal{D}$ (non-planar)	✓		0.104	0.394	0.044	0.908

TABLE VI
ABLATION STUDY ON THE WHOLE FRAMEWORK FOR MONOCULAR DEPTH ESTIMATION. D&N$\mathcal{D}$: INTEGRATED DEPTH AND NORMAL-DISTANCE HEADS; D: DEPTH HEAD; N$\mathcal{D}$: NORMAL-DISTANCE HEAD; PC: PLANE-AWARE CONSISTENCY CONSTRAINT; CIR: CONTRASTIVE ITERATIVE REFINEMENT MODULE.

The results on the Eigen split are presented in Table III and our method outperforms previous leading methods by a large margin on most metrics, such as Sq Rel and RMSE, when the maximum depth is limited to 80m. For the 0-50m depth range, our method consistently outperforms the P3Depth. In addition, we observe that when compared to the PWA, P3Depth achieves superior performance within the 0-50m range but deteriorates

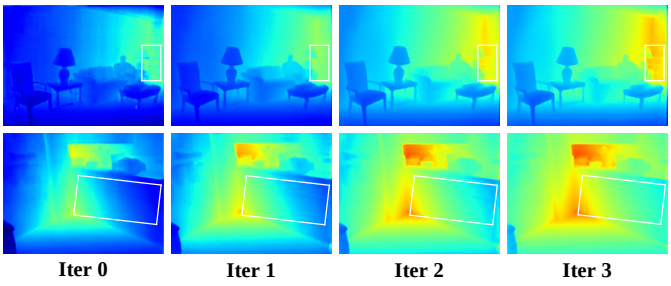

Fig. 10. **Visualization of the refinement process in the contrastive iterative refinement module**. The first depth map in the upper row at iter 0 is from the normal-distance head, while the first depth map in the lower row at iter 0 is from the depth head.

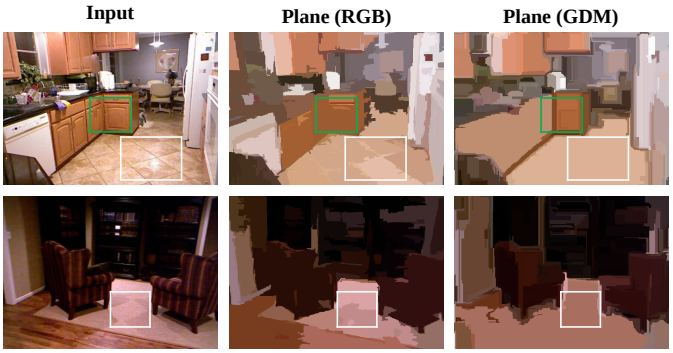

Fig. 11. **Illustration of extracted planes from RGB images and geometric dissimilarity maps**. GDM: geometric dissimilarity map.

Setting	Abs Rel ↓	RMSE ↓	$\log_{10}$ ↓	$\delta_{1.25}$ ↑
Ours ($b = 0.05$)	**0.087**	0.313	**0.038**	0.935
Ours ($b = 0.1$)	**0.087**	0.312	**0.038**	**0.936**
Ours ($b = 0.2$)	**0.087**	**0.311**	**0.038**	**0.936**
Ours ($b = 0.4$)	**0.087**	0.312	**0.038**	0.934
Ours ($b = 0.8$)	**0.087**	0.313	**0.038**	0.935
Ours (w/o U)	0.088	0.314	**0.038**	0.933

TABLE VII
EFFECT OF UNCERTAINTY MODELING. b: ERROR TOLERANCE COEFFICIENT; U: UNCERTAINTY.

significantly within the 0-80m range. This discrepancy could be attributed to the prevalence of planar regions, such as roads, in nearby areas, while more distant regions are characterized by the high-curvature features such as bushes, trees, and other clutter, leading to the failure of the planarity prior. By contrast, our method relaxes such failure cases via the incorporation of a depth head, yielding promising results in both depth ranges. Fig. 7 demonstrates qualitative depth results. As we can see, our method exhibits a higher proficiency in delineating planar regions, *e.g.*, wall and is less susceptible to color changes.

Table IV demonstrates the results on the official split. Once again, our method exceeds the compared methods, consistently showing improvements in all metrics. Additionally, the main ranking metric, SILog, experiences a considerable reduction, with the proposed method **ranking 1st** among all submissions on the KITTI depth prediction online benchmark at the time of submission. Fig. 9 displays qualitative depth results, which are

reported by the online server. The higher-quality depth maps from our method further emphasize our contributions.

Zero-shot generalization. Table V demonstrates the comparison of generalization performance in a challenging zero-shot setting. The models are trained using the NYU-Depth-v2 dataset but evaluated using the SUN RGB-D dataset. The impressive performance shows that the proposed physics-driven deep learning framework has strong generalization capabilities across different datasets. It indeed learns transferable features rather than merely memorizing training data statistics.

Ablation study. To gain a deeper understanding of how each key component contributes to the performance, we conduct an ablation study on the whole framework and provide the results in Table VI.

The results indicate that the normal-distance head surpasses the depth head when deployed independently, which appears to contradict the finding in [27] that predicting depth directly is superior to predicting plane coefficients. The reason behind may originate from our use of a more explicit normal-distance representation, as opposed to the implicit plane representation used in [27]. This explicit representation enables us to introduce direct supervisory signals to guide the learning of surface normal and distance, hence resulting in more accurate depth estimates. Building upon the normal-distance head, we enforce the plane-aware consistency constraint on surface normal and distance maps, which consistently leads to improvements on most metrics. Then, we incorporate the depth head into our framework. The resulting improvements highlight the inherent complementarity of the depth estimates generated by these two heads. Finally, we append the contrastive iterative refinement module, thereby completing the entire framework and attaining the best results.

In addition, we compare the accuracy of depth maps from normal-distance head and depth head in planar and non-planar regions, respectively, and the results support our standpoint.

To examine how the uncertainty modeling affects the performance, we adjust the error tolerance coefficient in uncertainty modeling, including directly removing uncertainty modeling. The results are demonstrated in Table VII. We find that setting the error tolerance coefficient to 0.1 or 0.2 gives better results than other values. Further, when the uncertainty modeling is removed, the worst performance is shown. This indicates that uncertainty plays an essential role in the refinement process.

Fig. 10 illustrates the refinement process in the contrastive iterative refinement module. As can be seen, when the number of iterations increases, the details of the locker in the depth maps from normal-distance head become increasingly distinct, while the kitchen cabinet surface in the depth maps from depth head displays a more continuous appearance.

In Fig. 11, we show extracted planes based on RGB images and geometric dissimilarity maps. The RGB-based plane detection suffers from the following failure cases: pixels located on the same plane may appear in different colors and hence be segmented into different planes (white boxes); pixels lying in different planes may be segmented into one plane because of the similar colors (green boxes). In contrast, our geometric dissimilarity map-based plane detection is more robust to such cases benefiting from the piece-wise constant property.

Method	RMSE	REL	$\delta_{1.25}$	$\delta_{1.25^2}$	$\delta_{1.25^3}$
S2D [41]	0.230	0.044	97.1	99.4	99.8
[41]+Bilateral [89]	0.479	0.084	92.4	97.6	98.9
[41]+SPN [43]	0.172	0.031	98.3	99.7	99.9
DepthCoeff [90]	0.118	0.013	99.4	99.9	-
CSPN [16]	0.117	0.016	99.2	99.9	**100.0**
CSPN++ [44]	0.116	-	-	-	-
DeepLiDAR [54]	0.115	0.022	99.3	99.9	**100.0**
DepthNormal [55]	0.112	0.018	99.5	99.9	**100.0**
NLSPN [17]	0.092	0.012	99.6	99.9	**100.0**
ACMNet [14]	0.105	0.015	99.4	99.9	**100.0**
TWISE [91]	0.097	0.013	99.6	99.9	**100.0**
RigNet [15]	0.090	0.013	99.6	99.9	**100.0**
DySPN [18]	0.090	0.012	99.6	99.9	**100.0**
CFormer [23]	0.090	0.012	99.6	99.9	**100.0**
Ours	**0.081**	**0.010**	**99.7**	**100.0**	**100.0**

TABLE VIII
QUANTITATIVE DEPTH COMPLETION COMPARISON ON THE NYU-DEPTH-V2 DATASET. WE NOTE THAT S2D [41] UTILIZES 200 SAMPLED DEPTH POINTS PER IMAGE AS INPUT, WHEREAS THE OTHER METHODS USE 500.

Method	RMSE↓ (mm)	MAE↓ (mm)	iRMSE↓ (1/km)	iMAE↓ (1/km)
CSPN [16]	1019.64	279.46	2.93	1.15
S2D [41]	814.73	249.95	2.80	1.21
DepthNormal [55]	777.05	235.17	2.42	1.13
DeepLiDAR [54]	758.38	226.50	2.56	1.15
FuseNet [92]	752.88	221.19	2.34	1.14
CSPN++ [44]	743.69	209.28	2.07	0.90
NLSPN [17]	741.68	199.59	1.99	0.84
ACMNet [14]	744.91	206.09	2.08	0.90
TWISE [91]	840.20	195.58	2.08	**0.82**
RigNet [15]	712.66	203.25	2.08	0.90
GuideFormer [22]	721.48	207.76	2.14	0.97
DySPN [18]	709.12	**192.71**	**1.88**	**0.82**
CFormer [23]	708.87	203.45	2.01	0.88
Ours	**698.71**	192.75	1.89	0.83

TABLE IX
QUANTITATIVE DEPTH COMPLETION COMPARISON ON THE KITTI DATASET. THE RMSE IS THE MAIN RANKING METRIC.

	Method	RMSE↓ PackNet-SAN	GuideNet	NLSPN	CFormer	**Ours**
Sample Number	0	-	-	0.562	0.490	**0.471**
	50	-	-	0.223	0.208	**0.187**
	200	0.155	0.142	0.129	0.127	**0.118**
	500	0.120	0.101	0.092	0.090	**0.081**

TABLE X
SPARSITY STUDY ON THE NYU-DEPTH-V2 DATASET. EVALUATION WITH 0, 50, 200 AND 500 SAMPLES. THE COMPARED METHODS ARE PACKNET-SAN [83], GUIDENET [12], NLSPN [17] AND CFORMER [23].

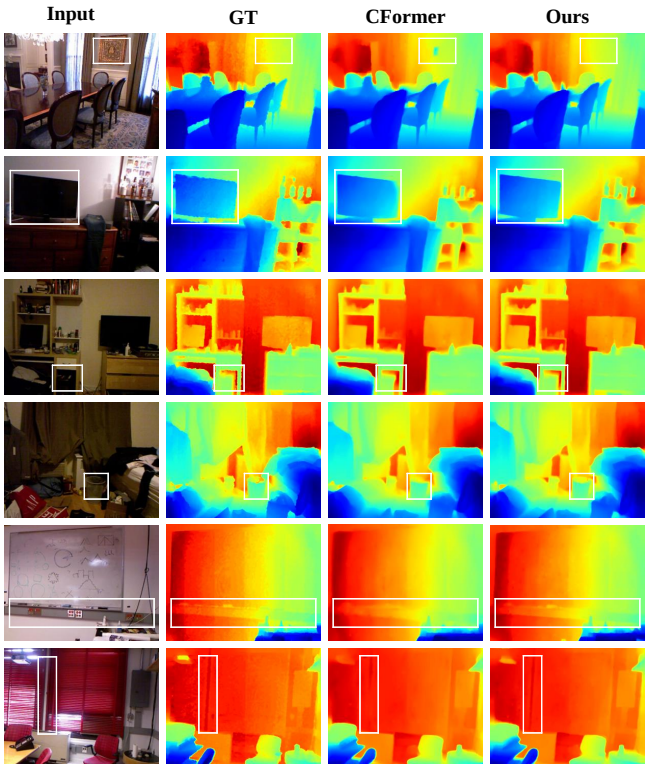

Fig. 12. **Qualitative depth results for completion on the NYU-Depth-v2 dataset**.

Fig. 13. **Qualitative point cloud results for depth completion on the NYU-Depth-v2 dataset**.

D. Depth Completion

Comparison to previous competitors. Table VIII presents the results on the NYU-Depth-v2 dataset. Despite the fact that the state-of-art performance has been nearly saturated for quite some time, such as from NLSPN to CFormer, our method is able to surpass the compared methods by a large margin. In particular, our method improves the RMSE by 10% and the REL by 16.7% over the CFormer. The large performance gap

Scanning Lines	Method	RMSE↓ (mm)	MAE↓ (mm)	iRMSE↓ (1/km)	iMAE↓ (1/km)
1	NLSPN	3507.7	1849.1	13.8	8.9
	DySPN	3625.5	1924.7	13.8	8.9
	CFormer	3250.2	1582.6	10.4	6.6
	Ours	**3219.4**	**1552.9**	**10.3**	**6.5**
4	NLSPN	2293.1	831.3	7.0	3.4
	DySPN	2285.8	834.3	6.3	3.2
	CFormer	2150.0	740.1	5.4	2.6
	Ours	**2132.3**	**710.6**	**5.2**	**2.5**
16	NLSPN	1288.9	377.2	3.4	1.4
	DySPN	1274.8	366.4	3.2	1.3
	CFormer	1218.6	337.4	3.0	**1.2**
	Ours	**1172.8**	**322.1**	**2.9**	**1.2**
64	NLSPN	889.4	238.8	2.6	1.0
	DySPN	878.5	228.6	2.5	1.0
	CFormer	848.7	215.9	2.5	**0.9**
	Ours	**819.6**	**209.1**	**2.3**	**0.9**

TABLE XI
SPARSITY STUDY ON THE KITTI DATASET. EVALUATION WITH 1, 4, 16 AND 64 SCANNING LINES. THE COMPARED METHODS ARE NLSPN [17], DYSPN [18] AND CFORMER [23].

Setting	SPN	PC	RMSE ↓	REL ↓	$\delta_{1.25}$ ↑	$\delta_{1.25^2}$ ↑
D			0.098	0.016	99.5	99.9
N$\mathcal{D}$			0.086	0.013	99.6	**100.0**
D	✓		0.090	0.012	99.6	99.9
N$\mathcal{D}$	✓		0.084	0.011	99.6	**100.0**
N$\mathcal{D}$	✓	✓	0.082	0.011	**99.7**	**100.0**
D&N$\mathcal{D}$	✓	✓	**0.081**	**0.010**	**99.7**	**100.0**

TABLE XII
ABLATION STUDY FOR DEPTH COMPLETION. SPN: SPN REFINEMENT. D&N$\mathcal{D}$: INTEGRATED DEPTH AND NORMAL-DISTANCE HEADS; D: DEPTH HEAD; N$\mathcal{D}$: NORMAL-DISTANCE HEAD; PC: PLANE-AWARE CONSISTENCY CONSTRAINT.

emphasizes that the proposed method significantly contributes to improving the accuracy. In Fig. 12, we provide qualitative depth results, and our method delivers more accurate estimates in planar regions and preserves depth edges. In Fig. 13, we demonstrate qualitative point cloud results. As can be seen, our method recovers the 3D structure reasonably. It is worth mentioning that in some cases, for instance, the highlighted pattern on the wall by white boxes in the first column, its point cloud recovered by ground-truth depth map shows slight errors in the description of internal structure, while our approach is capable of alleviating such failure cases, indicating that our physics-driven deep learning framework has robustness to the data noise.

Table IX presents the results on the KITTI dataset. It can be seen that our method outperforms the compared methods on the main ranking metric RMSE. In Fig. 14, we demonstrate qualitative results from the online server and our method can achieve higher-quality depth maps than the CFormer.

Sparsity study. To showcase the robustness of our method

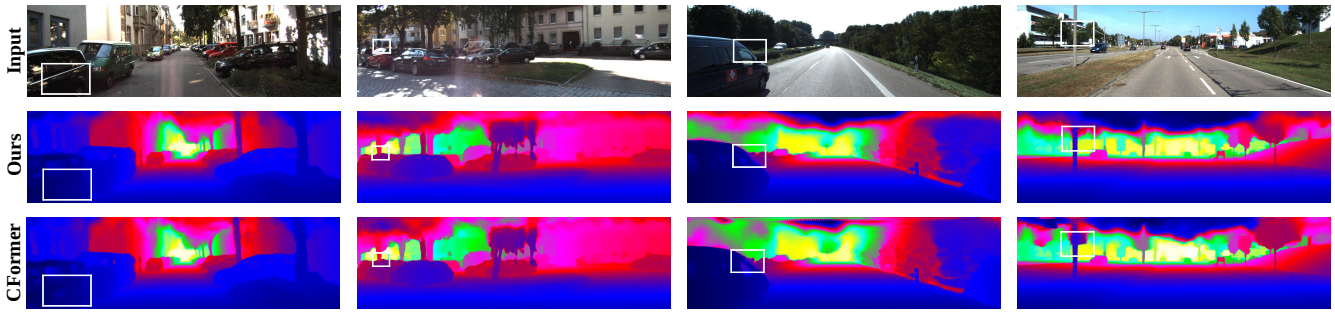

Fig. 14. **Qualitative depth completion results on the KITTI dataset**.

Method	RMSE ↓	Param. (M) ↓	Inf. (s) ↓	FLOPs (G)↓
Monocular depth estimation				
NeWCRFs [11]	0.334	270	0.089	281
Ours (w/o CIR&D)	0.318	270	0.090	281
Ours (w/o CIR)	0.315	346	0.103	338
Ours	0.311	348	0.121	481
Depth completion				
CFormer [23]	0.090	83	0.098	86
Ours (w/o D)	0.082	83	0.100	94
Ours	0.081	87	0.114	126

TABLE XIII
COMPARISON OF MODEL PARAMETERS, INFERENCE TIME AND FLOPS ON THE NYU-DEPTH-V2 DATASET. PARAM.: PARAMETERS; INF.: INFERENCE TIME; CIR: CONTRASTIVE ITERATIVE REFINEMENT; D: DEPTH HEAD.

when subjected to varying levels of data sparsity, we manually generate sparse data using different settings for training and testing. Regarding the NYU-Depth-v2 dataset, we randomly sample 0, 50, 200, and 500 points from the ground-truth depth, surface normal and distance maps to simulate various levels of sparsity. For the KITTI dataset, we follow [91] and subsample the raw LiDAR scans in the azimuth-elevation space, reducing them into 1, 4 and 16 lines. Besides, we adopt the training set with 10000 images provided by [23] and perform evaluation on the KITTI validation set. The results of other approaches are from [23]. As summarized in Tables X and XI, our method consistently surpasses the compared methods in all cases on both datasets.

Ablation study. In Table XII, we list the results of ablation study to better understand the impact of each key component. As can be seen, the SPN refinement is beneficial to improve performance whether using a normal-distance head or a depth head. In addition, the standalone performance of the normal-distance head surpasses that of the depth head by a large margin. However, this is not surprising considering that it could be easier to complete sparse surface normal and distance maps than it is to complete sparse depth maps. Specifically, the plane coefficients are piece-wise constant while the depth values tend to change from pixel to pixel. Once the sparse surface normal map or distance map has a value on a certain plane, one can easily complete this planar region by simply copying the value. Based on the normal-distance head, we apply the plane-aware consistency constraint for surface normal and distance maps, achieving improvements on most metrics. Finally, we combine the normal-distance head and depth head, and obtain the best results.

E. Comparison of Model Parameters, FLOPs and Inference Time

We conduct a comparison between the proposed method and previous methods focusing on their inference time, number of model parameters and FLOPs. We measure the inference time and FLOPs using a batch size of 1 on the NYU-Depth-v2 test set. For monocular depth estimation, we compare it against NeWCRFs, both utilizing the Swin-Large backbone. In terms of depth completion, we contrast it with CFormer, both using the backbone from [23]. The results are reported in Table XIII. It can be seen that our method achieves superior performance with comparable number of parameters, FLOPs and inference time to its counterparts for these two tasks. Without integrating the depth head and contrastive iterative refinement, our method is able to improve the NeWCRFs at a very small additional cost. Besides, we find that the contrastive iterative refinement is capable of improving the performance with a small number of parameters. Nonetheless, it will bring more inference time and FLOPs. More notably, without integrating the depth head, our method improves the CFormer by 8.9% in RMSE, with the same number of parameters, and a slight increase in inference time and FLOPs.

V. CONCLUSION

In this paper, we introduce new physics-driven deep learning frameworks for monocular depth estimation and completion based on the planar constraints in 3D scenes. The frameworks involve a normal-distance head to predict the piece-wise planar depth and a regular depth head to strengthen the robustness. In order to regularize the surface normal and distance maps, we develop a plane-aware consistency constraint. In addition, we develop a contrastive iterative refinement module to refine the depth maps. Extensive experiments indicate that the proposed method outperforms previous state-of-the-art monocular depth estimation and completion competitors on the NYU-Depth-v2, KITTI and SUN RGB-D datasets.

REFERENCES

[1] Keisuke Tateno, Federico Tombari, Iro Laina, and Nassir Navab. Cnn-slam: Real-time dense monocular slam with learned depth prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6243–6252, 2017.

[2] Weibin Jia, Wenjie Zhao, Zhihuan Song, and Zhengguo Li. Object servoing of differential-drive service robots using switched control. *Journal of Control and Decision*, 10(3):314–325, 2023.

[3] Caner Hazirbas, Lingni Ma, Csaba Domokos, and Daniel Cremers. Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture. In *Asian Conference on Computer Vision*, pages 213–228. Springer, 2016.

[4] Wonwoo Lee, Nohyoung Park, and Woontack Woo. Depth-assisted real-time 3d object detection for augmented reality. In *ICAT*, volume 11 (2), pages 126–132, 2011.

[5] Jeff Michels, Ashutosh Saxena, and Andrew Y Ng. High speed obstacle avoidance using monocular vision and reinforcement learning. In *Proceedings of the 22nd international conference on Machine learning*, pages 593–600, 2005.

[6] Takaaki Nagai, Takumi Naruse, Masaaki Ikehara, and Akira Kurematsu. Hmm-based surface reconstruction from single images. In *Proceedings of the International Conference on Image Processing*, volume 2, pages II–II. IEEE, 2002.

[7] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems*, pages 2366–2374, 2014.

[8] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2002–2011, 2018.

[9] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326*, 2019.

[10] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4009–4018, 2021.

[11] Weihao Yuan, Xiaodong Gu, Zuozhuo Dai, Siyu Zhu, and Ping Tan. Neural window fully-connected crfs for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3916–3925, June 2022.

[12] Jie Tang, Fei-Peng Tian, Wei Feng, Jian Li, and Ping Tan. Learning guided convolutional network for depth completion. *IEEE Transactions on Image Processing*, 30:1116–1129, 2020.

[13] Alex Wong and Stefano Soatto. Unsupervised depth completion with calibrated backprojection layers. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 12747–12756, 2021.

[14] Shanshan Zhao, Mingming Gong, Huan Fu, and Dacheng Tao. Adaptive context-aware multi-modal network for depth completion. *IEEE Transactions on Image Processing*, 30:5264–5276, 2021.

[15] Zhiqiang Yan, Kun Wang, Xiang Li, Zhenyu Zhang, Jun Li, and Jian Yang. Rignet: Repetitive image guided network for depth completion. In *Proceedings of the European Conference on Computer Vision*, pages 214–230. Springer, 2022.

[16] Xinjing Cheng, Peng Wang, and Ruigang Yang. Depth estimation via affinity learned with convolutional spatial propagation network. In *Proceedings of the European Conference on Computer Vision*, pages 103–119, 2018.

[17] Jinsun Park, Kyungdon Joo, Zhe Hu, Chi-Kuei Liu, and In So Kweon. Non-local spatial propagation network for depth completion. In *Proceedings of the European Conference on Computer Vision*, pages 120–136. Springer, 2020.

[18] Yuankai Lin, Tao Cheng, Qi Zhong, Wending Zhou, and Hua Yang. Dynamic spatial propagation network for depth completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1638–1646, 2022.

[19] Jiaqi Gu, Zhiyu Xiang, Yuwen Ye, and Lingxuan Wang. Denselidar: A real-time pseudo dense depth guided depth completion network. *IEEE Robotics and Automation Letters*, 6(2):1808–1815, 2021.

[20] Lina Liu, Xibin Song, Xiaoyang Lyu, Junwei Diao, Mengmeng Wang, Yong Liu, and Liangjun Zhang. Fcfr-net: Feature fusion based coarse-to-fine residual learning for depth completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2136–2144, 2021.

[21] Lina Liu, Yiyi Liao, Yue Wang, Andreas Geiger, and Yong Liu. Learning steering kernels for guided depth completion. *IEEE Transactions on Image Processing*, 30:2850–2861, 2021.

[22] Kyeongha Rho, Jinsung Ha, and Youngjung Kim. Guideformer: Transformers for image guided depth completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6250–6259, 2022.

[23] Youmin Zhang, Xianda Guo, Matteo Poggi, Zheng Zhu, Guan Huang, and Stefano Mattoccia. Completionformer: Depth completion with convolutions and vision transformers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 18527–18536, 2023.

[24] András Bódis-Szomorú, Hayko Riemenschneider, and Luc Van Gool. Fast, approximate piecewise-planar modeling based on sparse structure-from-motion and superpixels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 469–476, 2014.

[25] Anne-Laure Chauve, Patrick Labatut, and Jean-Philippe Pons. Robust piecewise-planar 3d reconstruction and completion from large-scale unstructured point data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1261–1268. IEEE, 2010.

[26] Chen Liu, Kihwan Kim, Jinwei Gu, Yasutaka Furukawa, and Jan Kautz. Planercnn: 3d plane detection and reconstruction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4450–4459, 2019.

[27] Vaishakh Patil, Christos Sakaridis, Alexander Liniger, and Luc Van Gool. P3depth: Monocular depth estimation with a piecewise planarity prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1610–1621, 2022.

[28] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59:167–181, 2004.

[29] Shuwei Shao, Zhongcai Pei, Weihai Chen, Xingming Wu, and Zhengguo Li. Nddepth: Normal-distance assisted monocular depth estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7931–7940, October 2023.

[30] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Proceedings of the European Conference on Computer Vision*, pages 746–760. Springer, 2012.

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

[32] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 567–576, 2015.

[33] Ashutosh Saxena, Sung H Chung, Andrew Y Ng, et al. Learning depth from single monocular images. In *Advances in Neural Information Processing Systems*, volume 18, pages 1–8, 2005.

[34] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *2016 Fourth International Conference on 3D Vision*, pages 239–248. IEEE, 2016.

[35] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2016.

[36] Yuanzhouhan Cao, Zifeng Wu, and Chunhua Shen. Estimating depth from monocular images as classification using deep fully convolutional residual networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(11):3174–3182, 2017.

[37] Zhenyu Li, Xuyang Wang, Xianming Liu, and Junjun Jiang. Binsformer: Revisiting adaptive bins for monocular depth estimation. *arXiv preprint arXiv:2204.00987*, 2022.

[38] Ashutosh Agarwal and Chetan Arora. Attention attention everywhere: Monocular depth prediction with skip attention. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, pages 5861–5870, 2023.

[39] Guanglei Yang, Hao Tang, Mingli Ding, Nicu Sebe, and Elisa Ricci. Transformer-based attention networks for continuous pixel-wise prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 16269–16279, October 2021.

[40] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations*, 2021.

[41] Fangchang Ma and Sertac Karaman. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In *IEEE International Conference on Robotics and Automation*, pages 4796–4803. IEEE, 2018.

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

[43] Sifei Liu, Shalini De Mello, Jinwei Gu, Guangyu Zhong, Ming-Hsuan Yang, and Jan Kautz. Learning affinity via spatial propagation networks. *Advances in Neural Information Processing Systems*, 30, 2017.

[44] Xinjing Cheng, Peng Wang, Chenye Guan, and Ruigang Yang. Cspn++: Learning context and resource aware convolutional spatial propagation networks for depth completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10615–10622, 2020.

[45] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 764–773, 2017.

[46] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4320–4328, 2018.

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

[48] Mu Hu, Shuling Wang, Bin Li, Shiyu Ning, Li Fan, and Xiaojin Gong. Penet: Towards precise and efficient image guided depth completion. In *2021 IEEE International Conference on Robotics and Automation*, pages 13656–13662. IEEE, 2021.

[49] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6):1397–1409, 2012.

[50] Yiqi Zhong, Cho-Ying Wu, Suya You, and Ulrich Neumann. Deep rgb-d canonical correlation analysis for sparse depth completion. *Advances in Neural Information Processing Systems*, 32, 2019.

[51] Yongchi Zhang, Ping Wei, Huan Li, and Nanning Zheng. Multiscale adaptation fusion networks for depth completion. In *International Joint Conference on Neural Networks*, pages 1–7. IEEE, 2020.

[52] David Gallup, Jan-Michael Frahm, and Marc Pollefeys. Piecewise planar and non-planar stereo for urban scene reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1418–1425. IEEE, 2010.

[53] Xiaojuan Qi, Renjie Liao, Zhengzhe Liu, Raquel Urtasun, and Jiaya Jia. Geonet: Geometric neural network for joint depth and surface normal estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 283–291, 2018.

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

[55] Yan Xu, Xinge Zhu, Jianping Shi, Guofeng Zhang, Hujun Bao, and Hongsheng Li. Depth completion from sparse lidar data with depth-normal constraints. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2811–2820, 2019.

[56] Uday Kusupati, Shuo Cheng, Rui Chen, and Hao Su. Normal assisted stereo depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2189–2199, 2020.

[57] Xiaoxiao Long, Cheng Lin, Lingjie Liu, Wei Li, Christian Theobalt, Ruigang Yang, and Wenping Wang. Adaptive surface normal constraint for depth estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 12849–12858, October 2021.

[58] Lam Huynh, Phong Nguyen-Ha, Jiri Matas, Esa Rahtu, and Janne Heikkilä. Guiding monocular depth estimation using depth-attention volume. In *Proceedings of the European Conference on Computer Vision*, pages 581–597. Springer, 2020.

[59] Wei Yin, Yifan Liu, Chunhua Shen, and Youliang Yan. Enforcing geometric constraints of virtual normal for depth prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5684–5693, 2019.

[60] Alejo Concha and Javier Civera. Using superpixels in monocular slam. In *IEEE International Conference on Robotics and Automation*, pages 365–372. IEEE, 2014.

[61] Alejo Concha and Javier Civera. Dpptam: Dense piecewise planar tracking and mapping from a monocular sequence. In *IEEE International Conference on Intelligent Robots and Systems*, pages 5686–5693. IEEE, 2015.

[62] Zehao Yu, Lei Jin, and Shenghua Gao. P 2 net: patch-match and plane-regularization for unsupervised indoor depth estimation. In *Proceedings of the European Conference on Computer Vision*, pages 206–222. Springer, 2020.

[63] Boying Li, Yuan Huang, Zeyu Liu, Danping Zou, and Wenxian Yu. Structdepth: Leveraging the structural regularities for self-supervised indoor depth estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 12663–12673, 2021.

[64] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in Neural Information Processing Systems*, 30, 2017.

[65] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Proceedings of the European Conference on Computer Vision*, pages 402–419. Springer, 2020.

[66] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning*, 2014.

[67] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2650–2658, 2015.

[68] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10012–10022, October 2021.

[69] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12009–12019, 2022.

[70] Jinqing Zhang, Haosong Yue, Xingming Wu, Weihai Chen, and Changyun Wen. Densely connecting depth maps for monocular depth estimation. In *Proceedings of the European Conference on Computer Vision Workshop*, pages 149–165. Springer, 2020.

[71] Sihaeng Lee, Janghyeon Lee, Byungju Kim, Eojindl Yi, and Junmo Kim. Patch-wise attention network for monocular depth estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1873–1881, 2021.

[72] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 12179–12188, 2021.

[73] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Localbins: Improving depth estimation by learning local distributions. In *Proceedings of the European Conference on Computer Vision*, pages 480–496. Springer, 2022.

[74] Zhenyu Li, Zehui Chen, Xianming Liu, and Junjun Jiang. Depthformer: Exploiting long-range correlation and local information for accurate monocular depth estimation. *arXiv preprint arXiv:2203.14211*, 2022.

[75] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *Advances in Neural Information Processing Systems Workshop Autodiff*, 2017.

[76] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[77] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

[78] David F Fouhey, Abhinav Gupta, and Martial Hebert. Data-driven 3d primitives for single image understanding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3392–3399, 2013.

[79] Lubor Ladicky, Jianbo Shi, and Marc Pollefeys. Pulling things out of perspective. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 89–96, 2014.

[80] Xiaolong Wang, David Fouhey, and Abhinav Gupta. Designing deep networks for surface normal estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 539–547, 2015.

[81] Xiaojuan Qi, Zhengzhe Liu, Renjie Liao, Philip HS Torr, Raquel Urtasun, and Jiaya Jia. Geonet++: Iterative geometric neural network with edge-aware refinement for joint depth and surface normal estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(2):969–984, 2020.

[82] Yukang Gan, Xiangyu Xu, Wenxiu Sun, and Liang Lin. Monocular depth estimation with affinity, vertical pooling, and label enhancement. In *Proceedings of the European Conference on Computer Vision*, pages 224–239, 2018.

[83] Vitor Guizilini, Rares Ambrus, Wolfram Burgard, and Adrien Gaidon. Sparse auxiliary networks for unified monocular depth prediction and completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11078–11088, 2021.

[84] Zhenheng Yang, Peng Wang, Wei Xu, Liang Zhao, and Ramakant Nevatia. Unsupervised learning of geometry from videos with edge-aware depth-normal consistency. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.
[85] Zhenheng Yang, Peng Wang, Yang Wang, Wei Xu, and Ram Nevatia. Lego: Learning edge with geometry all at once by watching videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 225–234, 2018.
[86] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Yan Yan, Nicu Sebe, and Jian Yang. Pattern-affinitive propagation across depth, surface normal and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4106–4115, 2019.
[87] Shubhra Aich, Jean Marie Uwabeza Vianney, Md Amirul Islam, and Mannat Kaur Bingbing Liu. Bidirectional attention network for monocular depth estimation. In *2021 IEEE International Conference on Robotics and Automation*, pages 11746–11752. IEEE, 2021.
[88] Xiaotian Chen, Xuejin Chen, and Zheng-Jun Zha. Structure-aware residual pyramid network for monocular depth estimation. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 694–700, 2019.
[89] Jonathan T Barron and Ben Poole. The fast bilateral solver. In *Proceedings of the European Conference on Computer Vision*, pages 617–632. Springer, 2016.
[90] Saif Imran, Yunfei Long, Xiaoming Liu, and Daniel Morris. Depth coefficients for depth completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12438–12447. IEEE, 2019.
[91] Saif Imran, Xiaoming Liu, and Daniel Morris. Depth completion with twin surface extrapolation at occlusion boundaries. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2583–2592, 2021.
[92] Yun Chen, Bin Yang, Ming Liang, and Raquel Urtasun. Learning joint 2d-3d representations for depth completion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10023–10032, 2019.

Shuwei Shao received the B.Eng. degree from Xidian University, Xi'an, Shanxi, China, in 2019. He is currently pursuing the Ph.D. degree in the School of Automation Science and Electrical Engineering, Beihang University, Beijing, China. His current research interests include depth perception, image registration and 3D reconstruction.

Zhongcai Pei received the B.Eng., M.Eng. and Ph.D. degrees from the Harbin Institute of Technology, Harbin, China, in 1991, 1994, and 1997, respectively. He has been with the School of Automation Science and Electronic Engineering, Beihang University, as an Associate Professor from 2000 and as a Professor since 2006. He has published over 50 technical papers in referred journals and conference proceedings and filed more than 20 patents. His research interests include bionic CPG mechanisms and control, computer vision, parallel mechanism and robotics.

Weihai Chen received the B.Eng. degree from Zhejiang University, China, in 1982, and the M.Eng. and Ph.D. degrees from Beihang University, Beijing, China, in 1988 and 1996, respectively. He is currently a Professor with the School of Automation Science and Electrical Engineering, Beihang University. He has published more than 200 technical articles in referred journals and conference proceedings and filed more than 30 patents. His research interests include computer vision, image processing, automation, bioinspired robotics, and precision mechanisms. He is also a member of the Technical Committee on Manufacturing Automation of the IEEE Robotics and Automation Society. He was a recipient of the 2017 Best Paper Award for the IEEE Transactions on Industrial Electronics and the 2018 IET Premium Award for the *IET Intelligent Transport Systems*.

Peter C. Y. Chen received the Ph.D. degree from the University of Toronto, Toronto, ON, Canada, in 1995. During 1985–1997, he worked on various projects for China International Marine Containers Ltd., China, Automation Tooling Systems, Inc., Canada, General Dynamics Corp., Electronics Division, USA, Philips Pte. Ltd., Singapore, the University of Toronto, CRS Robotics Inc., Canada, and Canadian Space Agency, Canada. From September 1997 to March 2000, he was a Research Fellow with the Singapore Institute of Manufacturing Technology, Singapore, where he developed modular reconfigurable robotic systems for manufacturing applications. He is currently with the Department of Mechanical Engineering, National University of Singapore, Singapore. His research interests include robotics and formal methods for reinforcement learning.

Zhengguo Li (F'23) received the B.Sci (Applied Mathematics). and M.Eng. (Automatic Control) from Northeastern University, Shenyang, China, in 1992 and 1995, respectively, and the Ph.D. degree (Automatic Control) from Nanyang Technological University, Singapore, in 2001.

His research interests include video processing and delivery, computational photography, switched and impulsive control, sensor fusion and physics-driven deep learning. He has co-authored one monograph, more than 200 journal/conference papers including more than 60 IEEE Transaction papers, and eleven granted USA patents, including normative technologies on scalable extension of H.264/AVC and HEVC. He has been actively involved in the development of H.264/AVC and HEVC since 2002. He had three informative proposals adopted by the H.264/AVC and three normative proposals adopted by the HEVC. Currently, he is with the Agency for Science, Technology and Research, Singapore.

Zhengguo served as a TPC Chair of IEEE IECON 2023 and 2020, a special session chair of IEEE ICASSP 2022, a Technical Brief Co-Founder of SIGGRAPH Asia, and leading Workshop Chair of IEEE ICME in 2013. He was an Associate Editor of IEEE Signal Processing Letters from 2014-2016, IEEE Transactions on Image Processing from 2016-2020, IEEE Transactions on Circuits and Systems for Video Technology from 2020-2023, APSIPA Transactions on Signal and Information Processing from 2022-2025, an editor of MDPI Sensors from 2022-2024, and a Senior Area Editor of IEEE Transactions on Image Processing from 2020-2024.