From Depth What Can You See? Depth Completion via Auxiliary Image Reconstruction

Kaiyue Lu*[1,2], Nick Barnes[1], Saeed Anwar[2,1], and Liang Zheng[1]

[1]*The Australian National University* [2]*Data61, CSIRO*

Abstract

Depth completion recovers dense depth from sparse measurements, e.g., LiDAR. Existing depth-only methods use sparse depth as the only input. However, these methods may fail to recover semantically consistent boundaries, or small/thin objects due to 1) the sparse nature of depth points and 2) the lack of images to provide semantic cues. This paper continues this line of research and aims to overcome the above shortcomings. The unique design of our depth completion model is that it simultaneously outputs a reconstructed image and a dense depth map. Specifically, we formulate image reconstruction from sparse depth as an auxiliary task during training that is supervised by the unlabelled gray-scale images. During testing, our system accepts sparse depth as the only input, i.e., the image is not required. Our design allows the depth completion network to learn complementary image features that help to better understand object structures. The extra supervision incurred by image reconstruction is minimal, because no annotations other than the image are needed. We evaluate our method on the KITTI depth completion benchmark and show that depth completion can be significantly improved via the auxiliary supervision of image reconstruction. Our algorithm consistently outperforms depth-only methods and is also effective for indoor scenes like NYUv2.

1. Introduction

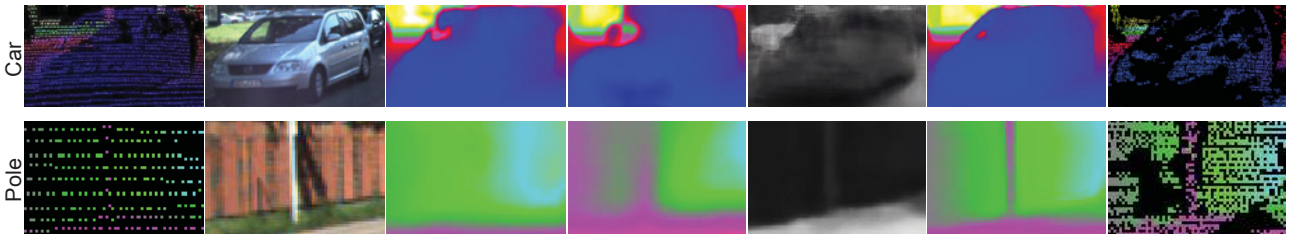
Dense and accurate depth is beneficial to many computer vision tasks, *e.g.*, 3D object detection [5, 39], optical flow estimation [31, 48], and semantic segmentation [42, 45]. However, depth maps acquired from sensors, like LiDAR, are too sparse to fulfill practical needs. Depth completion thus aims to recover dense depth from sparse measurements.

Existing studies for depth completion are generally classified into *depth-only* and *multiple-input* methods. Depth-only methods use sparse depth as the only input [36, 26, 11]. However, they may fail to recover semantically consistent boundaries, or small/thin objects due to the sparsity of input depth points and the lack of images to provide semantic cues (see Fig. 1). The intuitive solution is to take the RGB image or its gray scale as an additional input to the model, as used by multiple-input methods [40, 30, 7]. Nevertheless, aggregating features from two modalities is challenging and complicated [11, 30], and also, calibrating images to depth maps can be expensive in practice [14, 20]. Further, for end-use systems, such as autonomous vehicles, incorporating additional calibrated sensors, like the camera, and associated processing modules may significantly increase the cost.

The question arising from above is, can we continue the depth-only paradigm but incorporate more image features so as to provide richer semantics to overcome shortcomings of this paradigm? To answer this, we start from an observation that, from sparse depth we can still roughly see some object structures according to their general shape and depth difference to the background, *e.g.*, the car and pole examples in Fig. 1. This motivates us into thinking if some image semantics can be recovered from sparse depth, we will be able to relax the need of taking the image as input.

Motivated by the above considerations, we propose a depth completion model that takes sparse depth as the only input and at the same time has the ability to learn from images features to provide semantic cues. Specifically, we train the network to output **a reconstructed image** and **a dense depth map** simultaneously, as illustrated in Fig. 2(a). We formulate image reconstruction from sparse depth as an auxiliary task during training that is **supervised by the unlabelled gray-scale images**. During testing, no image is required as input. The unique design of our model allows the depth completion network to learn complementary image features that help to better understand object structures, and thus, produce more semantically consistent and accurate results than existing depth-only methods (see Fig. 1). Moreover, the extra supervision incurred by image reconstruction at the training stage is minimal, because no annotations other than the image are needed. Therefore, our method is practical in use. We evaluate our method on the KITTI depth completion benchmark and show that depth completion can be significantly improved via the auxiliary learning of image reconstruction.

*Corresponding author: kaiyue.lu@data61.csiro.au



(a) Sparse depth (b) RGB image (c) Glob_guide [38] (d) S2D [26] (e) Our reconst. image (f) Our depth (g) Ground truth

Figure 1. Depth completion from sparse depth. Only given (a) sparse depth as input without (b) corresponding RGB images, existing depth-only methods, like (c) Glob_guide [38] and (d) S2D [26] cannot appropriately complete the depth of objects with specific boundaries (*e.g.*, the car) and small/thin objects (*e.g.*, the pole), due to the lack of depth points and images to provide semantic cues. (e) Different from theirs, we implicitly recover these semantic cues via image reconstruction directly from sparse depth. It helps (f) our depth completion recover more semantically consistent boundaries and deal with small/thin objects more accurately, and our results are closer to (g) the ground truth. All the depth maps are colorized for better visualization. 直接从稀疏深度恢复语义线索，帮助深度补全边界更一致，细节更准确

In summary, we make the following major contributions:

- We propose a depth completion network that only takes sparse depth as input and outputs a reconstructed image and a dense depth map simultaneously. This practice largely overcomes the shortcomings of existing depth-only methods, *i.e.,* the lack of semantic cues.
- By formulating image reconstruction as an auxiliary task during training, we do not need additional annotations other than the image. This is cheap and easy to implement. During testing, no image is required.
- We demonstrate that our approach significantly outperforms depth-only methods on the KITTI depth completion benchmark and can be applied to indoor scenes.

2. Related Work

This section introduces existing literatures on depth completion and multi-task learning, as well as auxiliary learning.

2.1. Depth completion

Existing methods for depth completion can be roughly classified into depth-only and multiple-input. Depth-only methods only take sparse depth as input and output the dense depth map (see Fig. 2(b)). To handle data sparsity, Uhrig *et al*. [36] propose SparseConvs, a sparsity invariant CNN method, where a binary mask is generated to indicate the availability of depth value, *i.e*., 1 for available depth values and 0 for none. The binary mask can be updated iteratively but is over-saturated in shallow layers and has limited performance in deeper layers [17]. It can be improved by designing a more adaptive mask [15] or using other techniques like compressed sensing [8], confidence/attention map construction [12, 38], and multi-scale learning and refinement [27, 26]. However, they are computationally expensive and unable to recover the full structure of objects due to the lack of images to provide semantic cues.

Differently, multiple-input methods supplement extra information into the input and take advantage of complementary features from other modalities. Traditionally, the popular choice is to take the image as an additional input, since it can provide rich semantic cues [30, 40, 11, 26, 7, 16]. This is particularly useful in distinguishing different objects, generating consistent boundaries, and preserving details. In this case fusion strategies are widely employed, *e.g*., early fusion where the image and the depth map are concatenated to get a 4D tensor, and late fusion by extracting features from the image and the depth separately and then fusing them, as shown in Fig. 2(c) and (d) respectively. A number of studies also attempt to make use of other modalities, *e.g*., surface normals [30, 40], semantic classes [18], point clouds [6], and disparity maps [41]. However, these inevitably increase model complexity.

2.2. Multi-task learning

Multi-Task Learning (MTL) aims to improve performance by learning individual yet related tasks simultaneously [2]. Features are shared among these tasks to exploit common representations, while they can also be complementary to each other [19]. This learning strategy has been successfully employed in semantic segmentation [19, 29], object detection [22, 23], single image depth estimation [4, 46]. Similarly, for depth completion, Qiu *et al*. propose to regress completion and surface normal estimation at the same time [30]. Jaritz *et al*. jointly train the network with semantic segmentation and depth completion [18].

Recently, a variant of MTL, known as Auxiliary Learning (AL), is becoming popular. In this framework, a primary task is defined while all other tasks served as auxiliary regularizers that enhance the primary one [32]. AL has been proven to be effective in a number of computer vision tasks, *e.g*., hand-written digit recognition [43], semantic segmentation [24], face anti-spoofing [25], visual odometry [37] *etc*. We also employ it and focus on depth completion as the primary task. We expect the auxiliary task, *i.e*. image reconstruction, to facilitate it with complementary image features that can help to better understand object structures. To the best of our knowledge, our work is the first to introduce auxiliary learning to depth completion.

同时输出深度和图像，图像重构只作为辅助任务

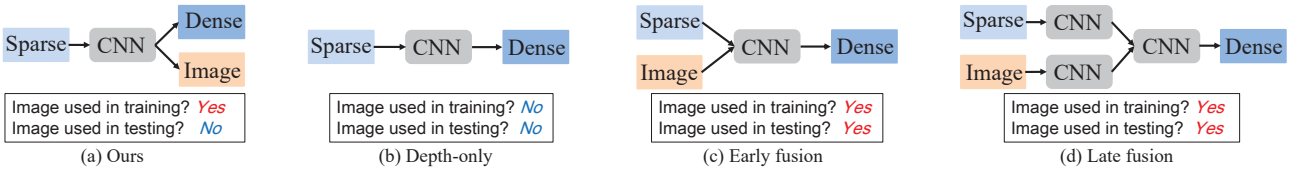


Figure 2. Different depth completion models. (a) Our model takes sparse depth as the only input, and outputs a reconstructed image and dense depth simultaneously. Image reconstruction is only used as an auxiliary task at the training stage. During testing, no image is required. (b) Depth-only models input sparse depth and output its dense map. (c) and (d) Multiple-input models take the image as an additional input with an early or late fusion strategy, and the image is required in both training and testing.

3. Methodology

In this section, we first give a general formulation to describe existing depth completion models and contrast these with ours. We then elaborate the details of our method.

3.1. Depth completion models

Given a sparse depth map $\mathbf{x}$ where the empty locations are filled with zeros, a general depth completion model learns to recover dense depth $\tilde{\mathbf{x}}$ supervised by its ground truth $\mathbf{x}^*$.

Depth-only model. A depth-only model D only takes sparse depth, $\mathbf{x}$, as input:

$$\tilde{\mathbf{x}} = D(\mathbf{x}; \theta_D), \tag{1}$$

where θ_D denotes the model parameters. The optimal model is parameterized by θ_D^*, and obtained during training by minimizing the loss function $\mathcal{L}$, *i.e.*,

$$\theta_D^* = \arg\min_{\theta_D} \mathcal{L}(\tilde{\mathbf{x}}, \mathbf{x}^*). \tag{2}$$

Multiple-input model. A multiple-input model T combines the sparse depth $\mathbf{x}$ and the corresponding calibrated image $\mathbf{r}$ as input:

多输入模型

$$\tilde{\mathbf{x}} = T(\mathbf{x}, \mathbf{r}; \theta_T), \tag{3}$$

and the optimal model is

$$\theta_T^* = \arg\min_{\theta_T} \mathcal{L}(\tilde{\mathbf{x}}, \mathbf{x}^*). \tag{4}$$

Our model. As illustrated in Fig. 3, our model G takes the sparse depth $\mathbf{x}$ as the only input, and outputs dense depth $\tilde{\mathbf{x}}$ and a reconstructed image $\tilde{\mathbf{r}}$ simultaneously:

多输出

$$\tilde{\mathbf{x}}, \tilde{\mathbf{r}} = G(\mathbf{x}; \theta_G) \Rightarrow \begin{cases} \tilde{\mathbf{x}} = G_{dpt}(\mathcal{F}(\mathbf{x}; \theta_{\mathcal{F}}); \theta_{dpt}, \theta_{shr}) \\ \tilde{\mathbf{r}} = G_{img}(\mathcal{F}(\mathbf{x}; \theta_{\mathcal{F}}); \theta_{img}, \theta_{shr}) \end{cases}, \tag{5}$$

where $\mathcal{F}$ parameterized by $\theta_{\mathcal{F}}$ extracts features from the input, θ_{dpt} and θ_{img} are parameters for the depth completion module G_{dpt} and image reconstruction module G_{img} respectively, and θ_{shr} represents feature sharing between the two modules. During training, the parameters of the joint model, $\theta_G = (\theta_{\mathcal{F}}, \theta_{dpt}, \theta_{img}, \theta_{shr})$, are optimized such that

$$\theta_G^* = \arg\min_{\theta_G} (w_{dpt} \cdot \mathcal{L}(\tilde{\mathbf{x}}, \mathbf{x}^*) + w_{img} \cdot \mathcal{L}(\tilde{\mathbf{r}}, \mathbf{r})), \tag{6}$$

where w_{dpt} and w_{img} are weighting factors of the two tasks. This is a typical multi-task learning framework [3], where the network jointly learns to recover dense depth and reconstruct the image *directly* from the sparse input. More specifically, we treat depth completion as the primary task, and image reconstruction as an auxiliary task, which is known as *auxiliary learning* [32]. The purpose is to transfer useful knowledge from the auxiliary task to the primary one to enhance the feature learning of the latter [10]. In our case, by enforcing feature correlations via sharing, we expect the depth completion network to learn more complementary image features to provide semantic cues for understanding object structures. Note that the auxiliary image reconstruction is supervised by unlabelled camera images, which are cheaper to acquire than manually-labelled data. In the following, we illustrate the network architecture, loss functions, and how image reconstruction facilitates depth completion.

During testing, we only focus on the primary depth completion and no image is required, *i.e.*,

$$\tilde{\mathbf{x}} = G_{dpt}(\mathcal{F}(\mathbf{x}; \theta_{\mathcal{F}}^*); \theta_{dpt}^*, \theta_{shr}^*). \tag{7}$$

3.2. Network architecture

The overall network architecture for *training* our model is based on Eq. 5 and shown in Fig. 3. We specify each module below and more details of the configuration of each layer can be found in the supplementary material.

Feature encoder $\mathcal{F}$. We extract multi-scale features from the input by convolving with different kernel sizes. This is inspired by the Inception architecture [35], but with 3×3, 5×5, 7×7, 9×9 kernels instead. In the last layer, all the feature maps are with $1/16$ resolution to the input and concatenated in a channel-wise manner. We denote the output of this encoder, representing initial features from the sparse input, as $f_0 = \mathcal{F}(\mathbf{x})$.

Depth completion module G_{dpt}. It is composed of a depth feature extractor G_{d1} and depth decoder G_{d2}. G_{d1} focuses on learning depth-specific features and gradually upsamples f_0 with transpose convolutions ($1/16 \rightarrow 1/8 \rightarrow 1/4 \rightarrow 1/2$). The intermediate features in G_{d1} are also transferred to the feature sharing module (see Fig. 4). Its output, $G_{d1}(f_0)$, containing both depth and shared features, is fed into G_{d2} to produce dense depth.

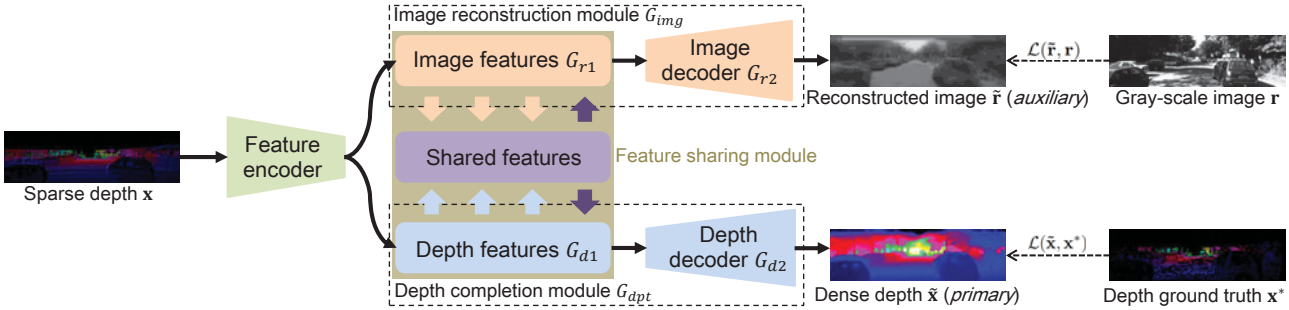


Figure 3. Network architecture for *training* our model. It contains: 1) the feature encoder - extracting initial features from the sparse input; 2) the depth completion module - specializing depth features and producing dense depth; 3) the image reconstruction module - specializing image features and reconstructing the image from sparse depth; and 4) the feature sharing module - aggregating features from depth and image modules. Depth completion is the primary task, while image reconstruction is an auxiliary and supervised by the gray-scale image.

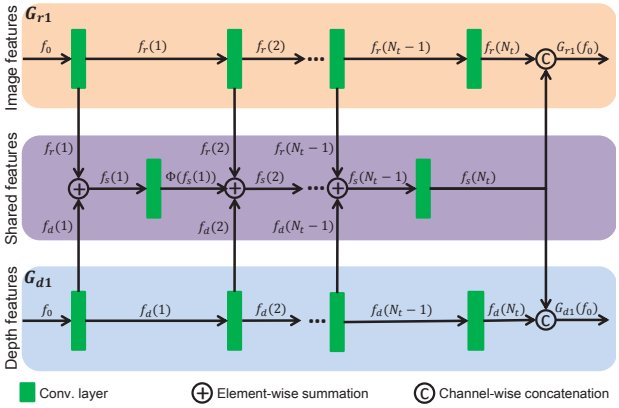


Figure 4. Structure of the feature sharing module. It aggregates depth and image features by element-wise summation, followed by convolutions in each layer. The depth and image feature modules output the concatenation of their last layer features and the shared features.

Image reconstruction module G_{img}**.** The underlying architecture of the image reconstruction module is identical to the depth completion module, where G_{r1} specializes and transfers image features, and the image decoder, G_{r2}, outputs the reconstructed image based on image-specific and shared features.

Feature sharing module. This module aggregates features from the depth and image feature modules via element-wise summation followed by convolutions in each layer, as illustrated in Fig. 4. Suppose there are N_t layers in each module, and we denote the feature maps in n-*th* convolutional layer in G_{d1}, G_{r1}, and the sharing module as $f_d(n)$, $f_r(n)$, and $f_s(n)$ respectively. We use $\Phi(\cdot)$ to represent the general convolutional operator. In the first layer, *i.e.*, $n = 1$,

$$\begin{cases} f_r(1) = \Phi(f_0) \\ f_d(1) = \Phi(f_0) \\ f_s(1) = f_r(1) \oplus f_d(1) \end{cases}, \tag{8}$$

where $\oplus$ is element-wise summation. In subsequent layers before the last layer, *i.e.*, $1 < n < N_t$,

$$\begin{cases} f_r(n) = \Phi(f_r(n-1)) \\ f_d(n) = \Phi(f_d(n-1)) \\ f_s(n) = f_r(n) \oplus f_d(n) \oplus \Phi(f_s(n-1)) \end{cases}. \tag{9}$$

In the last layer where $n = N_t$, only convolutions are performed,

$$\begin{cases} f_r(N_t) = \Phi(f_r(N_t-1)) \\ f_d(N_t) = \Phi(f_d(N_t-1)) \\ f_s(N_t) = \Phi(f_s(N_t-1)) \end{cases}. \tag{10}$$

The final output of both G_{d1} and G_{r1} is the channel-wise concatenation of their corresponding feature maps and the shared features, *i.e.*,

$$\begin{cases} G_{d1}(f_0) = Cat(f_d(N_t), f_s(N_t)) \\ G_{r1}(f_0) = Cat(f_r(N_t), f_s(N_t)) \end{cases}. \tag{11}$$

The two concatenated features are further fed into depth and image decoders to produce the dense depth $\tilde{\mathbf{x}}$ and reconstructed image $\tilde{\mathbf{r}}$, *i.e.*,

$$\begin{cases} \tilde{\mathbf{x}} = G_{dpt}(f_0) = G_{d2}(G_{d1}(\mathcal{F}(\mathbf{x}))) \\ \tilde{\mathbf{r}} = G_{img}(f_0) = G_{r2}(G_{r1}(\mathcal{F}(\mathbf{x}))) \end{cases}. \tag{12}$$

Loss functions. To train the network, we first define the ℓ_2 loss for depth completion (primary task):

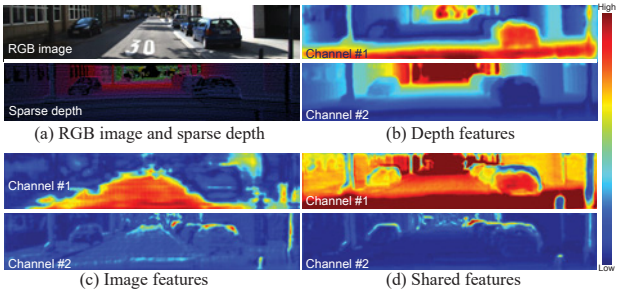
$$\ell_{dpt} = \frac{1}{N_1} \left\| \Psi \odot (\tilde{\mathbf{x}} - \mathbf{x}^*) \right\|_2^2, \tag{13}$$

where N_1 is the number of pixels that have depth values in ground truth $\mathbf{x}^*$, Ψ is a binary mask of $\mathbf{x}^*$ where 1 means available depth values and 0 for none, and $\odot$ is the element-wise multiplier. We use the gray-scale image, $\mathbf{r}$, to supervise auxiliary image reconstruction. The ℓ_2 loss function is:

$$\ell_{img} = \frac{1}{N_2} \left\| \tilde{\mathbf{r}} - \mathbf{r} \right\|_2^2, \tag{14}$$

where N_2 is the number of pixels in the image. Hence, the total loss for the entire network is:

$$\ell_{total} = w_{dpt} \cdot \ell_{dpt} + w_{img} \cdot \ell_{img}. \tag{15}$$



(a) RGB image and sparse depth (b) Depth features

(c) Image features (d) Shared features

Figure 5. Feature visualization. (a) The RGB image is used for reference and sparse depth is the only input. (b) Depth features focus more on objects that are visible in both near and far regions of the depth map. (c) Image features highlight global visual structure as well as some details that are not reflected in depth. (d) Shared features take advantage of both depth and image features, and cover most objects (upper) as well as some details like their boundaries and structures (bottom). Zoom in for clearer visualization.

Eq. 15 indicates that ℓ_{img} serves as a regularizer during training to facilitate parameter learning of depth completion and thus improves its overall performance.

Discussions. To further investigate the learning ability of our network, we select and visualize two representative feature maps from the first and second channels of the last layer of the depth features, image features, and shared features respectively, *i.e.*, $f_d(N_t)$, $f_r(N_t)$, and $f_s(N_t)$. The depth features shown in Fig. 5(b) indicate that they focus more on visible objects in both near and far regions of the depth map, *e.g.*, cars and poles. However, due to the sparsity of depth points and lack of image information, these features only partially reflect the real shape of the objects.

The image features in Fig. 5(c), by contrast, highlight the global structure, *e.g.*, the road, and some details that are not reflected in depth, *e.g.*, the missing parts around car boundaries and poles. These features are beneficial to distinguish highly occluded objects and recover the full structure of small/thin objects. Therefore, image features are complementary to depth features. After aggregating these features via the sharing module, the shared features shown in Fig. 5(d) take advantage of both depth and image features, and cover most objects as well as some details like their boundaries and structures. In summary, the auxiliary learning of image reconstruction enables the depth completion network to learn useful and complementary image features via sharing, and thus obtains more semantic cues for better completion. This can be achieved even without the image as input.

4. Experiment

In the following, we show the effectiveness of our method through extensive experiments. This includes quantitative and visual comparison with state-of-the-art approaches, ablation studies on several factors that affect completion performance, and generalization to indoor scenes.

4.1. Implementation details

Dataset. The KITTI depth completion benchmark [36] contains raw, sparse depth maps collected by LiDAR which are further separated into 85,898 frames for training, 1,000 for validation, and 1,000 for testing. Each depth map has the corresponding RGB image, and we convert the RGB image to gray-scale to supervise image reconstruction only at the training stage. The KITTI ground truth was generated by accumulating multiple LiDAR frames, and removing outliers by semi-global matching [36], which makes it semi-dense (depth completion becomes harder in this case because the semi-dense ground truth cannot completely reflect the depth of some object boundaries and small objects). Test samples have no ground truth available, and the results are evaluated on the benchmark server.

Training configuration. The network is implemented in PyTorch [28]. During training, the input is cropped from the bottom to 352×1216. We train the network on two NVIDIA 1080 Titan GPUs with a batch size of 16. The loss function is defined in Eq. 15, where $w_{dpt} = 1$ and $w_{img} = 10^{-4}$. We use the Adam optimizer [21], and the initial learning rate is 10^{-3} and decayed by half every five epochs.

Evaluation metrics. Similar to the benchmark [36], we calculate RMSE and MAE from depth (mm), and iRMSE and iMAE from inverse depth (1/km). RMSE measures depth completion errors directly and penalizes more for undesirable larger errors. Differently, MAE treats all the errors equally. Hence, we consider RMSE to be the more important evaluation metric, which is consistent with the KITTI benchmark where RMSE is used for ranking.

4.2. Comparison with the state-of-the-art methods

Quantitative comparison. In Table 1, we report quantitative results of our method as well as the state-of-the-art approaches on the KITTI test set. Compared with depth-only methods (highlighted in gray), our model trained with ℓ_2 loss achieves the best **RMSE = 901.43**, ranking first among them and surpassing the second place by 21.50. Our MAE and iMAE are both comparable to others. However, our iRMSE is less competitive. The underlying reason is iRMSE measures the accuracy of inverse depth, in which case depth points in closer regions with relatively smaller errors are more dominant. By contrast, we use ℓ_2 loss for depth to penalize larger errors. There thus exists a trade-off in balancing large and small errors with this metric. We consider that iRMSE is less reliable than RMSE in reflecting model accuracy mainly because iRMSE is not a direct metric to measure depth errors. We have mathematically justified this in the supplementary material that smaller errors may yield larger iRMSE. We refer the reader to Fig. 8(c) where our model performs competitively to the state-of-the-art methods in close regions, *e.g.*, 0-40m. iMAE has the same issue. Consequently, we still consider RMSE as the primary metric.

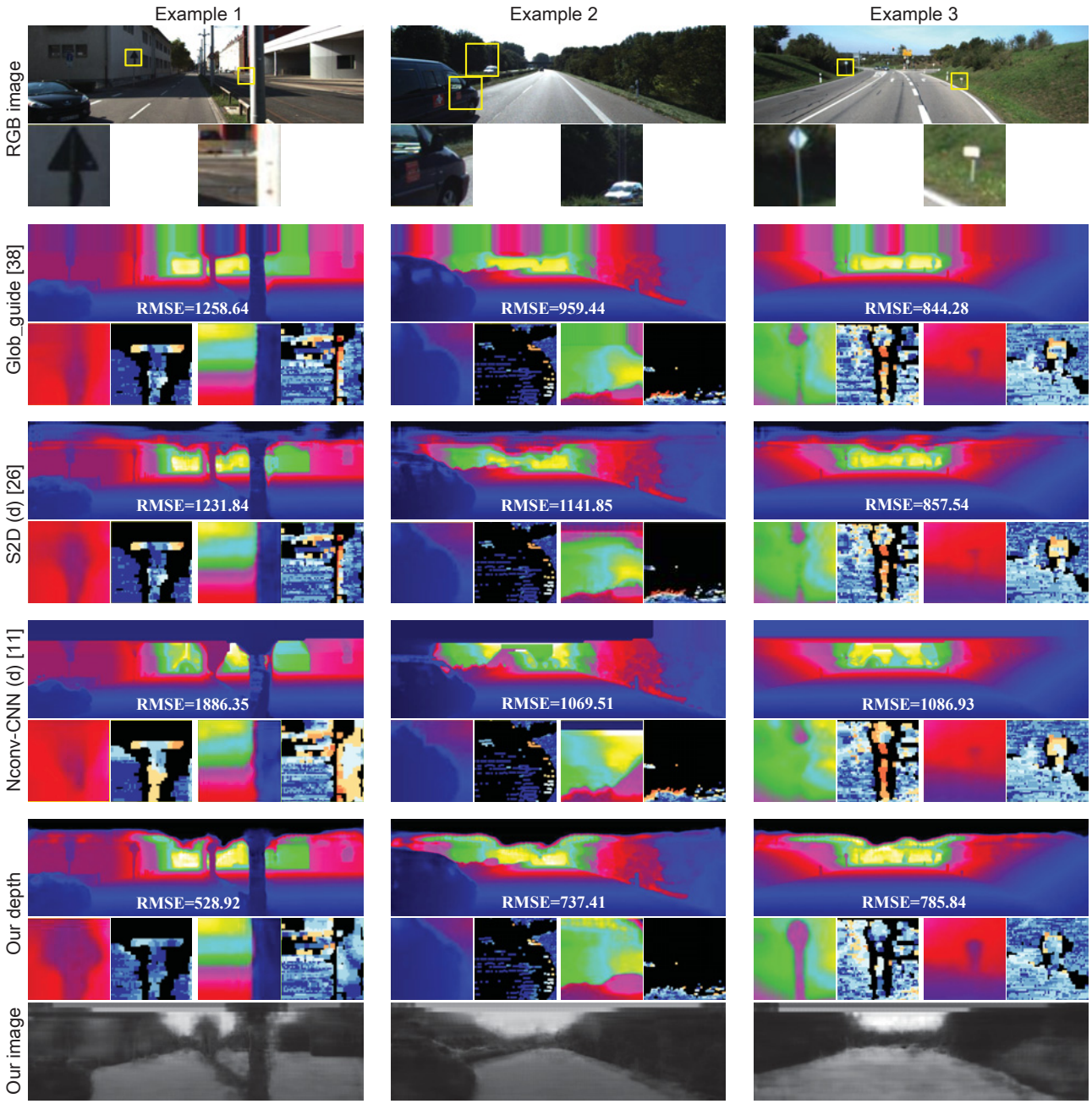
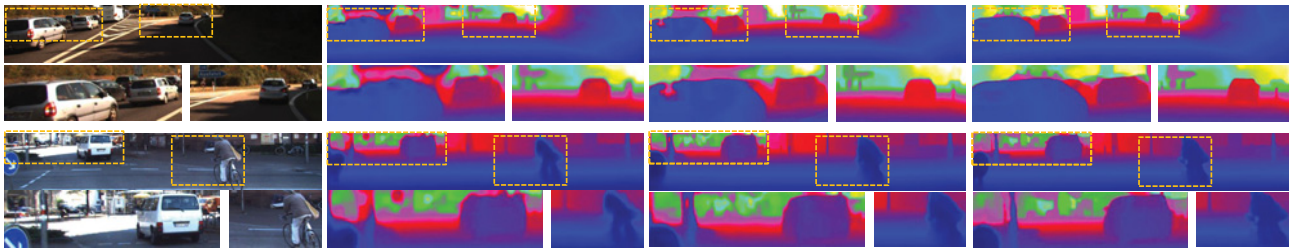


Figure 6. Visual comparison with state-of-the-art depth-only methods on the KITTI test set. Our model can produce more accurate depth completion results in small/thin objects, boundaries, and distant regions. To the right of each close-up is the error map, where small errors are displayed in blue and large errors in red. Black regions mean the ground truth labels are not used for evaluation.

In fact, several studies have observed that training with different loss functions may yield different results [6, 12]. For example, Spade-sD [17] achieves the best iRMSE and iMAE because it is directly trained on inverse depth. To further validate our method, we re-train the model with ℓ_1 loss, with the same network setting in Section 4.1. Unsurprisingly, using ℓ_1 loss yields a smaller MAE (best among depth-only methods) but slightly larger RMSE (still ranks first). Since we mainly focus on RMSE, in the following, our default model refers to that trained with ℓ_2 loss unless otherwise specified.

Our model is also comparable to multiple-input meth-



(a) RGB image (b) Only with depth features (c) With image features w/o sharing (d) With image features and sharing

Figure 7. Visual comparison of depth completion results after incorporating image reconstruction and feature sharing. (a) RGB images for reference. (b) Only with depth features cannot recover the full structure of objects. (c) With image features but without sharing, the results are slightly improved. (d) With shared features, the model performs better in recovering consistent object structures and small/thin objects.

Method	RMSE ↓	MAE ↓	iRMSE ↓	iMAE ↓
SparseConvs [36]	1601.33	481.27	4.94	1.78
ADNN [8]	1325.37	439.48	59.39	3.19
Spade-sD [17]	1035.29	248.32	**2.60**	**0.98**
NConv-CNN (d) [11]	1268.22	360.28	4.67	1.52
S2D (d) [26]	954.36	288.64	3.21	1.35
Glob_guide [38]	922.93	249.11	2.80	1.07
Ours (ℓ_2 loss)	**901.43**	292.36	4.92	1.35
Ours (ℓ_1 loss)	915.86	**231.37**	3.19	1.23
DeepLiDAR [30]	758.38	226.50	2.56	1.15
PwP [40]	777.05	235.17	2.42	1.13
S2D (gd) [26]	814.37	249.95	2.80	1.21
NConv-CNN (gd) [11]	829.98	233.26	2.60	1.03
CSPN [7]	1019.64	279.46	2.93	1.15

Table 1. Quantitative comparison with state-of-the-art methods on the KITTI test set. The best results are marked with **bold** among methods that do not use any images during testing (gray region). ↓ means smaller is better.

ods, *e.g.*, it surpasses CSPN [7] in terms of RMSE, and outperforms PwP [40], S2D (gd) [26], NConv-CNN-L2 (gd) [11], and CSPN [7] in MAE if trained with ℓ_1 loss. In summary, our approach generally lies in between depth-only and multiple-input methods, showing competitive performance even without using the image as input.

Visual comparison. We present qualitative results in Fig. 6 and compare with three state-of-the-art depth-only methods, *i.e.*, Glob_guide [38], S2D (d) [26], and Nconv-CNN (d) [11]. For each example, we also provide the RMSE and close-ups (left) with corresponding error maps (right). Overall, our model is able to produce more accurate depth completion results for small/thin objects, boundaries, and distant regions. Specifically, our method recovers the depth of narrow poles in Example 1 and 3 more appropriately in preserving their general structures. Besides, our completion results also have smaller errors along boundaries of the tree and car, as well as the distant regions, *e.g.*, the right close-up in Example 2 where the white car and its surroundings are relatively far away.

Moreover, our RMSE in these three examples is significantly better than others. The good performance is mainly owing to image reconstruction as an auxiliary task[1], because it enables our depth completion network to acquire more semantic cues and understand object structures better. Besides, since the image is truly dense, it can also overcome the shortcoming of the semi-dense ground truth in reflecting the full structures of objects. Therefore, our performance is largely improved over depth-only methods.

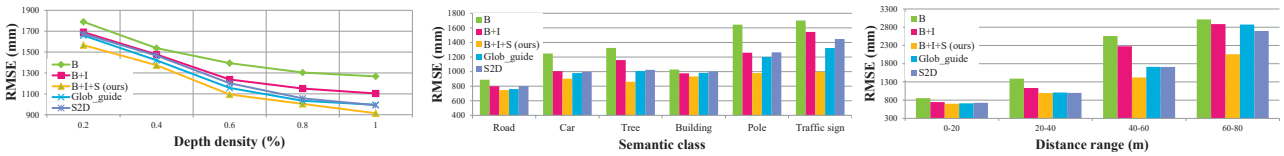
	RMSE ↓	MAE ↓
B	1267.01	322.32
B + I	1103.85	301.64
B + I + S (ours)	**914.65**	**297.38**

Table 2. Ablation study on the KITTI validation set. "B", "I", and "S" represent baseline only with depth features, image features, and feature sharing respectively. The best results are marked with **bold**. ↓ means smaller is better.

4.3. Model analysis & ablation studies

Impact of image reconstruction. Our proposed auxiliary image reconstruction can largely facilitate depth completion. To justify this, we set the baseline B as the combination of the feature encoder and depth completion module. Based on it, $B + I$ denotes the incorporation of the image reconstruction module but without feature sharing, while $B+I+S$ is our ultimate model with shared features. The quantitative comparison in terms of RMSE and MAE is reported in Table 2. With only image reconstruction as an additional task but no shared features, depth completion performance is slightly boosted. This is mainly because more parameters are introduced but the image features are not sufficiently transferred to the depth completion network. Feature sharing between depth and image modules enables the depth completion network to better take advantage of image features, and thus the overall performance is further improved. Fig. 7 shows qualitative comparisons, where after feature sharing, the model performs better in recovering consistent object structures and small/thin objects.

[1]These reconstructed images displayed in Fig. 6 are less comparable to the original images from appearance. However, for image reconstruction, we only care about the object structures it can reveal rather than the specific intensity.



(a) Different input density (b) Different semantic classes (c) Different distance ranges

Figure 8. Quantitative comparison with the baseline and state-of-the-art methods Glob_guide [38], S2D [26] in three cases on the KITTI validation set. "B", "I", and "S" represent baseline only with depth features, image features, and feature sharing respectively. Our model performs consistently better in all cases.

	RMSE ↓	REL ↓	$\delta_{1.25}$ ↑	$\delta_{1.25^2}$ ↑	$\delta_{1.25^3}$ ↑
Bilateral [33]	0.479	0.084	92.4	97.6	98.9
TVG [13]	0.635	0.123	81.9	93.0	96.8
Zhang *et al*. [44]	0.228	0.042	97.1	99.3	99.7
Ma *et al*. [27]	0.204	0.043	97.8	99.6	99.9
Nconv-CNN [12]	0.129	0.018	99.0	99.8	**100**
CSPN [7]	0.117	**0.016**	99.2	**99.9**	**100**
DeepLiDAR [30]	**0.115**	0.022	**99.3**	**99.9**	**100**
Ours	0.125	0.030	99.1	99.8	**100**

Table 3. Quantitative comparisons on the NYUv2 dataset. Note that ours is the only one that does not use the image during testing, while others take the image as an additional input at both training and testing stages. Best results are marked with **bold**. ↓ means smaller is better, and ↑ means larger is better.

Robustness to input density. We randomly drop depth points in the sparse input with different ratios, and compare RMSE with the baseline and other two state-of-the-art methods Glob_guide [38] and S2D [26] in Fig. 8(a). Our model performs consistently better than others, indicating its robustness to input sparsity.

Comparison in different semantic classes. To validate that our model is able to acquire semantically meaningful image features and use them to facilitate depth completion, we compare results within different semantic classes. Specifically, we fine-tune the off-the-shelf PSPNet [47] pre-trained on Cityscapes [9] with 400 labelled images from the KITTI Semantic Segmentation Benchmark [1]. We use this model to generate semantic masks for the KITTI validation set. We calculate the RMSE of depth completion in six representative classes, *i.e*., Road, Car, Tree, Building, Pole, and Traffic sign, as shown in Fig. 8(b). The performance in Road and Building classes of different methods is similar, mainly because these large and flat regions have more depth points in the input and thus are easier to complete. Our model performs significantly better than others in cars, trees, poles, and traffic signs, which tend to have more specific boundaries and smaller structures. The good performance largely benefits from the effective understanding of object structures with auxiliary image reconstruction and feature sharing.

Results in different distance ranges. Next, we compare completion results in different distance ranges. As illustrated in Fig. 8(c), our model performs slightly better in near regions (0-40m) but significantly better in distant regions (40-80m). This is mainly owing to (1) the use of ℓ_2 loss which penalizes more on larger errors that mostly exist in distant regions, and (2) the image features can reflect the global structure like the road (see Fig. 5(c)) which facilitates our model with a better discrimination in near and distant regions. Besides, the results in the nearest regions, *i.e*., 0-20m, are competitive to others, which is not properly reflected by iRMSE and iMAE.

Generalization to indoor scenes. Finally, we study the generalization ability of our model in indoor scenes, *i.e*., NYUv2 [34]. Following [27], we only retain 500 points in each depth map, the same for other methods we compare. We re-train our network *from scratch* with this new dataset (nearly 50K images from 249 scenes for training, and 654 for testing). The evaluation metrics are RMSE, REL (mean absolute relative error), and the percentage of completed depth with both the relative error and its inverse under a threshold t, *i.e*., $t = 1.25, 1.25^2, 1.25^3$. The quantitative results are reported in Table 3. Note that all the methods for comparison take the RGB image as an additional input. Our model outperforms non-learning based Bilateral [33] and TVG [13], and deep learning methods Zhang *et al*. [44], Ma *et al*. [27] and NConv-CNN [12] in terms of RMSE. Our performance is also comparable to CSPN [7] and DeepLiDAR [30]. In summary, our model can also generalize well to other datasets, and thus is a generic approach for depth completion that only takes sparse depth as input.

5. Conclusion

In this paper, we propose a depth completion model that takes sparse depth as the only input and outputs dense depth and a reconstructed image simultaneously. The auxiliary learning of image reconstruction from sparse depth during training enables the depth completion network to acquire more complementary image features for understanding object structures. It mostly overcomes the shortcomings of existing depth-only approaches due to the lack of semantic cues from images. Future work can be recovering other useful information from sparse depth if ground truth is available, *e.g*., semantic labels, to facilitate depth completion.

Acknowledgement: Dr Liang Zheng is the recipient of an Australian Research Council Discovery Early Career Award (DE200101283) funded by the Australian Government.

References

[1] Hassan Alhaija, Siva Mustikovela, Lars Mescheder, Andreas Geiger, and Carsten Rother. Augmented reality meets computer vision: Efficient data generation for urban driving scenes. *International Journal of Computer Vision (IJCV)*, 2018. 8

[2] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Multi-task feature learning. In *Advances in neural information processing systems*, pages 41–48, 2007. 2

[3] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Multi-task feature learning. In *Advances in neural information processing systems*, pages 41–48, 2007. 3

[4] Amir Atapour-Abarghouei and Toby P Breckon. Veritatem dies aperit-temporally consistent depth prediction enabled by a multi-task geometric and semantic scene understanding approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3373–3384, 2019. 2

[5] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3d object detection for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2147–2156, 2016. 1

[6] Yun Chen, Bin Yang, Ming Liang, and Raquel Urtasun. Learning joint 2d-3d representations for depth completion. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 2, 6

[7] Xinjing Cheng, Peng Wang, and Ruigang Yang. Depth estimation via affinity learned with convolutional spatial propagation network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 103–119, 2018. 1, 2, 7, 8

[8] Nathaniel Chodosh, Chaoyang Wang, and Simon Lucey. Deep convolutional compressed sensing for lidar depth completion. *arXiv preprint arXiv:1803.08949*, 2018. 2, 7

[9] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 8

[10] Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. Boosting for transfer learning. In *Proceedings of the 24th international conference on Machine learning*, pages 193–200. ACM, 2007. 3

[11] Abdelrahman Eldesokey, Michael Felsberg, and Fahad Shahbaz Khan. Confidence propagation through cnns for guided sparse depth regression. *IEEE transactions on pattern analysis and machine intelligence*, 2019. 1, 2, 7

[12] Abdelrahman Eldesokey, Michael Felsberg, and Fahad Shahbaz Khan. Confidence propagation through cnns for guided sparse depth regression. *IEEE transactions on pattern analysis and machine intelligence*, 2019. 2, 6, 8

[13] David Ferstl, Christian Reinbacher, Rene Ranftl, Matthias Rüther, and Horst Bischof. Image guided depth upsampling using anisotropic total generalized variation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 993–1000, 2013. 8

[14] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox. Rgb-d mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments. *The International Journal of Robotics Research*, 31(5):647–663, 2012. 1

[15] Zixuan Huang, Junming Fan, Shuai Yi, Xiaogang Wang, and Hongsheng Li. Hms-net: Hierarchical multi-scale sparsity-invariant network for sparse depth completion. *arXiv preprint arXiv:1808.08685*, 2018. 2

[16] Saif Imran, Yunfei Long, Xiaoming Liu, and Daniel Morris. Depth coefficients for depth completion. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[17] Maximilian Jaritz, Raoul De Charette, Emilie Wirbel, Xavier Perrotton, and Fawzi Nashashibi. Sparse and dense data with cnns: Depth completion and semantic segmentation. In *2018 International Conference on 3D Vision (3DV)*, pages 52–60. IEEE, 2018. 2, 6, 7

[18] Maximilian Jaritz, Raoul De Charette, Emilie Wirbel, Xavier Perrotton, and Fawzi Nashashibi. Sparse and dense data with cnns: Depth completion and semantic segmentation. In *2018 International Conference on 3D Vision (3DV)*, pages 52–60. IEEE, 2018. 2

[19] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7482–7491, 2018. 2

[20] Christian Kerl, Jorg Stuckler, and Daniel Cremers. Dense continuous-time tracking and mapping with rolling shutter rgb-d cameras. In *Proceedings of the IEEE international conference on computer vision*, pages 2264–2272, 2015. 1

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[22] Wonhee Lee, Joonil Na, and Gunhee Kim. Multi-task self-supervised object detection via recycling of bounding box annotations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4984–4993, 2019. 2

[23] Ming Liang, Bin Yang, Yun Chen, Rui Hu, and Raquel Urtasun. Multi-task multi-sensor fusion for 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7345–7353, 2019. 2

[24] Lukas Liebel and Marco Körner. Auxiliary tasks in multi-task learning. *arXiv preprint arXiv:1805.06334*, 2018. 2

[25] Yaojie Liu, Amin Jourabloo, and Xiaoming Liu. Learning deep models for face anti-spoofing: Binary or auxiliary supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 389–398, 2018. 2

[26] Fangchang Ma, Guilherme Venturelli Cavalheiro, and Sertac Karaman. Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. In *2019 IEEE International Conference on Robotics and Automation (ICRA)*, 2019. 1, 2, 7, 8

[27] Fangchang Ma and Sertac Karaman. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In

2018 IEEE International Conference on Robotics and Automation (ICRA), pages 1–8. IEEE, 2018. 2, 8

[28] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017. 5

[29] Quang-Hieu Pham, Thanh Nguyen, Binh-Son Hua, Gemma Roig, and Sai-Kit Yeung. Jsis3d: Joint semantic-instance segmentation of 3d point clouds with multi-task pointwise networks and multi-value conditional random fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8827–8836, 2019. 2

[30] Jiaxiong Qiu, Zhaopeng Cui, Yinda Zhang, Xingdi Zhang, Shuaicheng Liu, Bing Zeng, and Marc Pollefeys. Deeplidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1, 2, 7, 8

[31] Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J. Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1

[32] Bernardino Romera-Paredes, Andreas Argyriou, Nadia Berthouze, and Massimiliano Pontil. Exploiting unrelated tasks in multi-task learning. In *International conference on artificial intelligence and statistics*, pages 951–959, 2012. 2, 3

[33] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision*, pages 746–760. Springer, 2012. 8

[34] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision*, pages 746–760. Springer, 2012. 8

[35] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 3

[36] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *2017 International Conference on 3D Vision (3DV)*, pages 11–20. IEEE, 2017. 1, 2, 5, 7

[37] Abhinav Valada, Noha Radwan, and Wolfram Burgard. Deep auxiliary learning for visual localization and odometry. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6939–6946. IEEE, 2018. 2

[38] Wouter Van Gansbeke, Davy Neven, Bert De Brabandere, and Luc Van Gool. Sparse and noisy lidar completion with rgb guidance and uncertainty. In *International Conference on Machine Vision Applications (MVA)*, 2019. 2, 7, 8

[39] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q. Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1

[40] Yan Xu, Xinge Zhu, Jianping Shi, Guofeng Zhang, Junhu Bao, and Hongsheng Li. Depth completion from sparse lidar data with depth-normal constraints. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019. 1, 2, 7

[41] Yanchao Yang, Alex Wong, and Stefano Soatto. Dense depth posterior (ddp) from single image and sparse range. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[42] Jingwen Ye, Yixin Ji, Xinchao Wang, Kairi Ou, Dapeng Tao, and Mingli Song. Student becoming the master: Knowledge amalgamation for joint scene parsing, depth estimation, and more. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1

[43] Junbo Zhang, Guangjian Tian, Yadong Mu, and Wei Fan. Supervised deep learning with auxiliary networks. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 353–361. ACM, 2014. 2

[44] Yinda Zhang and Thomas Funkhouser. Deep depth completion of a single rgb-d image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 175–185, 2018. 8

[45] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Yan Yan, Nicu Sebe, and Jian Yang. Pattern-affinitive propagation across depth, surface normal and semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1

[46] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Yan Yan, Nicu Sebe, and Jian Yang. Pattern-affinitive propagation across depth, surface normal and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4106–4115, 2019. 2

[47] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 8

[48] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised event-based learning of optical flow, depth, and egomotion. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1