RGB-Depth Fusion GAN for Indoor Depth Completion

Haowen Wang, Mingyuan Wang, Zhengping Che, Zhiyuan Xu, Xiuquan Qiao, Mengshi Qi, Feifei Feng, Jian Tang
1 State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications
2 AI Innovation Center, Midea Group
3 School of Computer Science, Beijing University of Posts and Telecommunications
{hw.wang, mingyuan, qiaoxq, qms}@bupt.edu.cn {chezp, xuzy70, feifei.feng, tangjian22}@midea.com

Abstract

The raw depth image captured by the indoor depth sensor usually has an extensive range of missing depth values due to inherent limitations such as the inability to perceive transparent objects and limited distance range. The incomplete depth map burdens many downstream vision tasks, and a rising number of depth completion methods have been proposed to alleviate this issue. While most existing methods can generate accurate dense depth maps from sparse and uniformly sampled depth maps, they are not suitable for complementing the large contiguous regions of missing depth values, which is common and critical. In this paper, we design a novel two-branch end-to-end fusion network, which takes a pair of RGB and incomplete depth images as input to predict a dense and completed depth map. The first branch employs an encoder-decoder structure to regress the local dense depth values from the raw depth map, with the help of local guidance information extracted from the RGB image. In the other branch, we propose an RGB-depth fusion GAN to transfer the RGB image to the fine-grained textured depth map. We adopt adaptive fusion modules named W-AdaIN to propagate the features across the two branches, and we append a confidence fusion head to fuse the two outputs of the branches for the final depth map. Extensive experiments on NYU-Depth V2 and SUN RGB-D demonstrate that our proposed method clearly improves the depth completion performance, especially in a more realistic setting of indoor environments with the help of the pseudo depth map.

1. Introduction

Nowadays, depth sensors have been widely used to provide reliable 3D spatial information in a variety of applications, such as augmented reality, indoor navigation, and 3D reconstruction tasks. However, most existing commercial depth sensors (e.g., Kinect, RealSense, and Xtion) for indoor spatial perception are not powerful enough to generate a precise and lossless depth map, as shown in the top row of Fig. 1. These sensors often produce many hole regions with invalid depth pixels due to transparent, shining, and dark surfaces as well as too close or too far edges, and these holes significantly affect the performance of downstream tasks on the depth maps (a.k.a., depth images). To address the issue from imperfect depth maps, there have been a lot of approaches to reconstruct the whole depth map from the raw depth map, called depth completion. As RGB images provide rich color and texture information compared with depth maps, the aligned RGB image is commonly used to guide the depth completion of a depth map. To be more specific, the depth completion task is usually conducted as using a pair of raw depth and RGB images captured by one depth sensor to complete and refine the depth values.

Recent studies have produced significant progress in depth completion tasks with convolutional neural networks (CNNs). Ma and Karaman introduced an encoder-decoder network to directly regress
the dense depth map from a sparse depth map and an RGB image. The method has shown great progress compared to conventional algorithms [21,34,39], but its predicted dense depth maps are often too blurry. To further generate a more refined completed depth map, lots of works have recently arisen, which can be divided into two groups with different optimization methods. The first group of works [3,22,29] learn affinities for relative pixels and iteratively refine depth predictions. These methods highly rely on the accuracy of the raw global depth map and suffer the inference inefficiency. Other works [12,17,18,32] analyze the geometric characteristic and adjust the feature network structure accordingly, for instance, by estimating the surface normal or projecting depth into discrete planes. These methods require depth completeness without missing regions, and the model parameters may not be efficiently generalized to different scenes. In any case, the RGB image is merely used as superficial guidance or auxiliary information, and few methods deeply consider the textural and contextual information. At this point, the depth completion task is more or less degraded to a monocural depth estimation task that is conceptually simple but practically difficult.

More remarkably, most of the above methods [3,18,23] uniformly randomly sample a certain number of valid pixels from the dense depth image \(d_{\text{raw}}\) and \(d_{\text{gt}}\) to mimic the sparse depth map \(d^*\) for training and evaluation, respectively. Such sampling strategy is credible in some scenes, such as the outdoor range-view depth map generated by LiDAR. However, the sampled patterns are quite different from the real missing patterns, such as the large missing regions and semantic missing patterns shown in Fig. 1, in indoor depth maps. Therefore, though existing methods are shown to be effective for completing uniformly sparse depth maps, it remains unverified whether they perform well enough for indoor depth completion.

To solve these problems, we propose a novel two-branch end-to-end network to generate a completed dense depth map for indoor environments. Inspired by generative adversarial networks (GANs) [14,15,24,27], we introduce the RGB-depth Fusion GAN (RDF-GAN) for fusing an RGB image and a depth map. RDF-GAN maps a conditional RGB image from the RGB domain to a dense depth map from the depth domain through the latent spatial vector generated by the incomplete depth map. We further design a constraint network to restrict the depth values of the fused map, with the help of weighted-adaptive instance normalization (W-AdaIN) modules and a local guidance module. Afterwards, a confidence fusion head concludes the final depth map completion.

In addition, we propose an exploitation technique, which samples raw depth images to produce pseudo depth maps for training. According to the characteristic of the indoor depth missing, we utilize the RGB images and semantic labels to produce masking regions for raw depth maps, which is more realistic than the simple uniform sampling. Experiments show that the model learning from pseudo depth maps can more effectively fill in large missing regions for raw depth images captured indoors.

Our main contributions are summarized as the following:

- We propose a novel end-to-end GAN-based network, which effectively fuses a raw depth map and an RGB image to reproduce a reasonable dense depth map.
- We design and utilize the pseudo depth maps, which are in line with the raw depth missing distribution in indoor scenarios. Training with pseudo depth maps significantly improves the model’s depth completion performance, especially in more realistic settings of indoor environments.
- Our proposed method achieves the state-of-the-art performance on NYU-Depth V2 and SUN RGB-D for depth completion and proves its effectiveness in improving downstream task performance such as object detection.

2. Related Work

**Depth Completion.** Recent works have extensively applied deep neural networks for depth estimation and completion tasks with remarkable improvements. Ma and Karaman [23] used an encoder-decoder structure with CNNs to predict the full-resolution depth image directly from a set of depth samples and RGB images. On this basis, some methods incorporating additional output branches to assist in the generation of depth maps have been proposed. Qiu et al. [32] produced dense depth using the surface normal as the intermediate representation. Huang et al. [12] applied the boundary consistency to solve the issue of vague structures. Lee et al. [18] introduced the Plane-Residual representation to interpret depth information and factorized the depth regression problem into a combination of discrete depth plane classification and plane-by-plane residual regression. Zhang et al. [40] uses GANs to solve both semantic segmentation and depth completion tasks in outdoor scenarios. Cheng et al. [3] proposed the convolutional spatial propagation network (CSPN) and generated the long-range context through a recurrent operation to lessen the burden of directly regressing the absolute depth information. Park et al. [29] improved CSPN by non-local spatial and global propagations. These methods prove that the encoder-decoder network can effectively perform depth completion and obtain a more refined depth map through additional optimization. In this work, we extend the encoder-decoder structure to build our depth completion model.

**RGB-D Fusion.** The fusion of both RGB and depth data (a.k.a., the RGB-D fusion) is essential in many tasks such as semantic segmentation and depth completion. While most existing methods [23,25] only concatenate aligned pixels
from RGB and depth features, more effective and advanced RGB-D fusions have been proposed recently. Cheng et al. [4] designed a gated fusion layer to learn the different weights of each modality in different scenes. Park et al. [30] fused multi-level RGB-D features in a very deep network through residual learning. Du et al. [6] proposed a novel cross-modal translate network to represent the complementary information and enhance the discrimination of extracted features. In this work, we design the two-branch structure and the W-AdaIN modules to better capture and fuse RGB and depth features.

**Generative Adversarial Networks.** Generative adversarial networks (GANs) have achieved great success in a variety of image generation tasks such as image-style transfer, realistic image generation, and image synthesis. Mirza et al. [27] proposed the conditional GAN to direct the data generation process by combining the additional information as a condition. Karras et al. [15] introduced a style-based GAN to embed the latent code into a latent space to affect the variations of generated images. Ma et al. [24] proposed a GAN for infrared and visible images. In this work, we use a GAN-based structure fusing RGB images and depth maps to generate dense depth maps with fine-grained textures.

### 3. Method

In this section, we describe our end-to-end depth completion method, as shown in Fig. 2. The proposed model takes a raw (noisy and possibly incomplete) depth map and its corresponding RGB image as the input, and outputs the completed and refined dense depth map estimation. The model mainly consists of two branches: a constraint network branch (Section 3.1) and an RGB-depth Fusion GAN (RDF-GAN) branch (Section 3.2). The constraint network and RDF-GAN take a depth map and an RGB image as the input, respectively, and produce their depth completion results. To fuse the representations between the two branches, a local guidance module and a series of intermediate fusion modules called W-AdaIN (Section 3.3) are deployed at different stages of the model. Finally, a confidence fusion head (Section 3.4) combines the outputs of the two channels and provides more reliable and robust depth completion results. Moreover, we introduce the training strategy with pseudo depth maps (Section 3.5) and describe the overall loss function for training (Section 3.6).

#### 3.1. Constraint Network Branch

The first branch is composed of a constraint network, which reproduces a local full-resolution depth map and a confidence map through a convolutional encoder-decoder structure. The encoder-decoder structure is based on ResNet-18 [10] and pre-trained on the ImageNet dataset [5]. As illustrated in Fig. 3 and the bottom-left part of Fig. 2, given the raw depth image $d_{raw} \in \mathbb{R}^{H \times W \times 1}$ and the RGB image $r$, the network outputs a dense local depth map $d_l \in \mathbb{R}^{H \times W \times 1}$ and a local confidence map $c_l \in \mathbb{R}^{H \times W \times 1}$.

The input of this branch is a concatenation of the one-channel raw depth image $d_{raw}$ and the two-channel local guidance map $g$ from the RGB image. Given this input, the encoder downsamples the feature size to $H/2^3 \times W/2^3$ and expands the feature dimension to 512. The encoder $M(\cdot)$ learns the mapping from the depth map to the depth latent space $z$ as the fused depth feature information for RDF-GAN. The decoding stage applies a set of upsampling
blocks to increase the feature resolution with skip connection from the encoder. The output of the decoder is a local depth map and its corresponding local confidence map.

### 3.2. RDF-GAN Branch

To generate the fine-grained textured and dense depth map, we propose the second branch in our model, which is a GAN-based structure for RGB and depth image fusion. Different from most existing fusion methods that directly concatenate inputs from different domains, our fusion model, named as RDF-GAN, is inspired by the conditional GAN [13]. As illustrated in the top-left part of Fig. 2, we use the depth latent vector mapping from the encoder. The output of the decoder is a local guidance map and a fused confidence map, and use a discriminator to distinguish the real (ground truth) depth images from generated ones. The generator $G(z)$ has a similar structure to the constraint network. Given the cross-responding RGB image $r$ as the condition, the generator $G(\cdot)$ with the depth latent vector $z$ generates a fused dense depth map $d_f$ and a fused confidence map $c_f \in \mathbb{R}^{H \times W \times 1}$ for the scene. The latent vector $z$ propagates the depth information to the RGB image using the proposed W-AdaIN described in Section 3.3. We distinguish the fused depth map $d_f$ and the real depth image $d_{gt}$ by the discriminator $D(\cdot)$, whose structure is based on PatchGAN [18]. We adopt the objective function of WGAN [9] for training RDF-GAN. To be more specific, the RDF-GAN loss includes the discriminator loss $L_D$ and the generator loss $L_G$:

$$L_D = \mathbb{E}_{d_{raw} \sim D_{raw}} [D(G(M(d_{raw})))|r] - \mathbb{E}_{d_{gt} \sim D_{gt}} [D(d_{gt})],$$

(1)

$$L_G = \lambda_g L_1(G(M(d_{raw}))) - \mathbb{E}_{d_{raw} \sim D_{raw}} [D(G(M(d_{raw})))|r],$$

(2)

where $d_{raw}$ and $d_{gt}$ are the raw and ground-truth depth images drawn from the domains $D_{raw}$ and $D_{gt}$, respectively.

### 3.3. Feature Fusion Modules

To allow the feature information to be shared across all stages of the two branches, we design the local guidance module and W-AdaIN and apply them in the network.

**Local Guidance Module.** We adopt U-Net [33] as a feature extractor to produce a local guidance map $g \in \mathbb{R}^{H \times W \times 2}$ from an RGB image $r \in \mathbb{R}^{H \times W \times 3}$. The first and the second channels of the local guidance map represent the foreground probability and semantic features, respectively. Therefore, the local guidance module can guide the constraint network to focus on local depth correlations.

**W-AdaIN.** As shown in Fig. 4, we project depth pixels of the depth map into multiple discretized depth planes, according to the distance between the depth pixels and a pre-defined set of discrete depth values. Local regions are easier to be classified into the same depth plane because they have similar depth values. We also find that similar color gradations in a local region usually have similar depth values. Hence, we propose a W-AdaIN module for fusing the features of RGB and depth images. It is extended from AdaIN [15] and is defined as:

$$\text{W-AdaIN}(z, f_r) = A \cdot y_b \cdot \left( f_r - \mu(f_r) \right) + B \cdot y_b, \quad (3)$$

where $f_r$ is the feature map of RGB image; $A = \text{Attention}(z)$ and $B = \text{Attention}(f_r)$ are the weight matrices that are generated by the self-attention mechanism [38] on $z$ and $f_r$, respectively; $y_b$ and $y_b$ are the spatial scaling and bias factors obtained by affine transformations [15] with the latent matrix $z$; $\mu(\cdot)$ and $\sigma(\cdot)$ are the mean and variance, respectively. By its design, $A$ assigns similar weight values to the regions with similar depth values. Similarly, $B$ smoothes the depth blocks by assigning similar weight values of the local similar color gradations.
We design five methods to obtain the pseudo depth map:

1. **Highlight masking.** We segment the regions of probably specular highlights [1] in RGB images and mask them in raw depth maps.

2. **Black masking.** We randomly mask the depth pixels whose RGB values are all in [0, 5] (i.e., dark pixels).

3. **Graph-based segmentation masking.** We mask the probably noisy pixels of depth maps obtained by graph-based segmentations [7] on RGB images.

4. **Semantic masking.** As depth values for objects with some particular materials are usually missing, we mask one or two objects randomly by their semantic labels and only keep depth pixels on their edges.

5. **Semantic XOR masking.** We train U-Net [33] on 20% of the training set of RGB images and use the trained model to segment the other RGB images. We mask the depth pixels where the segmentation result and ground-truth are different, i.e., conducting the XOR operation on the segmentation results and the ground-truth to obtain the masking.

Finally, we randomly pick and combine the mask from the above five methods to generate the pseudo depth map, mimicking a more plausible missing depth distribution. The pseudo depth maps are used to train a more robust depth completion model for indoor scenarios. More details can be found in Section 2 of the supplementary.

### 3.6. Loss Function

We use the $L_1$ loss on the local depth map and final prediction. The overall loss function is defined as:

$$L_{overall} = L_D + L_G + \lambda_g L_1(d_i) + \lambda_{pred} L_1(d_{pred}),$$

where $\lambda_g$ in Eq. 2, $\lambda_1$, and $\lambda_{pred}$ are weight hyperparameters for different terms in the loss function, which are set to be 0.5, 1, and 10, respectively.

### 4. Experiments

#### 4.1. Datasets and Metrics

We conducted experiments on two widely-used benchmarks: NYU-Depth V2 [28] and SUN RGB-D [36].

**NYU-Depth V2.** The NYU-Depth V2 dataset [28] contains pairs of RGB and depth images collected from Microsoft Kinect in 464 indoor scenes. Densely labeled image pairs are split into the training set with 795 images and the test set with 654 images, and each set includes RGB images, raw depth images from sensors, labeled (reconstructed) depth maps, and segmentation masks. Following existing methods, we utilized the unlabeled ~50K images for training and the labeled 654 images in the test set for evaluation. The input images were resized to $320 \times 240$ and center-cropped cropped to $304 \times 228$. 
Table 1. Quantitative results on the NYU-Depth V2 dataset. \( \mathcal{R} \) and \( \mathcal{T} \) represent the raw and reconstructed depth map, respectively. \( * \) represents the random sparse sampling, where Sparse2Dense and DGCG in \( \mathcal{T} \Rightarrow \mathcal{T} \) use 200 pixels and others use 500 pixels.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Method</th>
<th>RMSE ↓</th>
<th>Rel ↓</th>
<th>( \delta_{1.25} ) ↑</th>
<th>( \delta_{1.25^3} ) ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathcal{R} \Rightarrow \mathcal{T} )</td>
<td>DC-BCS [12]</td>
<td>0.271</td>
<td>0.016</td>
<td>98.1</td>
<td>99.1</td>
</tr>
<tr>
<td></td>
<td>RGB-GU [17]</td>
<td>0.260</td>
<td>0.017</td>
<td>97.9</td>
<td>99.3</td>
</tr>
<tr>
<td></td>
<td>MS-CNN [18]</td>
<td>0.190</td>
<td>0.018</td>
<td>98.8</td>
<td>99.7</td>
</tr>
<tr>
<td></td>
<td>DM-LRN [19]</td>
<td>0.205</td>
<td>0.014</td>
<td>98.8</td>
<td>99.6</td>
</tr>
<tr>
<td></td>
<td>NLSNP [20]</td>
<td>0.153</td>
<td>0.015</td>
<td>98.6</td>
<td>99.6</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>0.139</td>
<td>0.013</td>
<td>98.7</td>
<td>99.6</td>
</tr>
</tbody>
</table>

| \( \mathcal{R}^* \Rightarrow \mathcal{T} \) | Sparse2Dense [23] | 0.335   | 0.060  | 94.2                 | 97.1                 | 98.8 |
|         | CSPN [3]      | 0.500   | 0.139  | 85.7                 | 92.9                 | 96.3 |
|         | NLSNP [29]    | 0.348   | 0.043  | 93.0                 | 96.7                 | 98.5 |
|         | Ours          | 0.309   | 0.053  | 93.6                 | 97.6                 | 99.0 |

| \( \mathcal{T}^* \Rightarrow \mathcal{T} \) | Sparse2Dense [23] | 0.230   | 0.044  | 97.1                 | 99.4                 | 99.8 |
|         | CSPN [3]      | 0.117   | 0.016  | 99.2                 | 99.9                 | 100.0 |
|         | 3coefficient [13] | 0.131   | 0.013  | 97.9                 | 99.3                 | 99.8 |
|         | DGGCG [17]    | 0.225   | 0.046  | 97.2                 | 99.7                 | 100.0 |
|         | DeepLidar [32] | 0.115   | 0.022  | 99.3                 | 99.9                 | 100.0 |
|         | NLSNP [29]    | 0.092   | 0.012  | 99.6                 | 99.9                 | 100.0 |
|         | PRR [18]      | 0.104   | 0.014  | 99.4                 | 99.9                 | 100.0 |
|         | Ours          | 0.103   | 0.016  | 99.4                 | 99.9                 | 100.0 |

Table 2. Quantitative results on the SUN RGB-D dataset.

\( \mathcal{R} \Rightarrow \mathcal{T} \) | RMSE ↓  | Rel ↓  | \( \delta_{1.25} \) ↑ | \( \delta_{1.25^3} \) ↑ |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sparse2Dense [23]</td>
<td>0.329</td>
<td>0.074</td>
<td>93.9</td>
<td>97.0</td>
</tr>
<tr>
<td>CSPN [3]</td>
<td>0.295</td>
<td>0.137</td>
<td>95.6</td>
<td>97.5</td>
</tr>
<tr>
<td>DeepLidar [32]</td>
<td>0.279</td>
<td>0.061</td>
<td>96.9</td>
<td>98.0</td>
</tr>
<tr>
<td>NLSNP [29]</td>
<td>0.207</td>
<td>0.063</td>
<td>97.3</td>
<td>98.1</td>
</tr>
<tr>
<td>Ours</td>
<td>0.255</td>
<td>0.059</td>
<td>96.9</td>
<td>98.4</td>
</tr>
</tbody>
</table>

SUN RGB-D. The SUN RGB-D dataset [36] contains 10,335 RGB-D images captured by four different sensors. This dataset, with different scenes and sensors, is diverse and helpful to effectively evaluate model generalization. Besides, its dense semantic annotations and 3D bounding boxes enable the evaluations of more training strategies and downstream tasks. Following the official split, we used 4,845 images for training and 4,659 for testing in 19 major scene categories. We used the refined depth map based on multiple frames [36] as the ground truth for evaluations. The input images were resized to 320 × 240 and randomly cropped to 304 × 228.

Evaluation Metrics. We adopted three metrics for the dense depth prediction evaluation: root mean squared error (RMSE), absolute relative error (Rel), and \( \delta_1 \), which is the percentage of predicted pixels whose relative error is within a relative threshold [23].

4.2. Comparisons with State-of-the-Art Methods

NYU-Depth V2. To draw a comprehensive performance analysis, we set up three different training and evaluation schemes. In the test, we use three different inputs to predict and reconstruct depth maps \( \mathcal{R} \) respectively, which are raw depth maps \( \mathcal{R} \), sparse depth maps with randomly sampled 500 valid depth pixels in raw depth map \( \mathcal{R}^* \), and sparse depth maps with randomly sampled 500 valid depth pixels in reconstructed depth map \( \mathcal{T}^* \). For more descriptions of the schemes, please refer to Section 3 in the supplementary. The performance comparison of our method and the other state-of-the-art methods on NYU-Depth V2 are shown in Tab. 1. Given the results, we concluded the following:

- \( \mathcal{R} \Rightarrow \mathcal{T} \): We used the pseudo depth maps generated in Section 3.5 as the input to train the proposed model and NLSNP [29]. Meanwhile, we compared with several baselines [12, 19, 35, 37] that are trained in the synthetic semi-dense sensor data [35]. Compared to all the baselines, our proposed method improves significant performance, especially on RMSE and Rel. We selected two representative scenes and visualized our prediction results in the last column of Fig. 6. The model trained by pseudo depth maps produced more accurate and textured depth predictions in the missing region.

- \( \mathcal{R}^* \Rightarrow \mathcal{T} \): Following the previous works [3, 18, 23, 29], we used the RGB image and the sparse depth map with randomly sampled depth pixels of raw depth image as the input for training. In the test stage, the input was the same as that for training, and the reconstructed depth map was used as the ground truth. We observed that our model outperformed the baseline with big margins on RMSE. The qualitative results were shown in the second and fourth rows of Fig. 6. Our method accurately predicts the contour of the sofa and smooth windows in red boxes compared to other methods. This proves that our dense depth predictions are well integrated with the textural information of RGB images by the RDF-GAN branch.

- \( \mathcal{T}^* \Rightarrow \mathcal{T} \): The setting is consistent with most existing works of depth completion [3, 18, 23, 29]. Our model without any iteration processing is only lower than the NLSNP [32] (but ours is 1.5 × faster in inference time than NLSNP). The visualizations shown in the first and third rows of Fig. 6 as well as Fig. 7 further indicate the superiority of our method.

- As shown in green boxes of Fig. 6, the downsampled input from the reconstructed depth map (\( \mathcal{T}^* \Rightarrow \mathcal{T} \)) reveals ground truth depth values, which is unavailable in practice, to the models. This supported the claim that the raw input setting (\( \mathcal{R} \Rightarrow \mathcal{T} \)) is more practicable for realistic indoor depth completion.

SUN RGB-D. On SUN RGB-D, we adopted the pseudo depth maps as the input and the raw depth data as the ground truth for training. In the test set, the raw depth image and the depth map synthesized by multiple frames were used as the input and the ground truth, respectively. In Tab. 2, our proposed method achieves the best performance in most metrics. From the visualization results in Fig. 1, our model complements the missing depth regions as much as possible.
Figure 6. Depth completion comparisons of different methods with different training strategies and inputs on NYU-Depth V2. The first and third rows take sparse samples on reconstructed depth maps as the input $T^* \Rightarrow T$. The second and fourth rows take sparse samples on raw depth maps as the inputs $R^* \Rightarrow T$. The last column shows the result of our model trained with pseudo maps $R \Rightarrow T$.

Figure 7. Depth completion comparisons on NYU-Depth V2 with $T^* \Rightarrow T$. Our model recovers more textural details in the red boxes.

Table 3. Quantitative comparisons of different $L_1$ loss settings.

<table>
<thead>
<tr>
<th>Setting</th>
<th>$\lambda_g$</th>
<th>$\lambda_l$</th>
<th>$\lambda_{pred}$</th>
<th>RMSE ↓</th>
<th>Rel ↓</th>
<th>$\delta_{1.25}$ ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-</td>
<td>-</td>
<td>✓</td>
<td>0.207</td>
<td>0.032</td>
<td>97.8</td>
</tr>
<tr>
<td>B</td>
<td>✓</td>
<td>-</td>
<td>✓</td>
<td>0.212</td>
<td>0.038</td>
<td>97.8</td>
</tr>
<tr>
<td>C</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
<td>0.174</td>
<td>0.025</td>
<td>98.3</td>
</tr>
<tr>
<td>D</td>
<td>0.5</td>
<td>1</td>
<td>10</td>
<td>0.103</td>
<td>0.016</td>
<td>99.4</td>
</tr>
</tbody>
</table>

Figure 8. Qualitative comparisons of different $L_1$ loss settings.

4.3. Ablation Studies

We conducted ablation studies with the setting of $T^* \Rightarrow T$ on the NYU-Depth V2 dataset.

Settings of $\lambda$s. We investigated the effects on model performance in different settings of $\lambda$s in the loss function, and the results are shown in Tab. 3. We compared the following four settings and found that including all $L_1$ loss terms leads to the best model. In Setting A, we only calculated the $L_1$ loss for the final depth prediction, and in Setting B, both $L_1$ losses for the final depth prediction and fused depth map were calculated. In these two settings, the model overly focused on textural information resulting in generating many local outliers, as shown in Fig. 8(a), and the predicted depth values in many regions had a large deviation from the ground-truth values. In Setting C, we took the $L_1$ losses for the local depth map and the final depth

Table 3. Quantitative comparisons of different $L_1$ loss settings.

with more detailed texture information for different sensors.
Table 4. Ablation study results for different modules. ‘Conv.’ means the convolution operation for the concatenation of the outputs from the two branches. ‘U-Net (I)’ and ‘U-Net (N)’ represent pre-training with ImageNet and NYU-Depth V2, respectively.

<table>
<thead>
<tr>
<th>Module</th>
<th>Method</th>
<th>RMSE ↓</th>
<th>REL ↓</th>
<th>δ_{1.25} ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fusion Head</td>
<td>Conv.</td>
<td>0.118</td>
<td>0.022</td>
<td>99.0</td>
</tr>
<tr>
<td></td>
<td>Confidence Fusion</td>
<td>0.117</td>
<td>0.019</td>
<td>99.1</td>
</tr>
<tr>
<td>Local Guidance</td>
<td>Concat.</td>
<td>0.113</td>
<td>0.017</td>
<td>99.2</td>
</tr>
<tr>
<td></td>
<td>U-Net</td>
<td>0.107</td>
<td>0.016</td>
<td>99.4</td>
</tr>
<tr>
<td></td>
<td>U-Net (I)</td>
<td>0.106</td>
<td>0.016</td>
<td>99.4</td>
</tr>
<tr>
<td></td>
<td>U-Net (N)</td>
<td>0.101</td>
<td>0.015</td>
<td>99.5</td>
</tr>
<tr>
<td>Stage Fusion</td>
<td>IN</td>
<td>0.106</td>
<td>0.016</td>
<td>99.4</td>
</tr>
<tr>
<td></td>
<td>AdaIN</td>
<td>0.110</td>
<td>0.017</td>
<td>99.3</td>
</tr>
<tr>
<td></td>
<td>W-AdaIN</td>
<td>0.103</td>
<td>0.016</td>
<td>99.4</td>
</tr>
</tbody>
</table>

Table 5. Performance comparisons of 3D object detection results with the raw and completed depth maps on SUN RGB-D.

<table>
<thead>
<tr>
<th>Method</th>
<th>mAP@25</th>
<th>mAP@50</th>
</tr>
</thead>
<tbody>
<tr>
<td>VoteNet [31]</td>
<td>59.07</td>
<td>35.77</td>
</tr>
<tr>
<td>Ours+VoteNet [31]</td>
<td>60.64</td>
<td>37.28</td>
</tr>
<tr>
<td>H3DNet [41]</td>
<td>60.11</td>
<td>39.04</td>
</tr>
<tr>
<td>Ours+H3DNet [41]</td>
<td>61.03</td>
<td>39.71</td>
</tr>
</tbody>
</table>

4.4. Object Detection on the Completed Depth Map

We used the completed depth map as the input of the 3D object detection task on the SUN RGB-D dataset [36] to evaluate the quality of our depth completions. Two SOTA models, VoteNet [31] and H3DNet [41], were used as the detectors. Tab. 5 shows that the two models both obtain a significant improvement with our completed depth map. As shown in Fig. 10, the point cloud converted from the completed depth map contains more points and better covers the shape of the object than the raw depth map. More discussions can be found in Section 4 of the supplementary.

5. Conclusion

In this work, we propose a novel two-branch end-to-end network for indoor depth completion. We design the RDF-GAN model to produce the fine-grained textural depth map and restraint it by a constraint network. In addition, we propose a novel and effective sampling method to produce pseudo depth maps for training indoor depth completion models. Extensive experiments have demonstrated that our proposed solution achieves state-of-the-art on the NYU-Depth V2 and SUN RGB-D datasets.

6. Acknowledgements

This work was done when Haowen Wang took an internship at Midea Group. This work was supported in part by the Funds for International Cooperation and Exchange of NSFC under Grant 61720106007, the Fundamental Research Funds for the Central Universities (2019XD-A03-3), the 111 Project under Grant B18008, and Shanghai Pujiang Program (21PJ1420300).
References


[34] Ashutosh Saxena, Sung H Chung, and Andrew Y Ng. Learning depth from single monocular images. In NIPS, 2005. 2
[40] Chongzhen Zhang, Yang Tang, Chaoqiang Zhao, Qiyu Sun, Zhenchye Ye, and Jürgen Kurths. Multitask gans for semantic segmentation and depth completion with cycle consistency. IEEE Transactions on Neural Networks and Learning Systems, 32(12):5404–5415, 2021. 2