# Deep Depth Completion from Extremely Sparse Data: A Survey

Junjie Hu, *Member, IEEE,* Chenyu Bao, Mete Ozay, Chenyou Fan, Qing Gao, Honghai Liu, *Fellow, IEEE* and Tin Lun Lam, *Senior Member, IEEE*

**Abstract**—Depth completion aims at predicting dense pixel-wise depth from an extremely sparse map captured from a depth sensor, e.g., LiDARs. It plays an essential role in various applications such as autonomous driving, 3D reconstruction, augmented reality, and robot navigation. Recent successes on the task have been demonstrated and dominated by deep learning based solutions. In this article, for the first time, we provide a comprehensive literature review that helps readers better grasp the research trends and clearly understand the current advances. We investigate the related studies from the design aspects of network architectures, loss functions, benchmark datasets, and learning strategies with a proposal of a novel taxonomy that categorizes existing methods. Besides, we present a quantitative comparison of model performance on three widely used benchmarks, including indoor and outdoor datasets. Finally, we discuss the challenges of prior works and provide readers with some insights for future research directions.

**Index Terms**—Depth Completion, deep learning, depth estimation, multi-modality fusion, spatial propagation network.

## 1 Introduction

ACQUIRING correct pixel-wise scene depth plays a substantial role in various tasks such as scene understanding [54], autonomous driving [99], robotic navigation [75], [104], simultaneous localization and mapping [35], intelligent farming [23], and augmented reality [19]. Thus, it has been a long-term goal studied in past decades. One cost-effective way of obtaining scene depth is to directly estimate it from a single image with monocular depth estimation algorithms [27], [32], [42], [61]. However, visual methods often yield a low inference accuracy and poor generalizability and thus are vulnerable to real-world deployment.

On the other hand, depth sensors provide accurate and robust distance measurements with true scene scales. Therefore, they are more applicable for applications that require a security guarantee and high performance [26], [76], [99], e.g., self-driving cars. In fact, measuring depths with LiDARs is probably still the most deployable way to obtain reliable depth in industrial applications. However, neither LiDAR nor commonly used RGBD cameras, like Microsoft Kinect, can provide a dense pixel-wise depth map. As shown in Fig. 1, the depth map captured by Kinect has small holes and the map captured by LiDAR is significantly more sparse. It is, therefore, necessary to fill the void pixels in practice.

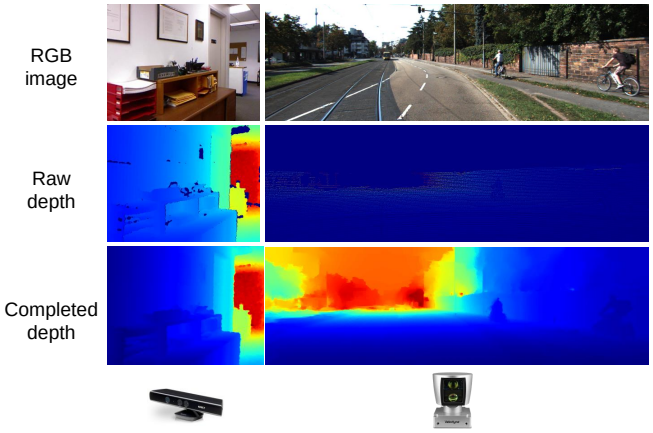



Fig. 1. Comparison between captured depth maps by different sensors. The raw sparse depth maps are shown in the middle. The left one is captured by a Kinect in an indoor scenario, and the right one is captured by a LiDAR in an outdoor street. Clearly, the map captured by LiDAR is significantly more sparse. The bottom row shows the completed depth map from the raw sparse map.

Since there is a clear difference among depth maps captured by different sensors, the completion problem and solution are usually sensor-dependent. For example, it is frequently called depth enhancement [48], [74], [96], depth inpainting [66], [80] and depth denoising [28], [96] in many works, where the goal is to infer missing depth values from dense raw depth maps and eliminate outliers (typically, the density is over 80% as discussed in [76]). In this article, we particularly focus on the completion task for extremely sparse data, e.g., for depth maps captured by LiDARs where the sparsity is usually over 95%. This problem is studied and handled separately in related literature and is much more challenging due to the low density of the sparse input. For simplicity, we refer to depth completion from extremely data as depth completion in the rest of the article.

In recent years, deep learning based methods have


- *J.Hu, Q.Gao and T.L.Lam are with the Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS), China. E-mail: hujunjie@cuhk.edu.cn,gaoqing@cuhk.edu.cn.*
- *C.Bao and T.L.Lam are with the School of Science and Engineering, the Chinese University of Hong Kong, Shenzhen, China. E-mail: boggy615@gmail.com, tllam@cuhk.edu.cn.*
- *M.Ozay is with the Samsung Research, UK. E-mail: meteozay@gmail.com.*
- *C.Fan is with the School of Artificial Intelligence, South China Normal University, China. E-mail: fanchenyou@scnu.edu.cn.*
- *Q.Gao is also with the School of Electronics and Communication Engineering, Sun Yat-sen University, Shenzhen, China.*
- *H.Liu is with the State Key Laboratory of Robotics and systems, Harbin Institute of Technology (Shenzhen), China. E-mail: honghai.liu@hit.edu.cn.*
- *J.Hu and C.Bao contribute equally.*
- *T.L.Lam is the corresponding author.*

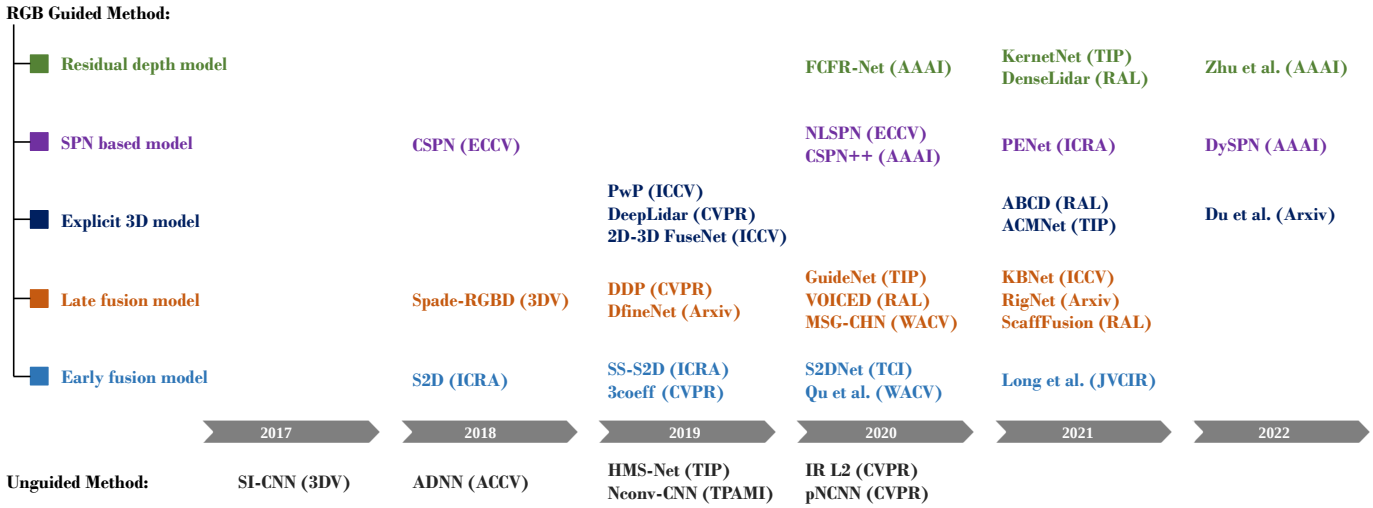



Fig. 2. A timeline for deep learning based depth completion methods. We show some selected works to visualize the evolution process. Unguided methods: SI-CNN [107], ADNN [14], HMS-Net [48], Ncon-CNN [110], IR L2 [73], pNCNN [21]. RGB guided methods: 1) Early fusion models: S2D [77], SS-S2D [76], 3coeff [51], S2DNet [36], Qu et al. [88], Long et al. [70]. 2) Late fusion models: Spade-RGBD [54], DDP [127], DfineNet [132], GuideNet [102], VOICED [118], MSG-CHN [63], KBNet [119], RigNet [125], ScaffFusion [116]. 3) Explicit 3D representation models: PwP [121], DeepLidar [87], 2D-3D fuseNet [9], ABCD [55], ACMNet [137], Du et al. [20]. 4) SPN-based models: CSPN [13], NLSPN [85], CSPN++ [12], PENet [44], DySPN [65]. 5) Residual depth models: FCFR-Net [68], KernelNet [67], DenseLiDAR [34], Zhu et al. [140].

shown compelling performance on the task and have led the development trend. It is shown in prior works that a network with several convolutional layers [107], or a simple auto-encoder [111] can complete missing depths. Moreover, depth completion can be further improved by leveraging RGB information. A typical method of this type [54], [97] is to use dual encoders for extracting features from a sparse depth map and its corresponding RGB image, respectively, and later fuse them with a decoder.

To push the envelope of depth completion, recent approaches tend to use complicated network structures and complex learning strategies. In addition to multi-branches used for feature extraction from multi-modality data, e.g., image and sparse depths, researchers have begun to integrate surface normal [87], affinity matrix [13], residual depth map [34], etc., into their frameworks. Besides, to cope with the lack of supervised pixels, some works introduced exploiting multi-view geometric constraints [76] and adversarial regularization [58]. These efforts have greatly facilitated the progress in the depth completion task.

Despite the tremendous progress made by learning based approaches, to the best of our knowledge, a comprehensive survey is lacking. This article aims to depict the development of learning based depth completion through hierarchically analyzing and categorizing existing methods and provide readers with a straightforward understanding of deep depth completion with some valuable instructions. Typically, we hope to answer the following questions:

1) What are the common characteristics of previous methods for achieving highly accurate depth completion?
2) What are the pros and cons of RGB guided approaches compared to unguided methods?
3) Since most previous works employed both visual and LiDAR data, what are the most effective strategies for multi-modal data fusion?
4) What are the current challenges?

With the above questions being considered, we survey the related works from January 2017 to May 2022 (at the time of writing). Fig. 2 visualizes the timeline of the selected methods based on the proposed taxonomy, where the bottom and the top show the unguided and five types of RGB guided methods, respectively. It is seen that although early studies tackle depth completion in an unguided fashion, we observed that studies published after 2020 have been gradually dominated by RGB guided methods. In this article, we investigate the previous studies from the aspects of network structure, loss function, learning strategy, and benchmark datasets. We especially stress methods with the proposal of novel algorithms or significant performance boosts and properly provide visual descriptions of their technical contributions to promote the clarification. Furthermore, we provide quantitative comparisons of existing methods with essential characteristics on the most popular benchmark datasets. Through the in-depth analysis of previous studies, we wish the reader can gain a clear understanding of deep depth completion.

In summary, our key contributions are as follows:

- To the best of our knowledge, this is the first survey for depth completion. We give an in-depth and comprehensive review, including both unguided and RGB guided methods.
- We propose a novel taxonomy to categorize previous methods and visualize their main characteristics, including network structures, loss functions, and learning strategies.
- The article covers the most advanced and recent progress of deep learning based depth completion with performance comparison on benchmark datasets. It provides readers with state-of-the-art methods.
- We provide several open issues and promising future research directions.

The remainder of this article is organized as follows: Section 2 gives the formulation of deep learning based depth completion and provides the proposed taxonomy. Section 3 reviews unguided methods, and Section 4 elaborates RGB

TABLE 1
A brief overview of the proposed taxonomy.

| Main categories | Sub-categories | Major characteristics |
|---|---|---|
| Unguided methods (Sec. 3) | Sparsity-aware CNNs (**SACNN**, Sec. 3.1) | Using the binary validity mask to indicate missing elements during convolution. |
| | Normalized CNNs (**NCNN**, Sec. 3.2) | 1). Built on normalized convolution 2). Replacing the validity mask with continuous confidence mask. |
| | Training with Auxiliary Images (**TwAI**, Sec. 3.3) | Integrating image reconstruction into latent or output space to encourage learning semantic cues. Image guided training and unguided inference are employed. |
| RGB guided methods (Sec. 4) | Early fusion models (**EFM**, Sec. 4.1)<br>• Encoder-decoder networks (**EDN**, Sec. 4.1.1)<br>• Coarse to refinement prediction (**C2RP**, Sec. 4.1.2) | Directly aggregating the image and sparse depth map input or fusing the multi-modality features at the first convolutional layer. |
| | Late fusion models (**LFM**, Sec. 4.2)<br>• Dual-encoder networks (**DEN**, Sec. 4.2.1)<br>• Double encoder-decoder networks (**DEDN**, Sec. 4.2.2)<br>• Global and Local Depth Prediction (**GLDP**, Sec. 4.2.3) | The framework usually consists of dual encoders or two sub-networks; the one is used for extracting RGB features and the other is used for extracting depth features. Fusion is conducted at the intermediate layers, e.g., fusing extracted features from encoders. |
| | Explicit 3D representation models (**E3DR**, Sec. 4.3)<br>• 3D-aware convolution (**3DAC**, Sec. 4.3.1)<br>• Intermediate surface normal representation (**ISNR**, Sec. 4.3.2)<br>• Learning from point clouds (**LfPC**, Sec. 4.3.3) | Explicitly learning 3D representations, such as applying 3D convolutions, embedding surface normals, and learning from 3D point clouds. |
| | Residual depth models (**RDM**, Sec. 4.4) | Learning a coarse depth map and a residual depth map. Their combination generates the final depth map. |
| | SPN-based models (**SPM**, Sec. 4.5) | 1). Based on the spatial propagation network. 2). First learning the affinity matrix, and then applying affinity based depth refinement. |

guided methods. Section 5 introduces the loss functions employed in previous approaches. Section 6 lists the benchmark datasets and introduces the evaluation metrics for the depth completion task. Section 7 compares the previous methods from comprehensively different perspectives. Section 8 summarizes the open challenges and provides valuable directions for future research. Section 9 gives the conclusion.

## 2 Deep Learning Based Depth Completion

In this section, we first give a common formulation of the depth completion task. Then, we outline the proposed taxonomy. Noting that some methods share common characteristics, we group them by jointly considering network structures and main technical contributions.

### 2.1 Problem Formulation

In depth completion, a deep neural network $N$ with parameters $\mathcal{W}$ predicts a dense depth map $\hat{Y} \in \hat{\mathcal{Y}}$ of a given sparse depth map $Y' \in \mathcal{Y}'$ by

$$\hat{Y} = N\left(Y'; \mathcal{W}\right). \tag{1}$$

**Unguided depth completion:** In (1), depth completion is performed using only the sparse input without guidance from different modality data. Therefore, it is called unguided depth completion. These methods are reviewed in detail in Section 3.

**RGB guided depth completion:** In many works, both the sparse depth map and its corresponding RGB image are utilized for inputs. In this case, the task is formulated by

$$\hat{Y} = N\left(Y', I; \mathcal{W}\right) \tag{2}$$

where $I$ denotes the RGB image whose pixels are aligned with $Y'$. Then, task employed by (2) is referred to as RGB guided depth completion which is elucidated in Section 4.

The parameters $\mathcal{W}$ of the network $N$ are optimized to train the network by solving

$$\hat{\mathcal{W}} = \underset{\mathcal{W}}{\operatorname{argmin}}\, \mathcal{L}\left(\hat{\mathcal{Y}}, \mathcal{Y}; \mathcal{W}\right) \tag{3}$$

where $\mathcal{Y}$ denotes the set of ground truth depth maps, and $\mathcal{L}$ is a loss function which is usually defined to penalize pixel-wise discrepancy between the prediction and the ground truth on the valid pixels through back-propagation while training $N$. Depending on the specific learning strategies, some other losses, such as unsupervised photometric loss, adversarial loss, and regularization terms on depth maps, are properly applied. An in-depth discussion of learning objectives and loss functions is given in Section 5.

### 2.2 Taxonomy

In this article, we propose a detailed taxonomy by jointly considering network structures and main technical contributions. An existing method is firstly categorized into either an unguided method or an RGB guided approach. Then, it is further classified into a more specific sub-category. Table 1 gives an overview of the proposed taxonomy with descriptions of the major factors for identifying categories.

As seen, unguided methods have three sub-categories, including methods 1) employing sparsity-aware CNNs, 2) employing normalized CNNs, and 3) training with Auxiliary Images. Guided methods include five sub-categories. Some of them also have more concrete classes. For the first and second categories, i.e., early fusion and late fusion models, the fusion strategy is the main factor considered in

our taxonomy. For the late three categories, i.e, explicit 3D representation models, residual depth models, and spatial propagation network (SPN) based models, the fusion strategy is not the major factor in identifying their types since they hold distinct characteristics and both early fusion and late fusion are used in previous methods.

## 3 Unguided Depth Completion

Given a sparse depth map, unguided methods aim at directly completing it with a deep neural network model. Previous methods can be generally categorized into three groups: methods using 1) sparsity-aware CNN, 2) normalized CNN, and 3) training with auxiliary images.

### 3.1 Sparsity-Aware CNNs

The key idea of sparsity-aware methods is that they identify valid and missing elements with a binary mask during convolution operation and thus enable standard CNNs to perform better for sparse depth inputs.

Uhrig et al. [107] proposed the first deep learning based unguided method. They first verified that normal convolutions are not able to handle sparse input as they typically cause mosaic effects and proposed a new sparse convolution operation. Then, they introduced a 6-layers CNN assembled with the proposed sparse convolution. The sparse convolution uses a binary validity mask to distinguish between valid and missing values and performs convolution among only valid data. The value of the validity mask is determined by its local neighbors via max-pooling. This first deep learning based method outperforms non-learning methods and shows the potential of deep learning on the task. Moreover, it inspired lots of subsequent studies.

However, the sparse convolution is not suitable to be directly applied to classical encoder-decoder networks, which can fully leverage the multi-scale features. Huang et al. [48] introduced three sparsity invariant (SI) operations, including SI upsampling, SI average, and SI concatenation, and built an encoder-decoder based HSMNet. They also demonstrated an application using RGB inputs by adding a small branch to HSMNet.

Chodosh et al. [14] formulated the depth completion as a multi-layer convolutional compressed sensing problem and proposed an end-to-end multi-layer dictionary learning algorithm. It is achieved by applying compressed sensing to the deep component analysis (DeepCA) objective [81] and optimizing by ADMM (alternation direction method of multipliers). The over-complete dictionaries are learned with a few convolutional layers via back-propagation.

### 3.2 Normalized CNNs

The sparsity-aware methods require validity masks to identify missing values for performing convolutions. As argued in [22], [54], [110], validity masks can degrade the model performance due to the saturation of the mask at early layers in CNNs. To tackle this issue, inspired by normalized convolution [59], Eldesokey et al. [22] introduced the normalized convolutional neural network (NCNN) that generates continuous uncertainty maps for depth completion. The essential difference is that features obtained using the NCNN are weighed with continuous uncertainty maps instead of binary validity masks, leading to better completion performance. In addition, convolution filters are constrained to be non-negative by the SoftPlus function [31] for faster convergence.

Although NCNN still takes a sparse mask as an initial input, it yields a continuous confidence map to indicate useful information across the intermediate layers. In reality, disturbed measurements exist due to the LiDAR projection errors. The initial sparse confidence input cannot exclude such noisy inputs. To solve this problem, Eldesokey et al. [21] further developed a self-supervised approach to estimate a continuous input confidence map for suppressing the disturbed measurements with a network. NCNN is also applied to RGB guided depth completion in [45], [110].

### 3.3 Training with Auxiliary Images

A few works smartly and implicitly utilize RGB information for unguided depth completion by introducing an auxiliary task of depth for reconstruction. For example, to overcome the lack of semantic cues, Lu et al. [73] employed an auxiliary learning branch in their framework. Instead of directly using an image as input, they only take a sparse depth map as input and simultaneously predict a reconstructed image and a dense depth map. The RGB images are only used in the training stage as a learning objective to encourage acquiring more complementary image features. A similar method is also seen in [130] where RGB and normal are used for auxiliary training. In [111], an auto-encoder is employed to generate RGB data in latent space, and then the auto-encoder predicts the final depth from it. This method is unsupervised and does not use denser depth maps as ground truths, showing inferior performance compared to [73]. Although these methods are RGB guided in training, they aim at performing unguided depth completion in inference. Therefore, we categorize them into unguided methods.

## 4 RGB Guided Depth Completion

Unguided methods usually underperform RGB guided methods and suffer from blurring effects and distortion of object boundaries. As studied in [46], depth maps of natural scenes can be decomposed into smooth surfaces and sharp discontinuities between them; the latter forms step edges in depth maps. This structure is a key property of depth maps. However, when depth maps are extremely sparse, prior information such as neighboring objects and sharp edges are significantly missing; therefore, it is even intractable to recover complete depth maps using CNNs.

Therefore, utilizing RGB information as an additional input is straightforward and reasonable. RGB images provide information about scene structures, including textures, lines and edges, to complement the missing cues of sparse depth maps, and encourage depth continuities inside smooth regions and discontinuities at boundaries. Moreover, they include some monocular cues, e.g., vanishing points [43], for promoting depth estimation. These benefits complement sparse depth maps.

Compared to unguided methods, RGB guided approaches typically have three advantages: i) they generally

outperform unguided methods in accuracy, ii) they are more robust to different sparsity levels, and iii) they gain more perceptually correct depth maps.

To date, different types of methods have been proposed, and they can be categorized into mainly five types: 1) early fusion models, 2) late fusion models, 3) explicit 3D representation models, 4) residual depth models, and 5) spatial propagation network (SPN) based models.

### 4.1 Early Fusion Models

Early fusion methods directly concatenate a sparse depth map and an RGB image before passing them through a deep model [17], [77], [87], or aggregate multi-modal features at the first convolutional layer of a model [51], [70], [121]. Previous methods of early fusion can be divided into two types: methods employing 1) encoder-decoder network and 2) two-stage coarse to refinement prediction.

#### *4.1.1 Encoder-decoder Networks*

This type of method utilizes a traditional encoder-decoder network (EDN) to solve the pixel-to-pixel regression problem. An early work is shown in [77] where Ma et al. proposed to accomplish depth completion from both a sparse depth map and its corresponding RGB image. Toward this end, they directly concatenated the RGB image and the sparse depth map and then fed them to an encoder-decoder network built on a ResNet-50 network [38]. This work also verified that RGB guided depth completion is more accurate and robust than unguided approach for different sparsity levels.

To better enforce the prediction to be consistent with the measurements, Qu et al. [88] replaced the last convolutional layer with a least squares fitting module. In this model, the extracted features obtained from the penultimate layer are treated as a set of bases, and the weights of these bases are obtained through a least squares fit on the depths at valid pixels. As discussed in the paper [88], the method is unable to handle extremely sparse input due to the lack of supervision with enough depth points.

Motivated by spatially-adaptive denormalization (SPADE) [86], Dmitry et al. [95] proposed to learn spatially-dependent scale and bias for normalized features. They introduced a novel decoder assembled with SPADE blocks with a modulation branch. The modulation branch takes the valid mask as input and predicts multi-scale modulation signals. These modulation signals are sent to the multiple SPADE blocks in the decoder at each spatial scale to update features. The method's effectiveness has been validated on both indoor depth enhancement and outdoor depth completion.

Instead of the direct concatenation, several approaches [51], [76], [132] used two separate convolutional units to extract features from RGB and depth input at the first layer of the encoder-decoder network, respectively. Then, the multi-modal features were concatenated and sent to the rest of the layers to obtain a complete depth map.

Early fusion methods built on EDN are straightforward. Compared to the late fusion strategy, they are good at model simplicity, yet underperform in accuracy.

#### *4.1.2 Coarse to Refinement Prediction*

Some methods employ a two-stage coarse to refinement prediction (C2RP) to achieve more accurate depth estimation. This kind of methods firstly estimates a coarse depth map in the first coarse prediction stage, then applies the second refinement prediction from the coarse depth map and the RGB image. For instance, Dimitrievski et al. [17] integrated a learnable morphological operator (two contraharmonic mean filter layers [78]) into a U-net [90] based framework. After the morphological operation, the predicted coarse depth map and the RGB image are passed through a U-net to get a refined output. Similarly, Hambarde et al. [36] proposed S2DNet which consists of two pyramid networks: S2DCNet and S2DFNet. The S2DCNet performs the first coarse prediction, and the S2DFNet performs the second refinement.

Unlike the above methods, several methods proposed to generate multiple maps in the coarse prediction stage. For instance, Chen et al. [10] generated a dense map with the nearest neighbor interpolation and a prior distance map between depth points based on a Euclidean distance transform of the validity mask. The dense map acts as a coarsely predicted map as explored in [17], and the distance map serves a similar role as the validity mask that informs the model about the valid depth points, but in a different manner from SACNN. As shown in [10], the inclusion of the distance map improves training stability.

Recently, Hedge et al. [39] proposed the DeepDNet. The assumption is that CNNs are considered to learn better features with uniform data rather than randomly distributed data. Therefore, they first convert the original sparse input into a grid sparse depth map with quad tree based preprocessing. Then, two coarse maps are generated by applying the nearest neighbor interpolation and Bi-cubic interpolation from the grid sparse map, respectively. Such random to uniform transformation gained a slightly better performance than [10] for synthesized depth maps on the NYU-v2 dataset. However, its effectiveness for more realistic scenarios, e.g., KITTI, remains unclear.

In [70], depth completion is decomposed into a first-stage relative depth estimation and a second-stage scale recovery problem. The final depth map is the multiplication of the relative depth map and its scale map. As argued in [70], such design reformulates the completion task in scale space, and thus is more robust for tackling the sparsity.

Notably, the performance of the C2RP highly relies on the quality of pre-estimated depth maps in the first stage of coarse prediction. Besides, the idea of rectifying from a coarse prediction is also frequently leveraged in subsequent studies, such as those built on SPNs and residual depth learning frameworks.

### 4.2 Late Fusion Models

Late fusion models usually employ two sub-networks to extract features from (i) RGB images using an RGB encoder network, and (ii) sparse depth inputs using a depth encoder network. The fusion is conducted at intermediate layers of the two sub-networks. Most of the previous methods exploit the late fusion strategy with various network structures. Specifically, they are categorized into three types: methods

employing 1) dual-encoder network, 2) double encoder-decoder network, and 3) global and local depth prediction.

#### 4.2.1 Dual-encoder Networks

Methods built on a dual-encoder network (DEN) commonly use an RGB encoder and a depth encoder for extracting multi-modal features. Then, these features are aggregated and fed into a decoder. DENs take a divide-and-conquer strategy that learns domain-specific features from RGB images and sparse depth maps with two separate encoders, respectively, then fuse them to form a correlational feature representation with a decoder.

In [54], Jaritz et al. introduced a two-branch encoder network based on a modified NASNet [141], where the intermediate features extracted from all encoders are directly concatenated and then outputted to a decoder. Notably, Jaritz et al. verified that the validity mask is not necessary for performance improvement for large networks. Instead of direct channel-wise concatenation, features extracted from the RGB encoder and the depth encoder are fused in element-wise summation in [92], [97].

Lately, more complicated fusion strategies have been explored. Fu et al. [25] improved the straightforward concatenation of RGB and depth features with an inductive fusion adapted from the conditional neural process [30]. Zhong et al. [138] suggested using the correlation between RGB and depth information. For this purpose, they proposed the CFCNet which extracts the most semantically correlated features from multi-modal inputs by applying deep canonical correlation analysis [126] between the sparse depth points and their corresponding pixels in RGB images.

The above approaches only fuse the outputted features from the RGB branch and depth branch at a single spatial scale. To establish a hierarchical joint representation, Zhang et al. [134] proposed a multi-scale adaptation fusion network (MAFN). The main contribution of MAFN is the adaptation fusion module (AFM) that incorporates features extracted from RGB and depth modalities and passes them to a neighbor attention module to enhance their local neighboring relational information. AFM is applied between the RGB and depth branches at multiple scales, as seen in Fig. 3.

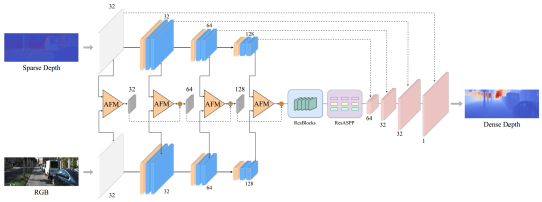



Fig. 3. The diagram of the MAFN. The framework is a DEN where features extracted from the RGB encoder and the depth encoder are fused with the adaption fusion (AFM) module at multi-scales. From [134].

Li et al. [63] introduced a cascaded hourglass network that consists of a branch (image encoder) used to extract features from images and three hourglass branches used to extract features from depth at different scales (1/4, 1/2, 1). The feature maps obtained from the image encoder at different scales are merged with the corresponding depth features by skip connection. The ground truth is downsampled to different scales to make use of the multi-scale supervision. Such design enables a significant drop in model complexity and improves inference efficiency.

To better tackle the sparsity, many works seek to exploit additional constraints to guide the learning process. A common solution is to apply epipolar constraints between temporally adjacent frames [15], [24], [99], [116], [117], [118], [119], or stereo pairs [97], [127]. Another constraint is adversarial loss which comes from adversarial training with the use of a generative adversarial network (GAN) [33]. Although these constraints provide unsupervised guidance to the models, they require additional inputs or other guidance networks during their training.

#### 4.2.2 Double encoder-decoder Networks

As discussed above, DEN-based methods usually consist of an RGB encoder, a depth encoder, and a decoder. The fusion is conducted between the two encoders. A double encoder-decoder network (DEDN) is an improvement of the dual-encoder network and further boosts completion performance. A vanilla DEDN contains two encoder-decoder networks. In like manner, one takes an image input, and the other takes sparse depth input. The image network is also called the guided network. For methods built on DEDN, the fusion is usually conducted between the decoder of the image branch and the encoder of the depth branch at multi-scales.

As a representative method depicted in Fig. 4, GuideNet [102] aims to learn a more effective fusion of RGB and depth features. Inspired by guided image filtering [37] and bilateral filtering [105], GuideNet introduced the guided convolution which automatically generates spatially-variant kernels from the image features and applies them to assign weights to the depth features. The guided convolution is applied to multi-scale image features. To reduce the computational complexity, motivated by MobileNet-V2 [93], the guided convolution is factorized into a channel-wise and a cross-channel convolution.

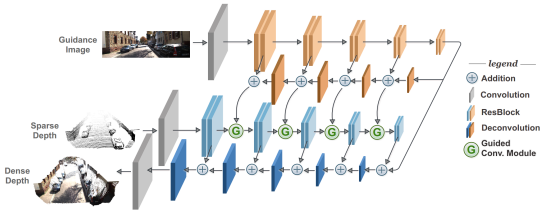



Fig. 4. The diagram of the GuideNet. The framework is a DEDN where the guided convolution learns fusion kernels from RGB features and applies them to depth features. From [102].

Inspired by [102] and [107], Schuster et al. [94] proposed sparse spatial guided propagation (SSGP) which combines image guided spatial propagation and sparsity convolution. SSGP is applicable to not only depth completion but also other interpolation problems such as optical flow and scene flow. More recently, Yan et al. [125] proposed RigNet with a novel repetitive design to handle blurry object boundaries and better recover scene structures. In RigNet, the branch used for extracting image features is implemented using a repetitive hourglass network (RHN), i.e., multiple encoder-decoder networks, to produce perceptually clear image features. The branch of RigNet used for extracting depth features is also a hourglass network stacked with a repetitive guidance module (RG). RG plays a similar role as the guided convolution [102] and is built on dynamic

convolution [8]. Since RG implements dynamic convolution repetitively, the convolution factorization proposed in [102] becomes less efficient. Thus, they designed an efficient guidance algorithm in which the kernel size in the channel-wise convolution drops from 3×3 to 1×1 by using global average pooling. RigNet achieves an extraordinary performance and currently ranks second on the KITTI depth completion dataset [107].

#### 4.2.3 Global and Local Depth Prediction

In several prior works, RGB and LiDAR data are referred to as global information, and the LiDAR data is referred to as local information. The global and local depth prediction (GLDP) methods employ a global network to infer depth from global information (global information is equivalent to early fusion of RGB images and sparse depths) and a local network to estimate depth from local information. The final dense depth map is obtained by merging the outputs of the global and local networks.

To exploit both the global and local features, a global depth and local depth map, as well as related confidence maps, were predicted in [108]. The confidence map predicted at each branch was used as a cross-guidance to refine the depth map predicted by the other branch. A similar method was also introduced in [62] where Lee et al. made two improvements. First, in order to extend the receptive field, they designed a residual atrous spatial pyramid (RASP) block to replace the traditional residual block. Second, unlike [108] where the confidence map was directly used to refine a depth map via element-wise multiplication, they introduced a new guidance module that applies both channel-wise and pixel-wise attention operations. The same framework was likewise used to address depth completion from the extremely sparse depths in order to explore depth completion from single-line depth maps in [72].

Technically, GLDP takes advantage of both early fusion and late fusion by using a global depth prediction network and a local depth estimation network. GLDP attains comparable performance to DEDN with few parameters.

### 4.3 Explicit 3D Representation Models

Most previous studies of RGB guided depth completion learn 3D geometric relationships in an implicit yet ineffective manner. Typically, the difficulty comes from the incapability of normal 2D convolution to capture the 3D geometric clues from the sparse input where the observed depth values are irregularly distributed. Hence, another type of previous approaches promotes explicit 3D representations (E3DR). Previous methods of this type can be classified into the methods employing 1) 3D-aware convolution, 2) intermediate surface normal representation, and 3) methods of learning geometric representations from point clouds.

#### 4.3.1 3D-aware Convolution

The insight behind 3D-aware convolution is that, since a depth point is correlated to its spatial neighbors, and there are many missing points irregularly distributed in the sparse input, instead of the standard convolutions, applying 3D-aware convolutions to the nearest neighbors of a depth point helps eliminate perturbations of missing values.

In 2D-3D FuseNet [9], features extracted from an RGB branch and a depth branch are fused by several 2D-3D fusion blocks that jointly learn 2D and 3D representations. The 2D-3D fusion block uses a multi-scale branch to extract appearance features in 2D grid space with normal convolution operations, and a branch to learn 3D geometric representations by applying two continuous convolutions [112] on K-nearest neighbors of a center point in 3D space. The idea of learning from spatially close K-nearest neighbors is then commonly employed in subsequent studies.

For instance, in the ACMNet [137], the nearest neighbors are identified similarly by comparing the spatial differences. Unlike [9], the non-grid convolution is implemented by graph propagation. As seen in Fig. 5, ACMNet has a DEDN structure where the encoder is composed of co-attention guided graph propagation modules (CGPMs), and the decoder is a stack of symmetric gated fusion modules (SGFMs). CGPM adaptively applies attention based graph propagation in both the image and depth encoders for multi-modality feature extraction, and SGFMs apply symmetric cross guidance between two decoders for multi-modality feature fusion.

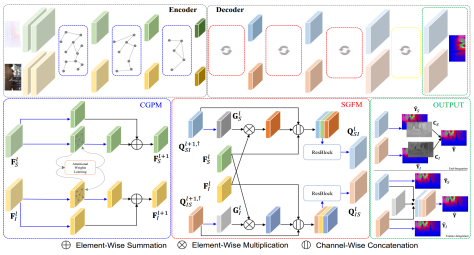



Fig. 5. The diagram of the ACMNet where the encoder uses several co-attention guided graph propagation modules (CGPMs) for multi-modality feature extraction and the decoder uses several symmetric gated fusion modules (SGFMs) for multi-modality feature fusion. From [137].

Xiong et al. [120] considered a graph model for depth completion and introduced a graph neural network (GNN) based depth completion algorithm. Note that the 3D graph of nearest neighbors is only constructed for a valid point in [9], [137], while it is constructed for each point from a dense depth pre-enhanced from a baseline model with a DEDN architecture in [120]. The method also studied and compared different sampling strategies for synthesizing sparse depth maps on the benchmark NYU-v2 dataset. The results show that quasi-random sampling [83] significantly outperforms random sampling[1]. These findings can help perform experiments of different sampling strategies on the indoor datasets for the depth completion task.

#### 4.3.2 Intermediate Surface Normal Representation

A few works utilized surface normal as an intermediate 3D representation of depth map and introduced methods employing surface normal guided completion. As studied in [47], [133], surface normal is a reasonably intermediate representation and can promote indoor depth enhancement. However, as pointed out by Qiu et al. [87] that reconstructing depth from normal in outdoor scenes is more sensitive to noise and occlusion; how to utilize surface normal in this

1. Previous methods of depth completion commonly validate their effectiveness on the NYU-v2 by synthesizing sparse depth maps via randomly sampling a few depth points.

case is still an open question. To address this issue, they proposed DeepLIDAR, a two-branch network consisting of a color pathway and a surface normal pathway depicted in Fig. 6. Both branches produce a dense depth map. The final depth map is obtained via the attention-based weighing of the outputs of the two pathways. In the surface normal branch, surface normal is utilized as the intermediate representation of the produced depth map.

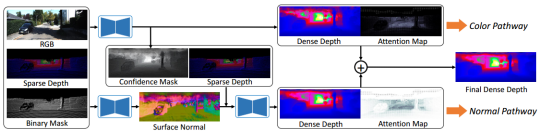



Fig. 6. The pipeline of the DeepLIDAR where surface normal is used as an intermediate representation of a depth map. From [87].

The use of surface normal is straightforward for the method proposed in [87]. As argued in [121], the relation between depth and surface normal can be established via the tangent plane equation in the camera coordinate system. By this intuition, Xu et al. [121] proposed the plane-origin distance that forces the consistency between depth and surface normal to regularize depth completion. Different from [87], the method also estimates a confidence map modeling as Laplace distribution to mitigate the effect of noise and applies a refinement network. Benefiting from the depth and normal consistency, they achieved comparable performance against [87] while using only about 20% of parameters.

#### 4.3.3 Learning from Point Clouds

Recently, a few studies directly learned geometric representations from point cloud, since it is a reliably strong prior of 3D structures. For example, Du et al. [20] proposed to first learn a geometric-aware embedding from point clouds with edge convolution [113]. Then, a DEN was utilized to perform depth completion from RGB images and geometric embeddings. Jeon et al. [55] also used a point cloud as input. By incorporating the attention mechanism into bilateral convolution [101], they designed an attention bilateral convolutional layer (ABCL) based encoder for feature extraction from 3D point clouds. Their framework also implements a DEN where a point cloud encoder is used to extract 3D features, and an image encoder is used to extract 2D features from an RGB image and a sparse depth input.

As shown in [20], [55], integrating point cloud into depth completion significantly boosts the model generalization accuracy in different environments. Compared to [55], the method of [20] achieves competing results with yet a simple and more lightweight framework.

### 4.4 Residual Depth Models

Residual depth models (RDMs) stress that the inferred depth maps should be precise in overall structure and faithful in local detail. Hence, the one-stage prediction procedure can be decoupled into the estimation of a dense map and a residual map. RDMs predict a depth map and a residual map, and their linear combination obtains the final depth. Through the prediction of the residual map, the model can refine the blur depth prediction and yield finer results on object boundaries.

These methods usually apply a two-stage coarse-to-refinement likewise prediction procedure. A simple application is shown in [64] where a sparse depth map is firstly completed to a dense map, and a residual map is then predicted. Finally, the element-wise summation of them generates the final depth map. Gu et al. [34] proposed DenseLiDAR, a similar method as shown in Fig. 7. In DenseLiDAR, a pseudo depth map with morphological operations is firstly predicted. Then, the pseudo depth map, the RGB image, and the sparse depth input are sent to a CNN to predict a residual map. Finally, the pseudo depth map is rectified with the residual map to yield the final depth map.

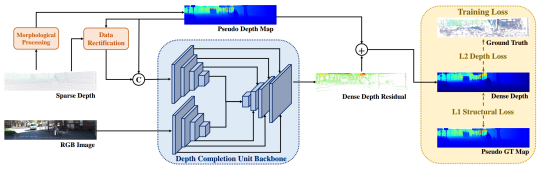



Fig. 7. The pipeline of the DenseLiDAR where depth completion is decomposed as learning of a coarse depth map and a residual depth map. From [34].

For other approaches, the improvement is derived from boosting the estimation of either the coarse depth map or the residual depth map. For instance, motivated by kernel regression, a differentiable kernel regression network was proposed to replace the hand-crafted interpolation for performing the coarse depth prediction from the sparse input in [67], [82]. In addition, FCFR-Net [68] implemented an energy-based operation for multi-modal feature fusion to boost the residual map learning.

Aiming at handling the uneven distribution and dealing with the outlier issue, Zhu et al. [140] introduced a novel uncertainty based framework which consists of two networks: a multi-scale depth completion block and an uncertainty attention residual learning network. Like other residual based methods, the former network yields a coarse prediction, and the later network performs refinement. The uncertainty based framework prevents over-fitting from outliers by relaxing constraints of the highly uncertain regions in the first completion stage and guides the network to generate the residual map in the refinement stage. Zhang et al. [135] combined the late fusion with residual learning and proposed a DEN-based multi-cue guidance network. Unlike other methods, the final depth is the combination of the sparse input and the estimated residual map.

### 4.5 SPN-based Models

An affinity matrix, also called a similarity matrix, expresses how close or similar data points are to each other. It is used to refine and gain a fine-grained prediction in vision tasks. In spatial propagation networks (SPN) [69], learning an affinity matrix is formulated as learning a group of transformation matrices. Following [69], [85], the affinity refinement process of SPN is defined by

$$x_{m,n}^{t} = w_{m,n}^{c} x_{m,n}^{t-1} + \sum_{i,j \in \mathcal{N}_{m,n}} w_{m,n}^{i,j} x_{i,j}^{t-1} \tag{4}$$

where $(m, n)$ and $(i, j)$ denote the coordinates of reference and neighbor pixels, respectively, and $\mathcal{N}_{m,n}$ is a set of neighbor pixels of the reference pixel at $(m, n)$. $t$ denotes the itera-

tion step of refinement. $w^c_{m,n}$ and $w^{i,j}_{m,n}$ are the affinity of the reference pixel and the affinity between the pixels at $(m, n)$ and $(i, j)$, respectively, where $w^c_{m,n} = 1 - \sum_{i,j \in N_{m,n}} w^{i,j}_{m,n}$.

Since a depth point is correlated to its neighbors, the SPN is reasonably applicable to depth regression problems, and a family of previous studies developed their algorithms based on SPNs. Cheng et al. proposed the pioneering convolutional spatial propagation network (CSPN) [13], [109] which is the first SPN-based model used for depth completion. Compared to the original SPN [69], CSPN has two major improvements. First, in SPN, a point is linked to three local neighbors from the nearest row or column, while in CSPN, a $3 \times 3$ local window is used to connect local neighbors. Second, CSPN efficiently propagates a local area in all directions via a convolution operation instead of propagating in different directions and integrating with max-pooling as SPN. The final value of a depth point is determined by its local neighbors via the diffusion process with the affinity matrix. Specifically, the network proposed in [77] is modified with skip connections and an additional output branch to generate the affinity matrix. Given a coarse predicted depth map and the affinity matrix, a CSPN is plugged into the network [77] for refinement, as shown in Fig. 8. The hyper-parameters including kernel size (size of local neighbors) and the number of iterations, need to be tuned by hyper-parameter search.

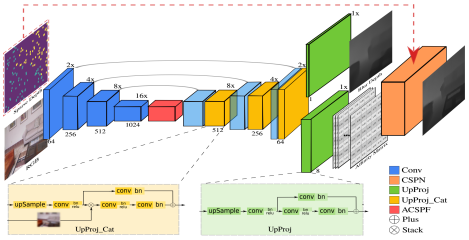



Fig. 8. The framework of CSPN based depth completion. The CSPN module is plugged into the network to rectify a coarsely predicted depth map. From [109].

To solve the difficulty of determining kernel sizes and iteration numbers, Cheng et al. further proposed CSPN++ [12] that enables an context aware CSPN (CA-CSPN) and an resource aware CSPN (RA-CSPN). For the implementation of CA-CSPN, various configurations of kernel sizes and numbers of iterations are first defined, and two extra hyper-parameters are introduced to weigh different kernel sizes and iterations adaptively. Thus, CA-CSPN consumes a large number of computational resources. To tackle this issue, RA-CSPN selects the best kernel size and number of iterations for each pixel by minimizing the computational resource usage. To this end, a computational cost function is aggregated to the optimization target to balance the trade-off between accuracy and training time.

While CSPN and CSPN++ mainly focus on the refinement from an existing encoder-decoder method [77], PENet [44] takes advantage of both SPN and late fusion models. PENet uses the DEDN structure where one network predicts from RGB images and sparse depths, and the other network predicts from sparse depths and a pre-densified depth map. A CSPN++ is then applied to the fused depth map of these predictions.

The above methods use fixed local neighbors for spatial propagation during affinity learning. However, this will involve the unnecessary use of irrelevant local neighbors. To address this problem, Park et al. proposed a non-local SPN [85] where non-local neighbors with affinities and a depth confidence map are learned, and the propagation is implemented through the deformable convolutions [139] on the K non-local neighbors. Besides, they also designed the confidence-incorporated affinity normalization module to encourage more affinity combinations and reduce the negative effect of unreliable depth values.

In [122], a deformable spatial propagation network (DSPN) is proposed to adaptively generate different receptive fields and affinity matrices for each pixel. Likewise, [65] introduced attention based dynamic SPN (DySPN) that can learn an adaptive affinity matrix by decoupling neighboring pixels based on their distances. Such attention mechanism recursively generates different attention maps to refine the affinity matrix and bring us the new state-of-the-art method for depth completion. DySPN currently ranks first on the KITTI depth completion benchmark [107].

Both methods of 3D-aware convolution and SPN-based models apply spatial constraints and can achieve superior performance. The former leverages spatial constraints during convolution while the latter forces spatial correlations via affinity-based post-refinement. In general, methods of 3D-aware convolution are more efficient than SPN-based models which is more time-consuming.

## 5 Learning Objectives for Training Models

Since depth completion and monocular depth estimation have the same target outputs, i.e., predicting dense depth maps, they share the same learning objectives, such as depth loss, surface normal loss, and photometric loss. In this section, we describe the learning objectives used in previous studies. A brief overview is given in Table 2 in which we will review the commonly used objectives in detail in the following sections.

### 5.1 Depth Consistency

Given a sparse input $Y'$, the predicted dense map $\hat{Y}$ where $\hat{Y} = N(Y'; \mathcal{W})$, and the semi-dense ground truth depth map $Y$, many works [54], [70], [97], [106], [127] used the $l_1$ loss (mean absolute error) between the predicted depth map and the ground truth depth map on valid pixels by

$$l_1 = \frac{1}{n} \sum_{i=1}^{n} \|\hat{Y}_i - Y_i\|_1 \tag{5}$$

where $\|\cdot\|_1$ denotes the $\ell_1$ norm, $\hat{Y}_i \in \hat{\mathcal{Y}}$ and $Y_i \in \mathcal{Y}$ denote the predicted depth and the ground truth depth at $i^{th}$ pixel, and $n$ is the total number of valid depth points from $\mathcal{Y}$. Also, most existing methods [20], [77], [134] used the $l_2$ loss, also known as root mean squared error (RMSE) by

$$l_2 = \frac{1}{n} \sum_{i=1}^{n} \|\hat{Y}_i - Y_i\|_2, \tag{6}$$

where $\|\cdot\|_2$ denotes the $\ell_2$ norm. Note that in many methods [62], [76], [77], [77], [87], the $l_2$ loss is referred to as MSE. Therefore, in this article, we do not technically distinguish between the RMSE and MSE when they are used as loss functions.

The $l_1$ loss treats each valid pixel equally, while the $l_2$ loss is more sensitive to outliers and usually penalizes distant depth points more heavily. To take advantage of both losses, some methods attempt to combine them from different aspects. For example, several approaches [36], [65] linearly combined them as a loss function. Van Gansbeke et al. [108] proposed focal-MSE where the mean absolute error was taken as a focal term for weighing the $l_2$ loss of depth. Also, some works [88], [110] used the Huber loss [49] combining $l_1$ and $l_2$ to reduce the influence of large errors. It is defined by

$$l_{huber} = \begin{cases} \frac{1}{n}\sum_{i=1}^{n} \frac{1}{2}(\hat{Y}_i - Y_i)^2, & |\hat{Y}_i - Y_i| \leq \delta \\ \frac{1}{n}\sum_{i=1}^{n} \delta\left(|\hat{Y}_i - Y_i| - \frac{1}{2}\delta\right), & |\hat{Y}_i - Y_i| > \delta \end{cases} \tag{7}$$

where $|\cdot|$ denotes the absolute value operator and $\delta$ is usually set to 1. Besides, a few studies [71], [110] employ the Berhu loss [84] which is a reversion of Huber loss defined by

$$l_{berhu} = \begin{cases} \frac{1}{n}\sum_{i=1}^{n} |\hat{Y}_i - Y_i|, & |\hat{Y}_i - Y_i| \leq \delta \\ \frac{1}{n}\sum_{i=1}^{n} \frac{(\hat{Y}_i - Y_i)^2 + \delta^2}{2\delta}, & |\hat{Y}_i - Y_i| > \delta \end{cases} \tag{8}$$

Fig. 9 visualizes the comparisons of MAE, MSE, Huber, and the Berhu loss functions for $\delta = 1$. As shown, the Huber norm acts as $l_2$ when the error is less than $\delta$ and acts as $l_1$ otherwise. On the other hand, the Berhu norm acts inversely to the Huber norm, i.e., acts as $l_1$ when the error is less than $\delta$ and acts as $l_2$ otherwise.

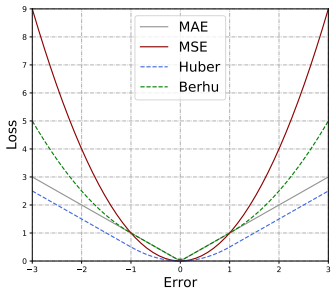



Fig. 9. The comparison of MAE, MSE, Huber and Berhu norm.

Another attempt for handling the above issue of regression is to formulate depth prediction as a classification problem as an early work [7] on monocular depth estimation. In this case, the depth range is discretized into a set of bins and a cross entropy loss is used. For depth completion, [51], [67] exploit this setting.

Besides the above discussed loss functions, to tackle the outliers and inherent noises of the sparse input, uncertainty aware learning objectives are also exploited. Uncertainty estimation [57] has been originally proposed to improve the robustness and accuracy of deep models. Inspired by [57], a couple of methods [21], [140] introduce the uncertainty driven depth loss function where the completion is posed as maximizing the posterior probability. Assuming the likelihood term $p(\hat{Y}_i|\sigma_i, Y_i)$ is modeled by a Gaussian distribution, following [21], [140], then

$$p(\hat{Y}_i|\sigma_i, Y_i) \approx \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{(\hat{Y}_i - Y_i)^2}{2\sigma_i^2}\right) \tag{9}$$

$\hat{Y}_i$ and $\sigma_i$ can be obtained via maximum likelihood estimation by

$$\begin{aligned} \hat{Y}_i, \sigma_i &= \underset{\hat{Y}_i, \sigma_i}{\operatorname{argmax}} \log p(\hat{Y}_i|\sigma_i, Y_i) \\ &= \underset{\hat{Y}_i, \sigma_i}{\operatorname{argmax}} -\frac{1}{2}\log(2\pi) - \log(\sigma_i) - \frac{(\hat{Y}_i - Y_i)^2}{2\sigma_i^2} \\ &= \underset{\hat{Y}_i, s_i}{\operatorname{argmax}} -\frac{1}{2}\log(2\pi) - \frac{1}{2}\log(s_i) - \frac{(\hat{Y}_i - Y_i)^2}{2s_i} \end{aligned} \tag{10}$$

where $s_i \triangleq \sigma_i^2$ denotes the uncertainty of prediction at the $i^{th}$ pixel. Given equation (10), the uncertainty driven depth loss for depth completion is defined by

$$l_{ud} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{(\hat{Y}_i - Y_i)^2}{s_i} + \log(s_i)\right) \tag{11}$$

In practice, an exponential function is usually applied to avoid division by zero during the training and the following uncertainty aware learning objective is used instead:

$$l_{ud} = \frac{1}{n}\sum_{i=1}^{n}(\exp^{-s_i}(\hat{Y}_i - Y_i)^2 + s_i). \tag{12}$$

In both works [21], [140], the uncertainty map $s$ is estimated with an additional branch within the depth completion framework.

### 5.2 Structural Loss Functions

A common problem of previous works is that the predicted depth maps suffer from blur effects and distorted boundaries. To overcome this problem, researchers proposed to apply regularization to scene structures by introducing loss functions of depth gradient, surface normal, and perceptual quality. Specifically, the gradient loss $l_{grad}$, is implemented by minimizing the mean absolute error [34], [67]. For surface normal difference denoted by $l_{normal}$, the negative cosine difference is commonly utilized [87], [121]. The effect of gradient and surface normal loss has been well studied in [42]. As shown in Fig. 10, the gradient loss contributes to penalizing errors emerging at the boundary of an object, while the surface normal loss can alleviate minor structural errors. Lastly, the structural similarity index measure (SSIM) loss [114], denoted by $l_{ssim}$, is penalized to ensure the perceptual quality [34], [131]. Since dense ground truth depth maps are required, previous methods using the structural loss need to generate pseudo dense ground truth maps if they are not available from training data.

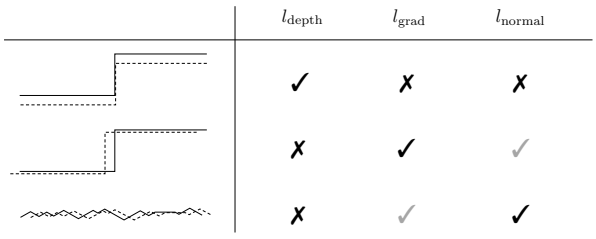



Fig. 10. Robustness of depth, gradient, and surface normal loss to depth differences. For simplicity, the solid and dotted lines denote two one-dimensional depth maps, respectively. It is observed that depth loss is insensitive to the shift and the occlusion of edges, while gradient and surface normal loss can handle these structural differences. From [42].

### 5.3 Smoothness Regularization

Smoothness regularization is utilized to suppress noises and ensure local smoothness for depth prediction. There

TABLE 2
A list of loss functions used for depth completion in previous works.

| Loss function | Type | Notation | Explanation |
|---|---|---|---|
| Depth Consistency | Supervised | $l_1$ | $l_1$ loss of depth on valid pixels, Eq.(5) |
| | | $l_2$ | $l_2$ loss of depth on valid pixels, Eq.(6). |
| | | $l_{huber}$ | Huber loss of depth on valid pixels, Eq.(7). |
| | | $l_{berhu}$ | Berhu loss of depth on valid pixels, Eq.(8). |
| | | $l_{ce}$ | Cross entropy of depth on valid pixels by formulating depth regression as a classification task. |
| | | $l_{ud}$ | Uncertainty driven loss of depth on valid pixels, Eq.(12). |
| Structural loss | Supervised | $l_{grad}$ | Gradient loss between the predicted depth map and the pseudo ground truth depth map. |
| | | $l_{normal}$ | Negative cosine difference of surface normal. |
| | | $l_{ssim}$ | SSIM loss between the predicted depth map and the pseudo ground truth depth map. |
| Smoothness regularization | Unsupervised | $l_{tv}$ [14] | Total variation of the predicted depth map. |
| | | $l_{smooth}$ | $l_1$ norm on second-order derivative of predicted depth map, Eq.(13) or edge-aware smoothness loss, Eq.(14). |
| Geometric constraint | Unsupervised | $l_{photo}$ | Photometric loss derived from temporally adjacent images or stereo images, Eq.(17). |
| | | $l_{stereo}$ | $l_2$ loss of depth between the predicted depth map and the pseudo ground truth depth map generated from stereo images. |
| Adversarial loss | Unsupervised | $l_{adv}$ | Adversarial loss between the predicted and the pseudo ground truth depth map, Eq.(18). |
| Others | Supervised | $l_{tp}$ [116] | $l_1$ loss between the prior (initial) depth map and the final estimated depth map. |
| | | $l_{cpn}$ [127] | $l_2$ loss between an estimated depth map and its reconstruction from the conditional prior network. |
| | | $l_{cosine}$ [70] | Cosine similarity between the predicted depth map and the pseudo ground truth depth map. |
| | | $l_{conf}$ [121] | Loss for learning the confidence map. |
| | | $l_p^{img}$ [73] | $l_p$ Loss for image reconstruction. |
| | | $l_{ur}$ [140] | Uncertainty aware loss for learning the residual depth map. |

are typically two frequently used learning objectives for imposing depth smoothness. The first objective used in [76], [97], [123], [132] is to minimize the $\ell_1$ norm on second-order derivative of predicted depth map by

$$l_{smooth} = \frac{1}{n}\sum_{i=1}^{n}\left(|\partial_x^2\,\hat{Y}_i| + |\partial_y^2\,\hat{Y}_i|\right) \tag{13}$$

where $\partial_x$ and $\partial_y$ denotes the gradients along the horizontal and vertical direction of the dense depth map. The second is the edge-aware smoothness loss used in [15], [92], [99], [116], [118], [119] that allows depth discontinuity at boundaries by

$$l_{smooth} = \frac{1}{n}\sum_{i=1}^{n}\left(\left|\partial_x\hat{Y}_i\right| e^{-|\partial_x I_i|} + \left|\partial_y\hat{Y}_i\right| e^{-|\partial_y I_i|}\right) \tag{14}$$

Besides, the total variation is also used in [14] for noise suppression.

### 5.4 Multi-view Geometric Constraints

One of the most challenging issues for depth completion is the lack of dense and high quality ground truth. To cope with this problem, researchers also attempt to seek solutions from the perspective of utilizing loss functions. Among them, temporal photometric loss obtained from consecutive images provides an unsupervised supervision signal [2] to guide depth completion.

Ma et al. [76] are the first that introduce photometric loss for depth completion. Based on the epipolar geometry, the predicted depth map of an image is warped to the nearby frame. Then, the differences at corresponding pixels are penalized. Formally, given an image $I_t$ and its temporally adjacent image $I_s$, where $s \in \{t-1, t+1\}$. Let $p_i$ denote a pixel at the pixel coordinate, the warping of $p_i$ from $I_t$ to $I_s$ is computed by

$$\begin{bmatrix}\hat{p}_i^\top \\ 1\end{bmatrix} = \phi\left(K(R\hat{Y}(p_i)K^{-1}\begin{bmatrix}p_i^\top \\ 1\end{bmatrix} + t)\right) \tag{15}$$

where $K$ denotes the camera intrinsic matrix, $\phi(a,b,c) = (a/c, b/c, 1)$ is an operation for scale alignment. $R, t$ are the rotation and translation respectively from the target frame $t$ to the adjacent frame $s$. $\hat{Y}(p_i)$ is the predicted depth of the pixel $p_i$ of the image $I_t$. The warped image is obtained by

$$\hat{I}_s(p_i) = I_s(\langle\hat{p}_i\rangle) \tag{16}$$

where $\langle\rangle$ denotes bilinear sampling operator. Then, the common photometric loss is defined by

$$l_{photo} = \frac{1}{m}\sum_{i=1}^{m}\|I_t(p_i) - \hat{I}_s(p_i)\|_1 \tag{17}$$

where $m$ denotes the number of warped pixels.

Researchers attempted to improve the above photometric loss from different perspectives in subsequent studies. The photometric loss is susceptible to moving objects. To alleviate this problem, Chen et al. [11] integrated a MaskNet into the self-supervised framework. The MaskNet predicts the masks of moving objects such that the influence of moving objects can be reduced. Also, to ensure perceptual consistency, Wong et al. [118], [119] integrated the SSIM difference [114] between warped and original images into the photometric loss.

Different approaches to calculating the photometric loss have also been explored. In [53], optical flow is used to estimate the relative pose between two consecutive frames, and a pose estimation net is used for this purpose in [132]. In [99], relative poses are calculated in feature spaces at multiple scales. Specifically, consecutive frames are sent to the FeatNet for multi-scale feature extraction. The relative pose is calculated with the Gauss-Newton algorithm [5] at each scale.

Wong et al. have put much effort into improving unsupervised depth completion. As pointed out in [117], the conventional use of the photometric loss treats each pixel equally. Unfortunately, this incurs significant meaningless errors at occluded regions. To address this issue, they proposed an adaptive weighting function [117] that acts as a flipped sigmoid function. The weights for the photometric loss are approximately equal to 1 at each pixel at the beginning, and will get smaller if the residual at certain pixel increases during the training procedure. In [118], the sparse

2. This signal is also called self-supervised signal in some studies [53], [76], [99]

depth is firstly densified by scaffolding operation [3], [6], and then passed to an EDN for refinement. In ScaffFusion [116], the parameter-free scaffolding operation used in [118] is replaced with a spatial pyramid pooling block as well as an encoder-decoder network. Then, ScaffFusion predicts a depth scale and a residual depth map for refinement. To increase generalizability on different cameras, KBNet [119] takes the calibration matrix as an additional input such that it can adjust to different cameras during inference.

A few previous works studied depth completion under the stereo setting except for the temporal photometric loss. When a stereo pair is available, as seen in [127], the multi-view photometric consistency can be derived in a different fashion. Besides, in order to handle the lack of supervision, stereo images are used to generate ground truth depths for missing pixels in [97]. However, despite these advantages, the stereo setting inevitably lowers the generalizability of these methods [97], [127] in practice.

### 5.5 Adversarial Loss

Several approaches also adopt adversarial loss to promote depth completion [1], [58], [106], [131]. In these works, a generator is used to infer a depth map from the RGB and sparse depth map, and a discriminator is used to distinguish between the reconstructed depth map and ground truth by

$$l_{\text{adv}} = \min_{G} \max_{D} \mathbb{E}[\log D(Y)] + \mathbb{E}[\log(1 - D(G(I, Y')))] \tag{18}$$

where $Y$ is dense ground truth which is usually obtained by other completion algorithms, $G$ and $D$ are the generator and discriminator, respectively.

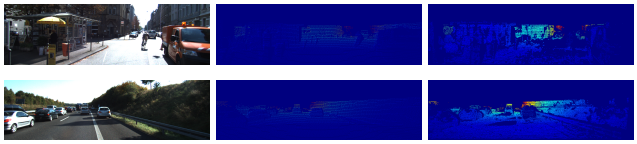
## 6 Datasets and Evaluation Metrics

In this section, we introduce the benchmark datasets commonly used in previous works in detail. We also comprehensively survey the related datasets for reference.

### 6.1 Real-world Datasets

**KITTI depth completion dataset [107]:** The KITTI dataset is a widely used large-scale outdoor dataset that contains over 93,000 semi-dense depth maps with the corresponding raw sparse LiDAR scans and RGB images. The training, validation, and test set have 86,000, 7,000 and 1,000 samples, respectively. The full resolution of images and depth maps can reach $1216 \times 352$, which is larger than most existing RGBD datasets. The raw LiDAR scans are captured by a Velodyne HDL-64E. To have a semi-dense ground truth depth map, Uhrig et al. [107] purified the raw data with the semi-global matching (SGM) and densified the sparse depth map by accumulating 11 laser scans.

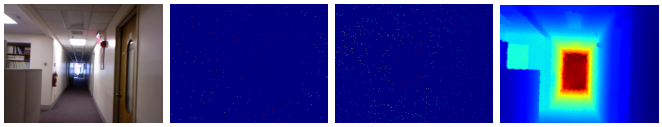
It should be noted that the ground truths can be used differently in implementing previous methods. The density of the original sparse depth maps is only about $5\%$ (as observed in Fig. 11 (b)), and the semi-dense ground truths provided by the KITTI benchmark can reach about $30\%$ (as visualized in Fig. 11 (c)). Most previous works take the denser ground truths to implement their methods, whereas several unsupervised approaches [116], [117], [118], [119], [127] assume that only original sparse depth maps are available. In this case, the depth consistency is only applied to those $5\%$ valid pixels.



(a) (b) (c)

Fig. 11. Sample images from the KITTI depth completion dataset [107]. (a) RGB images. (b) Raw sparse depth maps. (c) Ground truth depth maps.

**NYU-v2 [98]:** The NYU-v2 dataset consists of 464 indoor scenes with 408,000 RGBD images captured by Microsoft Kinect with an original resolution of $640 \times 480$. Previous studies of depth completion implement their methods by randomly selecting 200 (Fig. 12 (b)) or 500 depth points (Fig. 12 (c)) as sparse inputs. The total valid pixels are less than $1\%$ in both cases.



(a) (b) (c) (d)

Fig. 12. A sample image from the NYU-v2 dataset [98]. (a) An RGB image. (b) A spare depth map (200 points). (c) A sparse depth map (500 points). (d) The corresponding ground truth depth map.

**VOID [118]:** The VOID dataset contains 56 sequences collected with the Intel RealSense D435i camera from both indoor and outdoor scenes, in which 48 sequences (about 47,000 frames) are designed for training, and the rest of 8 sequences are used for testing. The resolution of each frame is $640 \times 480$. Each sequence has three different density levels with 1500, 500, and 150 points. This dataset was employed to evaluate the methods in [92], [116], [117], [118], [119].

**DenseLivox [130]:** The DenseLivox dataset is collected with a cheaper Livox LiDAR with much denser depth maps (the density is $88.3\%$) than KITTI. DenseLivox also provides some extra data like bound-occlusion and normal. This dataset was employed to evaluate the method in [130].

### 6.2 Synthetic Datasets

**SYNTHIA [91]:** The SYNTHIA is captured in a virtual city that includes street blocks, highways, suburban areas, and other common objects and has four different appearances corresponding to four seasons in reality. Different lighting conditions are applied to improve the diversity of virtual RGB images. The dataset has two complementary sets with the image resolution of $960 \times 20$, *SYNTHIA-Rand* and *SYNTHIA-Seqs*. The former (13,400 frames) is obtained randomly within the city, and the latter (200,000 frames) is captured from a virtual vehicle across different seasons. This dataset was employed to evaluate the methods in [54], [88].

**Aerial depth [104]:** The Aerial depth is a virtual outdoor dataset specially designed for simulating data captured in UVA working conditions. The dataset contains 83797 RGB and depth images from 18 virtual 3D models, and 67435 of them are selected for training and the rest for validation. This dataset was employed to evaluate the method in [104].

**Virtual KITTI [29]:** This dataset is a virtual version of the KITTI dataset. Five videos of the KITTI

TABLE 3
Summary of essential characteristics of existing unguided methods on the KITTI dataset. For denoting the loss function, we omit the coefficient of each loss term for simplicity. S and U denotes supervised learning and unsupervised learning of models, respectively.

| Method | Publication | Year | Type | Loss Function | Learning | RMSE (mm) | Params (M) | Platform | Code |
|---|---|---|---|---|---|---|---|---|---|
| SI-CNN [107] | 3DV | 2017 | SACNN | $l_2$ | S | 1601.33 | 0.025 | TensorFlow | ✓ |
| DCCS [14] | ACCV | 2018 | SACNN | $l_2 + l_{tv}$ | S | 1325.37 | 0.0017 | TensorFlow | ✓ |
| HMS-Net [48] | TIP | 2019 | SACNN | $l_2$ | S | 937.48 | - | - | - |
| NConv-CNN [22] | BMVC | 2018 | NCNN | $l_{berhu}$ | S | 1268.22 | 0.00048 | PyTorch | ✓ |
| pNCNN [21] | CVPR | 2020 | NCNN | $l_{ud}$ | S | 960.05 | 0.67 | PyTorch | ✓ |
| DCAE [111] | WACVW | 2022 | TwAI | $l_1 + l_1^{img}$ | S&U | 1464.69 | 2.29 | PyTorch | - |
| IR L1 [73] | CVPR | 2020 | TwAI | $l_1 + l_2^{img}$ | S | 915.86 | 11.63 | PyTorch | - |
| IR L2 [73] | CVPR | 2020 | TwAI | $l_2 + l_2^{img}$ | S | **901.43** | 11.63 | PyTorch | - |

(0001/0002/0006/0018/0020) are cloned through the unity engine. The dataset consists of 35 virtual videos (about 17000 frames). Each cloned virtual video is further modified to obtain 7 variations. The modification includes changing features of the objects, the camera's position and orientation, and the lighting condition. This dataset was employed to evaluate the methods in [55], [88], [97], [116].

**SceneNet RGB-D [79]:** This dataset contains 5 Million RGBD indoor images from over 15,000 synthetic trajectories with $320 \times 240$ image resolution. Each trajectory has 300 rendered frames. Due to ray-tracing, the generated images can reach the real-photo level quality. This dataset was employed to evaluate the method in [116].

### 6.3 Evaluation Metrics

Depth completion and monocular depth estimation generally share the same evaluation metrics. We list the most commonly used measures as follows:

- **RMSE**: Root mean squared error defined in Eq.(6).
- **MAE**: Mean absolute error defined in Eq.(5).
- **iRMSE**: RMSE of the inverse depth, defined by $\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{Y_i}-\frac{1}{\hat{Y}_i}\right)^2}$.
- **iMAE**: MAE of the inverse depth, defined by $\frac{1}{n}\sum_{i=1}^{n}\left|\frac{1}{Y_i}-\frac{1}{\hat{Y}_i}\right|$.

The above four measures are metrics commonly used to evaluate models in the KITTI benchmark. Among them, KITTI ranks algorithms in competitions in the order of RMSE. Thus, many previous methods have aimed to choose RMSE ($l_2$) as a loss function to train models. Besides, several metrics are also frequently used in many methods for depth evaluation, such as

- **REL**: Mean relative error defined by $\frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i-\hat{Y}_i|}{\hat{Y}_i}$.
- $\boldsymbol{\delta}$: Thresholded accuracy defined by $\max(\frac{Y_i}{\hat{Y}_i}, \frac{\hat{Y}_i}{Y_i}) = \delta < \tau$ where $\tau$ is a given threshold.

REL and $\delta$ are commonly used for evaluation of models on indoor datasets, e.g., NYU-v2.

Evaluation of depth maps is an open issue. The above metrics cannot precisely measure the quality of reconstructed compositional patterns such as objects. Therefore, researchers also attempted to propose new evaluation metrics. In [42], object boundaries extracted from the depth map are measured. Koch et al. [60] introduced the planarity error and location accuracy of depth boundaries. Jiang et al. [56] proposed two metrics for quantifying the flatness of planes and the straightness of lines for depth maps. However, owing to the lack of dense ground truth, such metrics are still difficult to be applied to depth completion.

## 7 Experimental Analyses

In this section, we compare and review previous methods from comprehensive aspects. Specifically, we select some representative works from each category and elucidate their major characteristics, including network structure, loss function, learning strategy, model performance, etc. Table 3 and Table 4 show a comparison of existing unguided and RGB guided methods on the KITTI dataset, respectively, where the RMSE values are taken from either the public KITTI benchmark or the original papers. Table 5 shows a comparison of RGB guided methods on the NYU-v2 dataset.

Besides, Table 6 shows a comparison of several methods on the VOID dataset. Note that we use **S**, **U**, and **S&U** to denote supervised methods, purely unsupervised methods without applying depth consistency, and unsupervised methods with depth consistency on only valid depth points from sparse depth inputs in Table 3, 4, 5, and 6. We use the RMSE metric for performance comparison. In the following sections, our findings are summarized.

### 7.1 Main Characteristics of Existing Methods

1) A relatively smaller number of prior works employ the route of performing completion from the sparse depth input. In comparison, more recent works are RGB-guided, among which the majority route is to perform late fusion of RGB and depth images instead of early fusion.
2) PyTorch is the most popular deep learning library for implementing depth completion methods. The overwhelming majority of previous studies implement their methods with PyTorch.
3) KITTI is the most popularly used evaluation benchmark. Almost all leading methods provide results on this dataset. Moreover, NYU-v2 is the second most popular dataset. Since depth maps of NYU-v2 are captured by Kinect, previous works implement their methods by randomly and uniformly sampling 200 or 500 pixels as valid depth points. Besides, VOID is also a frequently used benchmark for recent unsupervised methods.
4) More complicated neural network modules have been recently developed to advance the performance of depth completion models. For example, many methods propose to embed surface normal, affinity matrices, and residual maps into their network models.

TABLE 4
Summary of essential characteristics of selected existing RGB guided methods on the KITTI dataset. For denoting loss functions, we omit the coefficient of each loss term for simplicity. S and U denotes supervised learning and unsupervised learning of models, respectively. Accordingly, the top and bottom parts of the table show the supervised and unsupervised methods implemented for depth completion, respectively.

| Method | Publication | Year | Type | Loss Function | Learning | RMSE (mm) | Params (M) | Platform | Code |
|---|---|---|---|---|---|---|---|---|---|
| 3coef [51] | CVPR | 2019 | EFM/EDN | $l_{ce}$ | S | 965.87 | - | TensorFlow | ✓ |
| EncDec-Net[EF] [110] | TPAMI | 2019 | EFM/EDN | $l_1$ | S | 965.45 | 0.484 | Pytorch | ✓ |
| Qu et al. [88] | WACV | 2020 | EFM/EDN | $l_{huber}$ | S | 998.80 | - | Pytorch | - |
| Morph-Net [17] | ACIVS | 2018 | EFM/C2RP | $l_2$ | S | 1045.45 | - | Matlab | ✓ |
| S2DNet [36] | TCI | 2020 | EFM/C2RP | $l_1 + l_2$ | S | 830.57 | - | PyTorch | - |
| Long et al. [70] | JVCIR | 2021 | EFM/C2RP | $l_1 + l_{cosine}$ | S | 776.13 | - | - | - |
| Spade-RGBD [54] | 3DV | 2018 | LFM/DEN | iMAE | S | 917.64 | 5.3 | - | - |
| MS-Net[LF] [110] | TPAMI | 2019 | LFM/DEN | $l_1$ | S | 859.22 | 0.356 | PyTorch | ✓ |
| MSG-CHN [63] | WACV | 2020 | LFM/DEN | $l_2$ | S | 762.19 | 1.2 | PyTorch | ✓ |
| MAFN [134] | IJCNN | 2020 | LFM/DEN | $l_2$ | S | 803.50 | - | - | - |
| Ryu et al. [92] | RAL | 2021 | LFM/DEN | $l_2 + l_{smooth}$ | S | 809.09 | 1.9 | - | - |
| DVMN [89] | ITSC | 2021 | LFM/DEN | $l_2 + l_{smooth}$ | S | 776.31 | - | - | - |
| GuideNet [102] | TIP | 2020 | LFM/DEDN | $l_2$ | S | 736.24 | 62.62 | PyTorch | ✓ |
| SSGP [94] | WACV | 2021 | LFM/DEDN | $l_2$ | S | 838.22 | 4.61 | - | - |
| RigNet [125] | Arxiv | 2021 | LFM/DEDN | $l_2$ | S | 712.66 | - | PyTorch | - |
| Van et al. [108] | MVA | 2019 | LFM/GLDP | focal-MSE | S | 772.87 | 2.545 | PyTorch | ✓ |
| CrossGuidance [62] | Access | 2020 | LFM/GLDP | $l_2$ | S | 807.42 | 5.4 | PyTorch | - |
| 2D-3D FuseNet [9] | ICCV | 2019 | E3DR/3DAC | $l_1 + l_2$ | S | 752.88 | 1.898 | - | - |
| ACMNet [137] | TIP | 2021 | E3DR/3DAC | $l_2$ | S | 732.99 | 4.9 | PyTorch | ✓ |
| DeepLiDAR [87] | CVPR | 2019 | E3DR/ISNR | $l_2 + l_{normal}$ | S | 758.38 | 144 | PyTorch | ✓ |
| PwP [121] | ICCV | 2019 | E3DR/ISNR | $l_2 + l_{normal} + l_{conf}$ | S | 777.05 | 28.99 | PyTorch | - |
| ABCD [55] | RAL | 2021 | E3DR/LfPC | $l_2$ | S | 764.61 | 32.93 | PyTorch | - |
| Du et al. [20] | Arxiv | 2022 | E3DR/LfPC | $l_2$ | S | 773.90 | 4.189 | PyTorch | ✓ |
| FCFR-Net [68] | AAAI | 2021 | RDM | $l_2$ | S | 735.81 | - | - | - |
| DenseLiDAR [34] | RAL | 2021 | RDM | $l_2 + l_{grad} + l_{ssim}$ | S | 755.41 | - | - | - |
| Zhu et al. [140] | AAAI | 2022 | RDM | $l_{ud} + l_{ur}$ | S | 751.59 | - | PyTorch | - |
| CSPN [13] | ECCV | 2018 | SPM | $l_2$ | S | 1019.64 | 17.41 | PyTorch | ✓ |
| CSPN++ [12] | AAAI | 2020 | SPM | $l_2$ | S | 743.69 | 26 | - | - |
| NLSPN [85] | ECCV | 2020 | SPM | $l_1 + l_2$ | S | 741.68 | 25.84 | PyTorch | ✓ |
| PENet [44] | ICRA | 2021 | SPM | $l_2$ | S | 730.08 | - | PyTorch | ✓ |
| DySPN [65] | AAAI | 2022 | SPM | $l_1 + l_2$ | S | **709.12** | - | PyTorch | - |
| SS-S2D (d) [76] | ICRA | 2019 | EFM/EDN | $l_{photo} + l_{smooth}$ | U | 1299.85 | 26.1 | PyTorch | ✓ |
| DFineNet [132] | Arxiv | 2019 | EFM/EDN | $l_2 + l_{photo} + l_{smooth}$ | S&U | 943.89 | - | PyTorch | ✓ |
| DDP [127] | CVPR | 2019 | LFM/DEN | $l_1 + l_{cpn} + l_{photo} + l_{ssim}$ | S&U | 1263.19 | 18.8 | TensorFlow | - |
| DFuseNet [97] | ITSC | 2019 | LFM/DEN | $l_2 + l_{stereo} + l_{smooth}$ | S&U | 1206.66 | - | PyTorch | ✓ |
| VOICED [118] | RAL | 2020 | LFM/DEN | $l_1 + l_{photo} + l_{smooth}$ | S&U | 1169.97 | 9.7 | TensorFlow | - |
| AdaFrame [117] | RAL | 2021 | LFM/DEN | $l_1 + l_{photo} + l_{smooth}$ | S&U | 1125.67 | 6.4 | PyTorch | ✓ |
| ScaffFusion-S&U [116] | RAL | 2021 | LFM/DEN | $l_1 + l_{photo} + l_{smooth} + l_{tp}$ | S&U | **847.22** | 7.8 | TensorFlow | ✓ |
| ScaffFusion-U [116] | RAL | 2021 | LFM/DEN | $l_{photo} + l_{smooth} + l_{tp}$ | U | 1121.89 | 7.8 | TensorFlow | ✓ |
| KBNet [119] | ICCV | 2021 | LFM/DEN | $l_1 + l_{photo} + l_{smooth}$ | S&U | 1069.47 | 6.9 | PyTorch | ✓ |
| Song et al. [99] | TITS | 2021 | LFM/DEN | $l_1 + l_{photo} + l_{smooth}$ | S&U | 1216.26 | 9.7 | PyTorch | ✓ |

5) The learning objectives identified for depth completion tasks are intuitive and relatively straightforward to optimize. For example, many methods penalize just $l_1$ or $l_2$ loss of depth maps, and still achieve good performance.

### 7.2 Unguided and Guided Methods

There are two benefits of unguided methods. First, unguided methods are more robust to environments with light or weather changes since they only take sparse depth maps as inputs. Moreover, for the same reason, they are more computationally efficient. However, unguided methods show inferior performance due to the lack of semantic cues and the irregular distribution of captured depth points. As seen in Table 3, the best unguided method [73] yields RMSE of 901.43 millimeters on the KITTI dataset. Note that [73] also uses RGB images to guide model training. The best result obtained using an RGB-free method in both the training and inference stage is demonstrated in [48] with RMSE of 937.48. On the other hand, as seen in Table 4, the best RGB guided method, i.e., DySPN, demonstrates a significantly better result with RMSE of 709.12. Moreover, many RGB guided methods can easily beat the best unguided approach. Specifically, except for 3coef [51], EncDec-Net[EF] [110], Morph-Net [17] and CSPN [13], all other RGB guided methods with supervised learning outperform HMS-Net, showing the advance of leveraging RGB information. Another difference is that unguided methods cannot utilize additional unsupervised losses derived from images, e.g., photometric loss.

### 7.3 Comparison of RGB Guided Methods

For RGB guided methods, from Table 4, we can observe the following results:

- Early fusion models generally underperform other types of methods.

TABLE 5
Summary of essential characteristics of existing RGB guided methods on the NYU-v2 dataset.

| Method | Publication | Year | Type | Loss Function | Learning | RMSE (mm) | Params (M) | Platform | Code |
|---|---|---|---|---|---|---|---|---|---|
| 3coef [51] | CVPR | 2019 | EFM/EDN | $l_{ce}$ | S | 131 | - | TensorFlow | ✓ |
| Long et al. [70] | JVCIR | 2021 | EFM/EDN | $l_1 + l_{cosine}$ | S | 100 | - | - | - |
| DFuseNet [97] | ITSC | 2019 | LFM/EDN | $l_2 + l_{stereo} + l_{smooth}$ | S&U | 219 | - | PyTorch | ✓ |
| MS-Net[LF] [110] | TPAMI | 2019 | LFM/DEN | $l_{huber}$ | S | 129 | 0.356 | PyTorch | ✓ |
| KBNet [119] | ICCV | 2021 | LFM/DEN | $l_1 + l_{photo} + l_{smooth}$ | S&U | 105 | 6.9 | PyTorch | ✓ |
| SelfDeco [15] | ICRA | 2021 | LFM/DEN | $l_1 + l_{photo} + l_{smooth}$ | S&U | 178 | | PyTorch | - |
| GuideNet [102] | TIP | 2020 | LFM/DEDN | $l_2$ | S | 101 | 62.62 | PyTorch | ✓ |
| RigNet [125] | Arxiv | 2021 | LFM/DEDN | $l_2$ | S | **90** | - | PyTorch | - |
| PwP [121] | ICCV | 2019 | E3DR/ISNR | $l_2 + l_{normal}$ | S | 112 | 28.99 | PyTorch | - |
| DeepLiDAR [87] | CVPR | 2019 | E3DR/ISNR | $l_2 + l_{normal}$ | S | 115 | 144 | PyTorch | ✓ |
| ACMNet [137] | TIP | 2021 | E3DR/3DAC | $l_2$ | S | 105 | 1.35 | PyTorch | ✓ |
| FCFR-Net [68] | AAAI | 2020 | RDM | $l_2$ | S | 106 | - | - | - |
| KernelNet [67] | TIP | 2021 | RDM | $l_1 + l_{ce} + l_{grad}$ | S | 111 | - | PyTorch | ✓ |
| CSPN [13] | ECCV | 2018 | SPM | $l_2$ | S | 117 | 17.41 | PyTorch | ✓ |
| CSPN++ [12] | AAAI | 2020 | SPM | $l_2$ | S | 115 | 26 | - | - |
| NLSPN [85] | ECCV | 2020 | SPM | $l_1$ | S | 92 | 25.84 | PyTorch | ✓ |
| DySPN [65] | AAAI | 2022 | SPM | $l_1 + l_2$ | S | **90** | - | PyTorch | - |

- For later fusion approaches, although a considerable number of methods are built on DEN, approaches [102], [125] based on DEDN demonstrate more significant performance improvement.
- Explicit 3D representation methods, SPN-based methods, and residual depth methods show more advanced performance and generally outperform other approaches.

More specifically, the Top-10 performing methods on the KITTI dataset are (i) four SPN-based models; DySPN [65], PENet [44], NLSPN [85], and CSPN++ [12], (ii) two residual depth models; FCFR-Net [68] and [140], (iii) two late fusion methods built on DEDN; RigNet [125] and GuideNet [102], and (iv) two explicit 3D representation models; ACMNet [137] and 2D-3D FuseNet [9]. Based on that, we can say that the naive fusion strategy such as aggregating inputs at an early stage or concatenating features extracted by a dual-encoder network in late stage is not sufficient for achieving satisfactory performance. The common feature of the Top-10 performing methods is that they propose to either explicitly model geometric relationship of depth points by applying 3D-aware convolution as ACMNet and 2D-3D FuseNet, refinement with residual depth map as residual depth models and affinity matrix as SPN-based methods; or learn more effective guided kernel to weigh depth features with a complicated network design as RigNet and GuideNet.

Consistent results are also observed in analyses on the NYU-v2 dataset. As shown in Table 5, the best results are demonstrated by DySPN and RigNet. Besides, GuideNet, ACMNet, FCFR-net, and NLSPN also show improved performance compared to other methods.

Intuitively, the performance of depth completion has the potential to be further improved by aggregating core technical components of the above methods. For instance, by taking advantage of 3D representation networks and spatial propagation networks, we can not only learn the 3D relationship within the model in a feature space but also apply post-refinement with an affinity matrix in output space. In addition, we can also incorporate a DEDN with guided kernel learning into residual depth learning models. Such combinations are straightforward, nevertheless, can be considered in practical applications to pursue high accuracy.

### 7.4 Results of unsupervised Approaches

The bottom of Table 4 shows methods with unsupervised photometric loss. Results of purely unsupervised methods (without using depth consistency loss) are calculated by aligning the scale of the predicted depth map to the scale of ground truth. First, for methods without leveraging depth consistency, such as SS-S2D (d) [76] and ScaffFusion-U [116], we can see that purely unsupervised methods demonstrate unsatisfactory performance. Second, we also observe that their performances are still inferior to supervised methods even leveraging both depth consistency loss and additional photometric loss. As also discussed in Sec. 6.1, this is because these methods [15], [99], [116], [119], [127] use sparser depth maps as ground truths with a density of $5\%$ than supervised methods with a density of $30\%$. Among these methods, ScaffFusion [116], DFineNet [132], and KBNet [119] demonstrate better performance than other approaches. Similar results are also observed in Table 5 where supervised approaches outperformed unsupervised methods. This is not surprising since supervised methods could use pixel-wise ground truth depth maps for training on the NYU-v2 dataset.

Table 6 provides results for several approaches evaluated on the VOID dataset. As observed, the performance has been continuously improved from the early VOICED [118] to the recent KBNet [119]. Most works of Wong et al. apply pre-densification to sparse depth inputs, such as employing a learning based spatial pyramid pooling (SPP) block in [116]. As argued in [119], the max-pool layers in the SPP block tend to lose details in close range. Therefore, in KBNet, both max-pooling and min-pooling are implemented to ensure that the network can extract more comprehensive depth features. We believe this plays an important role in improving the KBNet's accuracy.

Overall, KBNet [119] and ScaffFusion [116] achieved the first and the second highest accuracy among unsupervised approaches on the VOID dataset. However, they still un-

TABLE 6
Summary of essential characteristics of existing RGB guided methods on the VOID dataset. * and ** denote results taken from [119] and paperswithcode [3], respectively.

| Method | Publication | Year | Type | Loss Function | Learning | RMSE (mm) | Params (M) | Platform | Code |
|---|---|---|---|---|---|---|---|---|---|
| SS-S2D* [76] | ICRA | 2019 | EFM/EDN | $l_1 + l_{photo} + l_{smooth}$ | S&U | 243.84 | 27.8 | - | - |
| DDP* [127] | CVPR | 2019 | LFM/DEN | $l_1 + l_{cpn} + l_{photo} + l_{ssim}$ | S&U | 222.36 | 18.8 | - | - |
| NLSPN** [85] | ECCV | 2020 | SPN | $l_1 + l_2$ | S | 79.12 | 25.84 | - | - |
| VOICED [118] | RAL | 2020 | LFM/DEN | $l_1 + l_{photo} + l_{smooth}$ | S&U | 146.40 | 9.7 | TensorFlow | - |
| Ryu et al. [92] | RAL | 2021 | LFM/DEN | $l_2 + l_{smooth}$ | S | 181.42 | - | - | - |
| AdaFrame [117] | RAL | 2021 | LFM/DEN | $l_1 + l_{photo} + l_{smooth}$ | S&U | 135.93 | 6.4 | PyTorch | ✓ |
| ScaffFusion [116] | RAL | 2021 | LFM/DEN | $l_1 + l_{photo} + l_{smooth} + l_{tp}$ | S&U | 119.14 | 7.8 | TensorFlow | ✓ |
| KBNet [119] | ICCV | 2021 | LFM/DEN | $l_1 + l_{photo} + l_{smooth}$ | S&U | 95.86 | 6.9 | PyTorch | ✓ |

derperform the supervised NLSPN. It reveals that the photometric loss used by current unsupervised approaches is still not fully reliable and accurate as they are vulnerable to outliers, e.g., dynamic objects, sky, and transparent objects, that are ubiquitous in real-world scenarios.

## 8 Open Challenges and Future Directions

### 8.1 Depth Mixing Problem

The depth mixing problem, also called the depth smearing problem, is attributed to the difficulty of correctly identifying pixels near object boundaries, and usually causes blurry edges and artifacts. In order to alleviate this problem, 3coef. [51] formulates depth completion as a one-hot encoding problem by dividing a depth map into a set of bins with fixed depth ranges. Imran et al. [52] isolate the foreground and background depths in occlusion-boundary regions and models them, respectively. NLSPN [85] makes the network learn non-local relative neighbors such that the pixels can be separated during an iterative propagation. A more simple way of achieving this separation process is to leverage the K-nearest algorithm [9], [124], [137]. Besides, a boundary consistency network was added after depth completion to encourage predicting more sharp and clear boundaries [47], [103]. However, this problem is still difficult for depth estimation tasks and needs to be continuously investigated.

### 8.2 Flawed Ground Truth

Another problem is the existence of defects in ground truth depths. First, unlike semantic segmentation, none of the existing real-world datasets can provide pixel-wise ground truth because of the limitation of depth sensors. Although many existing methods are trained in a supervised way, most pixels cannot be sufficiently supervised. Second, the semi-dense annotations are not entirely reliable due to outliers caused by occlusions, dynamic objects, etc. To overcome the sparsity problem, some researchers [76], [99] turn to self-supervised frameworks to alleviate the lack of ground truth depths. To cope with the second problem, Zhu et al. [140] handle outliers by incorporating uncertainty estimation into the depth completion network. Besides, a few works [1], [131] leverage synthetic datasets for model training. However, the domain gap between real-world and synthetic data prevents a wide application of these methods. Despite the above efforts made by previous studies, it is still an open issue how to exclude the effects of unreliable depths, and there are still lots of room for improvement.

3. https://paperswithcode.com/sota/depth-completion-on-void

### 8.3 Lightweight Networks

Most previous methods have complex network structures with a large number of parameters. Moreover, many of them take two-stage coarse-to-refinement prediction. Thus, these methods are time-consuming and require high usage of hardware resources. However, for applications such as autonomous driving and robotic navigation, computation resources are limited and real-time inference is required. Although a few prior studies [2], [103], [110], [124] have partially considered the real-time inference problem, they suffer from inferior performance. Besides, the network design is essentially empirical. Following advances in monocular depth estimation, we can further apply several techniques, such as applying knowledge distillation [40], network compression [115], and neural architecture search [50]. Without sacrificing too much accuracy, developing lightweight methods with fast inference speed has enormous potential for real-world deployment, thus, is a valuable and practical research point in future work.

### 8.4 Un-/self-supervised Frameworks

As discussed before, un-/self-supervised learning frameworks are solutions commonly employed in the absence of dense ground truths. As discussed in Sec. 7.4, the accuracy of current un-/self-supervised methods is still lower compared to supervised methods because they apply depth consistency to only valid depth points from sparse inputs and cannot leverage ground truth depth points as many as used by supervised methods. On the other hand, the photometric loss will only be effective when the predicted depth maps are close enough to the ground truth. However, that is still challenging due to the fact that the photometric loss is particularly susceptible to noises, moving objects, texture-less regions. Thus, there is much room for further improvement of unsupervised methods. Since this kind of methods is not robust to dynamic objects, distant regions, etc., the improvements can be brought by leveraging more effective network structures for performing auxiliary tasks, such as pose estimation and outlier removal.

### 8.5 Loss Functions and Evaluation Metrics

Employment of proper loss functions is also critical to achieving satisfactory performance for depth completion. Commonly used loss functions are usually defined by a weighted sum of $l_2$ or $l_1$ loss functions with other auxiliary loss functions, e.g., smoothness loss and SSIM loss.

However, as discussed in [51], both $l_1$ and $l_2$ loss functions have their own drawbacks. The choice of them is usually dataset dependent. Similarly, current metrics cannot precisely measure the quality of scene structures. Although several new metrics have been introduced in [42], [51], [56], [60] for evaluating depth maps, they have not gained broad popularity. Thus, designing more effective loss functions and convincing evaluation metrics is also a potential future research direction.

### 8.6 Domain Adaptation

Current benchmark datasets face the challenge of the lack of reliable depth points. Moreover, the data is captured under ideal lighting conditions in limited scenarios. Thus, models trained using this type of data have no guarantee of generalization in different working conditions and domains. Accordingly, it is reasonable to manipulate deep networks in simulated environments. Thereby, we can have not only per-pixel ground truth but also changeable lighting or weather conditions with a great number of different scenarios. Moreover, it encourages the development of more advanced methods that are difficult to be implemented in the real world. The challenge is then how to transfer the model from simulated environments to real-world scenarios. A few works explored domain adaptation methods for depth completion [1], [71]. However, this under-explored problem remains unknown and is worthy of further exploration.

### 8.7 Transformer-based Network Structures

Recently, visual transformers (ViT) have attracted extensive attention and continuously introduced new state-of-the-art results for many perception tasks, including classification [18], semantic segmentation [100], object detection [136] and monocular depth estimation [4]. Unlike CNNs, ViT receives a set of image patches as input and uses self-attention for local and global feature interactions. It may bring a new paradigm shift for depth completion where more effective multi-modality data fusion and novel strategies for handling input sparsity may exist.

### 8.8 Visualization and Interpretability

A few works have attempted to understand and visualize the mechanism of CNNs for monocular depth estimation. It is shown in [16], [41], [43] that CNNs tend to use some monocular cues from RGB images for inferring depths. In addition, as observed in [129] that the features generated inside CNNs are highly disentangled and activated to different depth ranges. An intriguing question is what will be different if we estimate depths when a few sparse depth points are available in inputs. Exploring and answering the above question is essential to the interpretation of learning based approaches, and has promising applications for improvement of their generalization ability, e.g., facilitating domain adaptation; and robustness of deep learning based depth completion methods.

### 8.9 Robustness to Different Sensors

Existing methods are only applicable to particular sensors. For instance, the most frequently used KITTI dataset is captured by a 64-line LiDAR. There is no guarantee that previous methods can be applied to lower scanline sensors, such as 32-line, 16-line LiDARs, and 1-line LiDARs. As demonstrated by [72], [76], [92], [128], the performance degradation is significant from a 64-line sensor to lower scanline sensors. Hence, maintaining the same level of accuracy for lower scanline sensors is challenging. This under-explored problem is also practical in real-world applications since higher scanline sensors are more expensive than lower ones. Therefore, ensuring the accuracy of learning based methods for various lower scanline sensors is also an important and valuable research topic.

## 9 Conclusion

In this article, we present a comprehensive survey of deep learning based depth completion methods. Our review covers traditional and state-of-the-art network structures, loss functions, learning strategies, benchmark datasets, and evaluation metrics. To depict the evolution process and draw the connections between existing works, we provide a fine-grained taxonomy that categorizes existing methods by jointly considering network structures and main technical contributions. Moreover, we visualize the main characteristics of existing methods as well as their quantitative performance on the most popular benchmark datasets to provide an intuitive and straightforward comparison. We then perform in-depth analyses that summarize their performances, similarities, and differences. Finally, we provide open challenges and promising future research directions. Through the above efforts, we hope our work can help readers navigate this field.

**Acknowledgments:** This work was partly supported by the National Natural Science Foundation of China (62073274, 62106156), Shenzhen Science and Technology Program (JCYJ20220818103000001), and the funding AC01202101103 from the Shenzhen Institute of Artificial Intelligence and Robotics for Society.

## References

[1] A. Atapour-Abarghouei and T. P. Breckon, "To complete or to estimate, that is the question: A multi-task approach to depth completion and monocular depth estimation," in *3DV*, 2019, pp. 183–193.

[2] L. Bai, Y. Zhao, M. Elhousni, and X. Huang, "Depthnet: Real-time lidar point cloud depth completion for autonomous vehicles," *IEEE Access*, vol. 8, pp. 227 825–227 833, 2020.

[3] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *TOMS*, vol. 22, no. 4, pp. 469–483, 1996.

[4] S. Bhat, I. Alhashim, and P. Wonka, "Adabins: Depth estimation using adaptive bins," in *CVPR*, 2021, pp. 4008–4017.

[5] Å. Björck, *Numerical methods for least squares problems*. SIAM, 1996.

[6] K. Q. Brown, "Voronoi diagrams from convex hulls," *Information processing letters*, vol. 9, no. 5, pp. 223–228, 1979.

[7] Y. Cao, Z. Wu, and C. Shen, "Estimating depth from monocular images as classification using deep fully convolutional residual networks," *TCVST*, vol. 28, no. 11, pp. 3174–3182, 2017.

[8] Y. Chen, X. Dai, M. Liu, D. Chen, L. Yuan, and Z. Liu, "Dynamic convolution: Attention over convolution kernels," in *CVPR*, 2020, pp. 11 030–11 039.

[9] Y. Chen, B. Yang, M. Liang, and R. Urtasun, "Learning joint 2d-3d representations for depth completion," in *ICCV*, 2019, pp. 10 023–10 032.

[10] Z. Chen, V. Badrinarayanan, G. Drozdov, and A. Rabinovich, "Estimating depth from rgb and sparse sensing," in *ECCV*, 2018, pp. 167–182.

[11] Z. Chen, H. Wang, L. Wu, Y. Zhou, and D. Wu, "Spatiotemporal guided self-supervised depth completion from lidar and monocular camera," in *VCIP*, 2020, pp. 54–57.

[12] X. Cheng, P. Wang, C. Guan, and R. Yang, "Cspn++: Learning context and resource aware convolutional spatial propagation networks for depth completion," in *AAAI*, vol. 34, no. 07, 2020, pp. 10 615–10 622.

[13] X. Cheng, P. Wang, and R. Yang, "Depth estimation via affinity learned with convolutional spatial propagation network," in *ECCV*, 2018, pp. 103–119.

[14] N. Chodosh, C. Wang, and S. Lucey, "Deep convolutional compressed sensing for lidar depth completion," in *ACCV*, 2018, pp. 499–513.

[15] J. Choi, D. Jung, Y. Lee, D. Kim, D. Manocha, and D. Lee, "Selfdeco: Self-supervised monocular depth completion in challenging indoor environments," in *ICRA*, 2021, pp. 467–474.

[16] T. v. Dijk and G. d. Croon, "How do neural networks see depth in single images?" in *ICCV*, 2019, pp. 2183–2191.

[17] M. Dimitrievski, P. Veelaert, and W. Philips, "Learning morphological operators for depth completion," in *ACIVS*, 2018, pp. 450–461.

[18] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *ICLR*, 2021.

[19] R. Du, E. Turner, M. Dzitsiuk, L. Prasso, I. Duarte, J. Dourgarian, J. Afonso, J. Pascoal, J. Gladstone, N. Cruces, S. Izadi, A. Kowdle, K. Tsotsos, and D. Kim, "DepthLab: Real-Time 3D Interaction With Depth Maps for Mobile Augmented Reality," in *UIST*, 2020, pp. 829–843.

[20] W. Du, H. Chen, H. Yang, and Y. Zhang, "Depth completion using geometry-aware embedding," *arXiv preprint arXiv:2203.10912*, 2022.

[21] A. Eldesokey, M. Felsberg, K. Holmquist, and M. Persson, "Uncertainty-aware cnns for depth completion: Uncertainty from beginning to end," in *CVPR*, 2020, pp. 12 014–12 023.

[22] A. Eldesokey, M. Felsberg, and F. S. Khan, "Propagating confidences through cnns for sparse data regression," in *BMVC*, 2018, p. 14.

[23] S. Farkhani, M. F. Kragh, P. H. Christiansen, R. N. Jørgensen, and H. Karstoft, "Sparse-to-dense depth completion in precision farming," in *ICVISP*, 2019, pp. 1–5.

[24] Z. Feng, L. Jing, P. Yin, Y. Tian, and B. Li, "Advancing self-supervised monocular depth learning with sparse lidar," in *CORL*, 2022, pp. 685–694.

[25] C. Fu, C. Dong, C. Mertz, and J. M. Dolan, "Depth completion via inductive fusion of planar lidar and monocular camera," in *IROS*, 2020, pp. 10 843–10 848.

[26] C. Fu, C. Mertz, and J. M. Dolan, "Lidar and monocular camera fusion: On-road depth completion for autonomous driving," in *ITSC*, 2019, pp. 273–278.

[27] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *CVPR*, 2018, pp. 2002–2011.

[28] J. Fu, S. Wang, Y. Lu, S. Li, and W. Zeng, "Kinect-like depth denoising," in *ISCAS*, 2012, pp. 512–515.

[29] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "Virtual worlds as proxy for multi-object tracking analysis," in *CVPR*, 2016, pp. 4340–4349.

[30] M. Garnelo, D. Rosenbaum, C. J. Maddison, T. Ramalho, D. Saxton, M. Shanahan, Y. W. Teh, D. J. Rezende, and S. M. A. Eslami, "Conditional neural processes," in *ICML*, 2018, pp. 1704–1713.

[31] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *AISTATS*, 2011, pp. 315–323.

[32] C. Godard, O. M. Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *CVPR*, 2017, pp. 6602–6611.

[33] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial nets," in *NeurIPS*, 2014.

[34] J. Gu, Z. Xiang, Y. Ye, and L. Wang, "Denselidar: A real-time pseudo dense depth guided depth completion network," *RAL*, vol. 6, no. 2, pp. 1808–1815, 2021.

[35] X. Guo, J. Hu, J. Chen, F. Deng, and T. L. Lam, "Semantic histogram based graph matching for real-time multi-robot global localization in large scale environment," *RAL*, vol. 6, no. 4, pp. 8349–8356, 2021.

[36] P. Hambarde and S. Murala, "S2dnet: Depth estimation from single image and sparse samples," *IEEE Transactions on Computational Imaging*, vol. 6, pp. 806–817, 2020.

[37] K. He, J. Sun, and X. Tang, "Guided image filtering," *TPAMI*, vol. 35, no. 6, pp. 1397–1409, 2013.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[39] G. Hegde, T. Pharale, S. Jahagirdar, V. Nargund, R. A. Tabib, U. Mudenagudi, B. Vandrotti, and A. Dhiman, "Deepdnet: Deep dense network for depth completion task," in *CVPRW*, 2021, pp. 2190–2199.

[40] J. Hu, C. Fan, H. Jiang, X. Guo, Y. Gao, X. Lu, and T. L. Lam, "Boosting light-weight depth estimation via knowledge distillation," *arXiv preprint arXiv:2105.06143*, 2021.

[41] J. Hu and T. Okatani, "Analysis of deep networks for monocular depth estimation through adversarial attacks with proposal of a defense method," *arXiv preprint arXiv:1911.08790*, 2019.

[42] J. Hu, M. Ozay, Y. Zhang, and T. Okatani, "Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries," in *WACV*, 2019, pp. 1043–1051.

[43] J. Hu, Y. Zhang, and T. Okatani, "Visualization of convolutional neural networks for monocular depth estimation," in *ICCV*, 2019, pp. 3869–3878.

[44] M. Hu, S. Wang, B. Li, S. Ning, L. Fan, and X. Gong, "Penet: Towards precise and efficient image guided depth completion," in *ICRA*, 2021, pp. 13 656–13 662.

[45] J. Hua and X. Gong, "A normalized convolutional neural network for guided sparse depth upsampling," in *IJCAI*, 2018, pp. 2283–2290.

[46] J. Huang, A. B. Lee, and D. Mumford, "Statistics of range images," in *CVPR*, vol. 1, 2000, pp. 324–331.

[47] Y.-K. Huang, T.-H. Wu, Y.-C. Liu, and W. H. Hsu, "Indoor depth completion with boundary consistency and self-attention," in *ICCVW*, 2019, pp. 0–0.

[48] Z. Huang, J. Fan, S. Cheng, S. Yi, X. Wang, and H. Li, "Hms-net: Hierarchical multi-scale sparsity-invariant network for sparse depth completion," *TIP*, vol. 29, pp. 3429–3441, 2019.

[49] P. J. Huber, "Robust estimation of a location parameter," in *Breakthroughs in statistics*, 1992, pp. 492–518.

[50] L. Huynh, P. Nguyen, J. Matas, E. Rahtu, and J. Heikkilä, "Lightweight monocular depth with a novel neural architecture search method," in *WACV*, 2022, pp. 3643–3653.

[51] S. Imran, Y. Long, X. Liu, and D. Morris, "Depth coefficients for depth completion," in *CVPR*, 2019, pp. 12 438–12 447.

[52] S. M. Imran, X. Liu, and D. D. Morris, "Depth completion with twin surface extrapolation at occlusion boundaries," in *CVPR*, 2021, pp. 2583–2592.

[53] S. Ito, N. Kaneko, and K. Sumi, "Seeing farther than supervision: Self-supervised depth completion in challenging environments," in *MVA*, 2021, pp. 1–5.

[54] M. Jaritz, R. D. Charette, E. Wirbel, X. Perrotton, and F. Nashashibi, "Sparse and dense data with cnns: Depth completion and semantic segmentation," in *3DV*, 2018, pp. 52–60.

[55] Y. Jeon, H. Kim, and S.-W. Seo, "Abcd: Attentive bilateral convolutional network for robust depth completion," *RAL*, vol. 7, no. 1, pp. 81–87, 2021.

[56] H. Jiang, L. Ding, J. Hu, and R. Huang, "Plnet: Plane and line priors for unsupervised indoor depth estimation," in *3DV*, 2021, pp. 741–750.

[57] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" *NeurIPS*, vol. 30, 2017.

[58] M. F. F. Khan, N. D. Troncoso Aldas, A. Kumar, S. Advani, and V. Narayanan, "Sparse to dense depth completion using a generative adversarial network with intelligent sampling strategies," in *ACM MM*, 2021, pp. 5528–5536.

[59] H. Knutsson and C.-F. Westin, "Normalized and differential convolution," in *CVPR*, 1993, pp. 515–523.

[60] T. Koch, L. Liebel, M. Körner, and F. Fraundorfer, "Comparison of monocular depth estimation methods using geometrically relevant metrics on the ibims-1 dataset," *CVIU*, vol. 191, p. 102877, 2020.

[61] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *3DV*, 2016, pp. 239–248.

[62] S. Lee, J. Lee, D. Kim, and J. Kim, "Deep architecture with cross guidance between single image and sparse lidar data for depth completion," *IEEE Access*, vol. 8, pp. 79 801–79 810, 2020.

[63] A. Li, Z. Yuan, Y. Ling, W. Chi, C. Zhang *et al.*, "A multi-scale guided cascade hourglass network for depth completion," in *WACV*, 2020, pp. 32–40.

[64] Y. Liao, L. Huang, Y. Wang, S. Kodagoda, Y. Yu, and Y. Liu, "Parse geometry from a line: Monocular depth estimation with partial laser observation," in *ICRA*, 2017, pp. 5059–5066.

[65] Y. Q. Lin, T. Cheng, Q. Zhong, W. Zhou, and H. Yang, "Dynamic spatial propagation network for depth completion," in *AAAI*, 2022.

[66] J. Liu, X. Gong, and J. Liu, "Guided inpainting and filtering for kinect depth maps," in *ICPR*, 2012, pp. 2055–2058.

[67] L. Liu, Y. Liao, Y. Wang, A. Geiger, and Y. Liu, "Learning steering kernels for guided depth completion," *TIP*, vol. 30, pp. 2850–2861, 2021.

[68] L. Liu, X. Song, X. Lyu, J. Diao, M. Wang, Y. Liu, and L. Zhang, "Fcfr-net: Feature fusion based coarse-to-fine residual learning for depth completion," in *AAAI*, vol. 35, no. 3, 2021, pp. 2136–2144.

[69] S. Liu, S. De Mello, J. Gu, G. Zhong, M.-H. Yang, and J. Kautz, "Learning affinity via spatial propagation networks," in *NeurIPS*, 2017, pp. 1520–1530.

[70] Y. Long, H. Yu, and B. Liu, "Depth completion towards different sensor configurations via relative depth map estimation and scale recovery," *Journal of Visual Communication and Image Representation*, vol. 80, p. 103272, 2021.

[71] A. Lopez-Rodriguez, B. Busam, and K. Mikolajczyk, "Project to adapt: Domain adaptation for depth completion from noisy and sparse sensor data," in *ACCV*, 2020.

[72] H. Lu, S. Xu, and S. Cao, "Sgtbn: Generating dense depth maps from single-line lidar," *IEEE Sensors Journal*, vol. 21, no. 17, pp. 19 091–19 100, 2021.

[73] K. Lu, N. Barnes, S. Anwar, and L. Zheng, "From depth what can you see? depth completion via auxiliary image reconstruction," in *CVPR*, 2020, pp. 11 303–11 312.

[74] S. Lu, X. Ren, and F. Liu, "Depth enhancement via low-rank matrix completion," in *CVPR*, 2014, pp. 3390–3397.

[75] F. Ma, L. Carlone, U. Ayaz, and S. Karaman, "Sparse depth sensing for resource-constrained robots," *IJRR*, vol. 38, no. 8, pp. 935–980, 2019.

[76] F. Ma, G. V. Cavalheiro, and S. Karaman, "Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera," in *ICRA*, 2019, pp. 3288–3295.

[77] F. Ma and S. Karaman, "Sparse-to-dense: Depth prediction from sparse depth samples and a single image," in *ICRA*, 2018, pp. 4796–4803.

[78] J. Masci, J. Angulo, and J. Schmidhuber, "A learning framework for morphological operators using counter–harmonic mean," in *ISMM*, 2013, pp. 329–340.

[79] J. McCormac, A. Handa, S. Leutenegger, and A. J. Davison, "Scenenet rgb-d: 5m photorealistic images of synthetic indoor trajectories with ground truth," *arXiv preprint arXiv:1612.05079*, 2016.

[80] D. Miao, J. Fu, Y. Lu, S. Li, and C. W. Chen, "Texture-assisted kinect depth inpainting," in *ISCAS*. IEEE, 2012, pp. 604–607.

[81] C. Murdock, M. Chang, and S. Lucey, "Deep component analysis via alternating direction neural networks," in *ECCV*, 2018, pp. 820–836.

[82] E. Nadaraya, "On estimating regression," *Theory of Probability and Its Applications*, vol. 9, pp. 141–142, 1964.

[83] H. Niederreiter, "Random number generation and quasi-monte carlo methods," *SIAM*, vol. 35, no. 4, pp. 680–681, 1993.

[84] A. B. Owen, "A robust hybrid of lasso and ridge regression," *Contemporary Mathematics*, vol. 443, no. 7, pp. 59–72, 2007.

[85] J. Park, K. Joo, Z. Hu, C.-K. Liu, and I. So Kweon, "Non-local spatial propagation network for depth completion," in *ECCV*, 2020, pp. 120–136.

[86] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *CVPR*, 2019, pp. 2337–2346.

[87] 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 *CVPR*, 2019, pp. 3308–3317.

[88] C. Qu, T. Nguyen, and C. Taylor, "Depth completion via deep basis fitting," in *WACV*, 2020, pp. 71–80.

[89] L. Reichardt, P. Mangat, and O. Wasenmüller, "Dvmn: Dense validity mask network for depth completion," in *ITSC*, 2021, pp. 2653–2659.

[90] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*, 2015, pp. 234–241.

[91] 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 *CVPR*, 2016, pp. 3234–3243.

[92] K. Ryu, K.-i. Lee, J. Cho, and K.-J. Yoon, "Scanline resolution-invariant depth completion using a single image and sparse lidar point cloud," *RAL*, vol. 6, no. 4, pp. 6961–6968, 2021.

[93] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *CVPR*, 2018, pp. 4510–4520.

[94] R. Schuster, O. Wasenmuller, C. Unger, and D. Stricker, "Ssgp: Sparse spatial guided propagation for robust and generic interpolation," in *WACV*, 2021, pp. 197–206.

[95] D. Senushkin, M. Romanov, I. Belikov, N. Patakin, and A. Konushin, "Decoder modulation for indoor depth completion," in *IROS*, 2021, pp. 2181–2188.

[96] J. Shen and S.-C. S. Cheung, "Layer depth denoising and completion for structured-light rgb-d cameras," in *CVPR*, 2013, pp. 1187–1194.

[97] S. S. Shivakumar, T. Nguyen, I. D. Miller, S. W. Chen, V. Kumar, and C. J. Taylor, "Dfusenet: Deep fusion of rgb and sparse depth information for image guided dense depth completion," in *ITSC*, 2019, pp. 13–20.

[98] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *ECCV*, vol. 7576, 2012, pp. 746–760.

[99] Z. Song, J. Lu, Y. Yao, and J. Zhang, "Self-supervised depth completion from direct visual-lidar odometry in autonomous driving," *T-ITS*, 2021.

[100] R. A. M. Strudel, R. G. Pinel, I. Laptev, and C. Schmid, "Segmenter: Transformer for semantic segmentation," in *ICCV*, 2021, pp. 7242–7252.

[101] H. Su, V. Jampani, D. Sun, S. Maji, E. Kalogerakis, M.-H. Yang, and J. Kautz, "Splatnet: Sparse lattice networks for point cloud processing," in *CVPR*, 2018, pp. 2530–2539.

[102] J. Tang, F.-P. Tian, W. Feng, J. Li, and P. Tan, "Learning guided convolutional network for depth completion," *TIP*, vol. 30, pp. 1116–1129, 2021.

[103] Z. Tao, P. Shuguo, Z. Hui, and S. Yingchun, "Dilated u-block for lightweight indoor depth completion with sobel edge," *IEEE Signal Processing Letters*, vol. 28, pp. 1615–1619, 2021.

[104] L. Teixeira, M. R. Oswald, M. Pollefeys, and M. Chli, "Aerial single-view depth completion with image-guided uncertainty estimation," *RAL*, vol. 5, no. 2, pp. 1055–1062, 2020.

[105] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *ICCV*, 1998, pp. 839–846.

[106] Y. Tsuji, H. Chishiro, and S. Kato, "Non-guided depth completion with adversarial networks," in *ITSC*, 2018, pp. 1109–1114.

[107] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, "Sparsity invariant cnns," in *3DV*, 2017, pp. 11–20.

[108] W. Van Gansbeke, D. Neven, B. De Brabandere, and L. Van Gool, "Sparse and noisy lidar completion with rgb guidance and uncertainty," in *MVA*, 2019, pp. 1–6.

[109] X. Cheng, P. Wang, and R. Yang, "Learning depth with convolutional spatial propagation network," *TPAMI*, vol. 42, pp. 2361–2379, 2020.

[110] A. Eldesokey, M. Felsberg, and F. S. Khan, "Confidence propagation through cnns for guided sparse depth regression," *TPAMI*, vol. 42, no. 10, pp. 2423–2436, 2019.

[111] K. Lu, N. Barnes, S. Anwar, and L. Zheng, "Depth completion auto-encoder," in *WACVW*, 2022, pp. 63–73.

[112] S. Wang, S. Suo, W.-C. Ma, A. Pokrovsky, and R. Urtasun, "Deep parametric continuous convolutional neural networks," in *CVPR*, 2018, pp. 2589–2597.

[113] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *TOG*, vol. 38, pp. 1 – 12, 2019.

[114] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *TIP*, vol. 13, no. 4, pp. 600–612, 2004.
[115] D. Wofk, F. Ma, T.-J. Yang, S. Karaman, and V. Sze, "Fastdepth: Fast monocular depth estimation on embedded systems," in *ICRA*, 2019, pp. 6101–6108.
[116] A. Wong, S. Cicek, and S. Soatto, "Learning topology from synthetic data for unsupervised depth completion," *RAL*, vol. 6, no. 2, pp. 1495–1502, 2021.
[117] A. Wong, X. Fei, B.-W. Hong, and S. Soatto, "An adaptive framework for learning unsupervised depth completion," *RAL*, vol. 6, no. 2, pp. 3120–3127, 2021.
[118] A. Wong, X. Fei, S. Tsuei, and S. Soatto, "Unsupervised depth completion from visual inertial odometry," *RAL*, vol. 5, no. 2, pp. 1899–1906, 2020.
[119] A. Wong and S. Soatto, "Unsupervised depth completion with calibrated backprojection layers," in *ICCV*, 2021, pp. 12 747–12 756.
[120] X. Xiong, H. Xiong, K. Xian, C. Zhao, Z. Cao, and X. Li, "Sparse-to-dense depth completion revisited: Sampling strategy and graph construction," in *ECCV*, 2020.
[121] Y. Xu, X. Zhu, J. Shi, G. Zhang, H. Bao, and H. Li, "Depth completion from sparse lidar data with depth-normal constraints," in *ICCV*, 2019, pp. 2811–2820.
[122] Z. Xu, H. Yin, and J. Yao, "Deformable spatial propagation networks for depth completion," in *ICIP*, 2020, pp. 913–917.
[123] L. Yan, K. Liu, and E. Belyaev, "Revisiting sparsity invariant convolution: A network for image guided depth completion," *IEEE Access*, vol. 8, pp. 126 323–126 332, 2020.
[124] L. Yan, K. Liu, and L. Gao, "Dan-conv: Depth aware non-local convolution for lidar depth completion," *Electronics Letters*, vol. 57, no. 20, pp. 754–757, 2021.
[125] Z. Yan, K. Wang, X. Li, Z. Zhang, B. Xu, J. Li, and J. Yang, "Rignet: Repetitive image guided network for depth completion," *arXiv preprint arXiv:2107.13802*, 2021.
[126] X. Yang, W. Liu, D. Tao, and J. Cheng, "Canonical correlation analysis networks for two-view image recognition," *Information Sciences*, vol. 385, pp. 338–352, 2017.
[127] Y. Yang, A. Wong, and S. Soatto, "Dense depth posterior (ddp) from single image and sparse range," in *CVPR*, 2019, pp. 3353–3362.
[128] S. Yoon and A. Kim, "Balanced depth completion between dense depth inference and sparse range measurements via kiss-gp," *IROS*, pp. 10 468–10 475, 2020.
[129] Z. You, Y.-H. Tsai, W.-C. Chiu, and G. Li, "Towards interpretable deep networks for monocular depth estimation," in *ICCV*, 2021, pp. 12 879–12 888.
[130] Q. Yu, L. Chu, Q. Wu, and L. Pei, "Grayscale and normal guided depth completion with a low-cost lidar," in *ICIP*, 2021, pp. 979–983.
[131] C. Zhang, Y. Tang, C. Zhao, Q. Sun, Z. Ye, and J. Kurths, "Multitask gans for semantic segmentation and depth completion with cycle consistency," *TNNLS*, vol. 32, no. 12, pp. 5404–5415, 2021.
[132] Y. Zhang, T. Nguyen, I. D. Miller, S. S. Shivakumar, S. Chen, C. J. Taylor, and V. Kumar, "Dfinenet: Ego-motion estimation and depth refinement from sparse, noisy depth input with rgb guidance," *arXiv preprint arXiv:1903.06397*, 2019.
[133] Y. Zhang and T. Funkhouser, "Deep depth completion of a single rgb-d image," in *CVPR*, 2018, pp. 175–185.
[134] Y. Zhang, P. Wei, H. Li, and N. Zheng, "Multiscale adaptation fusion networks for depth completion," in *IJCNN*, 2020, pp. 1–7.
[135] Y. Zhang, P. Wei, and N. Zheng, "A multi-cue guidance network for depth completion," *Neurocomputing*, vol. 441, pp. 291–299, 2021.
[136] Z. Zhang, X. Lu, G. Cao, Y. Yang, L. Jiao, and F. Liu, "Vit-yolo: Transformer-based yolo for object detection," in *ICCV*, 2021, pp. 2799–2808.
[137] S. Zhao, M. Gong, H. Fu, and D. Tao, "Adaptive context-aware multi-modal network for depth completion," *TIP*, vol. 30, pp. 5264–5276, 2021.
[138] Y. Zhong, C.-Y. Wu, S. You, and U. Neumann, "Deep rgb-d canonical correlation analysis for sparse depth completion," in *NeurIPS*, 2019, pp. 5332–5342.
[139] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable convnets v2: More deformable, better results," in *CVPR*, 2019, pp. 9308–9316.
[140] Y. Zhu, W. Dong, L. Li, J. Wu, X. Li, and G. Shi, "Robust depth completion with uncertainty-driven loss functions," in *AAAI*, 2022.
[141] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *CVPR*, 2018, pp. 8697–8710.

**Junjie Hu** (Member, IEEE) received the M.S. and Ph.D. degrees from the Graduate School of Information Science, Tohoku University, Sendai, Japan, in 2017 and 2020, respectively. He is currently a Research Scientist with the Shenzhen Institute of Artificial Intelligence and Robotics for Society. His research interests include machine learning, computer vision, and robotics.

**Chenyu Bao** received the B.Eng degree from the School of Mechanical Science and Engineering, HuaZhong University of Science and Technology in 2022. He is currently a graduate student in the School of Science and Engineering, Chinese University of Hong Kong, Shenzhen. His research interests include computer vision and robot perception.

**Mete Ozay** (M'09) received the B.Sc., M.Sc., Ph.D. degrees in mathematical physics, information systems, and computer engineering & science from METU, Turkey. He has been a visiting Ph.D. and fellow in the Princeton University, USA, a research fellow in the University of Birmingham, UK, and an Assistant Professor in the Tohoku University, Japan. His current research interests include pure and applied mathematics, theoretical computer science & neuroscience.

**Chenyou Fan** received the B.S. degree in computer science from the Nanjing University, China, in 2011, and the M.S. and Ph.D. degrees from Indiana University, USA, in 2014 and 2019, respectively. He is an Associate Professor at the School of Artificial Intelligence, South China Normal University, China. His research interests include machine learning and computer vision.

**Qing Gao** received his Ph.D. degree from the Chinese Academy of Sciences (CAS), Shenyang, China, in 2020. He is currently a Research Scientist with the Shenzhen Institute of Artificial Intelligence and Robotics for Society. His research interests include robotics, artificial intelligence, machine vision, and human–robot interaction.

**Honghai Liu** (Fellow, IEEE) received the Ph.D. degree in intelligent robotics from King's College London, London, U.K., in 2003. He is a Professor at the Harbin Institute of Technology (Shenzhen), Shenzhen, China. He is also a Chair Professor of Human–Machine Systems with the University of Portsmouth, Portsmouth, U.K. His research interests include multi-sensory data fusion, pattern recognition, intelligent video analytics, intelligent robotics, and their practical applications.

**Tin Lun Lam** (Senior Member, IEEE) received the Ph.D. degrees from the Chinese University of Hong Kong, Hong Kong, in 2010. He is an Assistant Professor with the Chinese University of Hong Kong, Shenzhen, China, and the Director of the Center for Intelligent Robots, Shenzhen Institute of Artificial Intelligence and Robotics for Society. His research interests include multi-robot systems, field robotics, and collaborative robotics.