Multi-Scale Progressive Fusion-based Depth Image Completion and Enhancement for Industrial Collaborative Robot Applications

Chuhua Xian[1], Jun Zhang[1], Wenhao Yang[2] and Yunbo Zhang[2,3*]

[1]Guangdong Provincial Key Lab of Computational Intelligence and Cyberspace Information, School of Computer Science and Engineering, South China University of Technology, Guangzhou, 510006, Guangdong, China.
[2]Department of Industrial & Systems Engineering, Kate Gleason College of Engineering, Rochester Institute of Technology, 77 Lomb Memorial Dr, Rochester, 14623, NY, USA.
[3]School of Information (iSchool), Golisano College of Computing and Information Sciences, Rochester Institute of Technology, 92 Lomb Memorial Dr, Rochester, 14623, NY, USA.

*Corresponding author(s). E-mail(s): ywzeie@rit.edu;
Contributing authors: chhxian@scut.edu.cn;

Abstract

The depth image obtained by consumer-level depth cameras generally has low resolution and missing regions due to the limitations of the depth camera hardware and the method of depth image generation. Despite the fact that many studies have been done on RGB image completion and super-resolution, a key issue with depth images is that there will be evident jagged boundaries and a significant loss of geometric information. To address these issues, we introduce a multi-scale progressive fusion network for depth image completion and super-resolution in this paper, which has an asymptotic structure for integrating hierarchical features in different domains. We employ two separate branches to learn the features of a multi-scale image given a depth image and its corresponding RGB image. The extracted features are then fused into different level features of these two branches using a step-by-step strategy to recreate the final depth image. To confine distinct borders and geometric features, a multi-dimension loss is also designed. Extensive depth completion and super-resolution studies reveal that our proposed method outperforms state-of-the-art methods both qualitatively and quantitatively. The proposed methods are also applied to two human-robot interaction applications, including a remote-controlled robot based on an unmanned ground vehicle (UGV), AR-based toolpath planning, and automatic toolpath extraction. All these experimental results indicate the effectiveness and potential benefits of the proposed methods.

1 Introduction

In the era of Industry 4.0, adopting collaborative robots (Cobot) has become a new trend in the manufacturing industry [1]. A Cobot has a human-size scale and works alongside human workers in a shared, collaborative workspace [1]. Therefore, the collaboration of humans and robots is enabled

with a higher rate and more flexibility in production. A 2015 paper [2] pointed out that adopting Cobot in assembly achieves the best performance than either manual assembly or fully automated assembly. With the expectations of a high growth rate of automation in manufacturing industries, especially in small and medium-sized enterprises, Fortune Business Insights projects the global market value of the Cobot will increase from $1.36 billion in 2021 to $16.4 billion in 2028 [3].

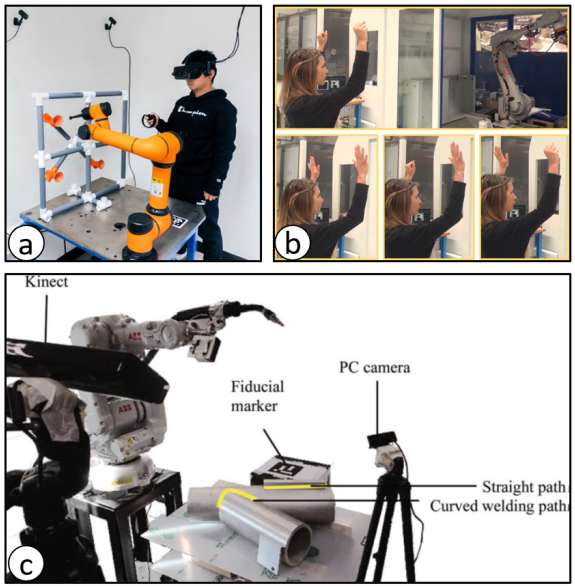

Fig. 1 The depth information is utilized in a variety of human-robot interaction applications, such as: (a) Augmented Reality-based interface [4] (image courtesy of Yang *et al.*); (b) gesture control [5] (image courtesy of Magrini *et al.*); and (c) toolpath extraction [6] (image courtesy of Ni *et al.*).

To enable the coexistence of Cobot and the human operator and the interaction between them, the sensing of the workspace, the human operator, and the workpiece is essential to a Cobot system [7]. Particularly, the depth information is utilized to avoid the collision between the Cobot and human operator [8–11] for safety. Moreover, the depth sensing provides new affordances for ways of human-robot interaction, such as guest control [5, 12] (see Fig. 1 (b)), toolpath extraction [6, 13] (Fig. 1 (c)), navigation and localization [14, 15], and augmented reality (AR)-based interfaces [4, 16–18] (Fig. 1 (a)). The current depth camera equipped with Cobot has low cost and real-time performance, yet it suffers from a lack of depth information and a lower resolution due to its hardware limitations. Specifically, the depth camera, based on either structured light or time of flight techniques, cannot handle a transparent or highly reflective object surface [19], therefore, there are regions with depth information missing. Another cause of missing regions is that the distance between the depth camera and the object is beyond the maximum sensing range of the camera [20]. Additionally, the existing depth cameras are unable to capture high-frequency information, which results in the loss of comprehensive information. Both the incomplete and low-resolution depth information will cause difficulty in the Cobot applications. For example, in Fig. 11 (b), the missing depth causes an incorrect visualization of the Cobot and the environment. For toolpath extraction (see Fig. 1 (c)), the incomplete depth information leads to an incomplete path, and the low-resolution depth leads to an inaccurate path.

Researchers in the computer vision field have proposed a variety of research works in depth image completion. For Cobot applications, the semi-dense depth images are generated using RGB-D cameras, which is different from outdoor applications requiring sparse depth images generated by LiDAR-based sensors [21–23]. The existing completion methods for semi-dense depth images either suffer from low efficiency [20], or a complex network [24] requiring more time to train and generate results. Another recent work [25] has a simpler network, but it processes both RGB information and depth information all together in the same network. Considering that the RGB image contains texture information and the depth image includes geometry information, it is inappropriate to process them together in the same network.

There are also methods dedicated to super-resolution methods for depth images, including optimization-based methods [26–29], and filtering-based methods [30–33]. While optimization-based methods fail to capture the global structure of the objects and are usually time-consuming, filtering-based methods, which utilize high-resolution images to filter the depth images, suffer from obvious edge aliasing and excessive loss of details and may introduce texture artifacts when the RGB image provides biased guidance. In recent years,

researchers have noticed that RGB images have a higher resolution than depth images, and the information in RGB images has the potential to be used to improve depth images. Therefore, convolutional neural networks (CNNs) based methods [34–36], have been proposed to fuse the semantic information of high-resolution RGB images with the features of low-resolution depth images to generate high-resolution depth images. Although some good results are generated [37], the existing CNN-based methods smooth out the sharp edges of objects' boundaries.

To address these aforementioned issues, we propose a network framework for both completion and super-resolution for depth images. The primary contributions of this paper are summarized as follows:

- A new network with a multi-scale progressive fusion strategy is proposed, which can tackle two main problems of the low-quality depth image captured by the RGB-D camera: depth image completion and depth image super-resolution.
- A hybrid training strategy combining random masks and real-synthetic data is proposed so that large-scale RGB-D datasets can be applied to depth image completion.
- A fusion module taking both the color features and the geometric features into consideration is proposed to generate a high-quality depth image.

The rest of this paper is organized as follows: We reviewed the major related works about completion and super-resolution for depth images, and industrial collaborative robot applications relying on depth information in Section 2. Section 3 describes the details of our proposed method, including the network architecture, the depth completion module, and the super-resolution module. In Section 4, we conducted thorough experiments to verify the proposed method and compared it with existing methods. We also tested our method in three HRI applications including: 1) a remote-controlled robot based on an unmanned ground vehicle (UGV) (see Fig. 10), 2) an AR-based toolpath planning (Fig. 11), and 3) an automatic toolpath extraction (Fig. 12). Finally, we concluded our work in Section 5.

2 Related Work

In this section, we will introduce recent developments in completion and super-resolution for depth images, and the collaborative robot applications that rely on depth information.

2.1 Completion for Depth Images

In 2018, Zhang *et al.* [20] proposed to complete a semi-dense depth image with global optimization with the help of surface normal and occlusion boundary from the RGB image. Significantly, Zhang also used multi-view reconstruction to generate a large dataset, which is very helpful for deep network approaches. In 2019, Huang *et al.* [24] based on Zhang's approach, proposed to use the network to get faster and better results. They firstly generated a normal and boundary map from the RGB image, then concatenated the RGB image, predicted normal and boundary maps, and raw depth image as the input of the network. In order to get a depth image with a clear boundary, they added a boundary consistency network. By replacing the Cholesky optimization with a network, they had a faster inference time and were more desirable for real-world applications. In 2020, Senushkin *et al.* [25] proposed a decoder modulation branch adding to the U-net architecture. They input the mask of the missing value and used the decoder modulation branch to control the decoding of a dense depth image. Some specific works were also proposed in these years. Botach et al. [38] proposed an RGB-D dataset of metallic industrial objects and presented the experiments performed for the depth completion task. Tan et al. [39] proposed a Mirror3DNet, which was used for completing the depth on the 3D mirror plane.

2.2 Super-Resolution for Depth Images

According to the different starting points and solutions, the related work of super-resolution for depth images can be classified into three categories: local super-resolution, global super-resolution, and learning-based super-resolution.

Local Super-Resolution Local methods usually consider the use of high-resolution RGB image guidelines and local pixel relationships to do up-sampling for low-resolution depth images. Joint Bilateral Up-sampling (JBU) [31] considered the

Gaussian distance of HR images and LR images in the spatial domain to up-sample the depth image. Liu *et al.* [29] extended Kopf's work [31], considering the geodesic paths of depth pixels based on joint filtering. Yang *et al.* [27] presented a framework including cost volume and sub-pixel refinement to produce a high-resolution depth image. Choi [40] proposed different up-sampling strategies for continuous and discontinuous regions in the depth image. For depth-discontinuous areas, the depth-histogram-based method they proposed made the recovered depth boundary sharper.

Global Super-Resolution This type of method usually considers the correlation between RGB images and depth images and treats the depth super-resolution task as a global optimization problem on this basis. Diebel [26] was the first to apply Markov Random Fields (MRF) to generate high-resolution depth images. Xie *et al.* [41] introduced the self-similarity and the guidance of a high-resolution edge map for depth super-resolution on the basis of MRF, which also achieved better results. Li *et al.* [42] proposed a cascaded global interpolation framework to recover the high-resolution depth image.

Learning-based Super Resolution With the development of deep learning-based methods in image processing, many learning-based super-resolution methods have also been extensively developed. Dong *et al.* [43] found the underlying relationship between high-resolution images and low-resolution images through a deep CNN network, which provided a non-linear mapping learning ability between image pairs. Guo *et al.* [44] presented residuals at different resolutions to guide LR depth images for accurate interpolations. Voynov *et al.* [36] proposed perceptual metrics to constrain the network to recover high-resolution depth images. Experiments proved that this kind of quality measure, which is similar to human perception, is more reasonable. Wang *et al.* [45] proposed a cascaded restoration network that considered the edge and color information of the input image. Experimental results showed that the restoration module, including edge information, improved the boundary resolution of the recovered depth.

2.3 Industrial Collaborative Robot Applications Relying on Depth Information

The depth information is essential for a variety of industrial collaborative robot applications, including collision avoidance, automatic toolpath generation, grasp generation, and human-robot interaction. These applications indicate the importance of depth information in human-robot collaboration.

Collision Avoidance Calcagni *et al.* [46] presented a collision avoidance strategy based on three Intel Realsense D455 RGBD cameras. Similarly, a dynamic obstacle avoidance strategy is developed by Dumonteil *et al.* [47], which relies on the depth information captured by a single Kinect sensor. Ragaglia *et al.* [9] utilized the depth information to avoid the robot colliding with human operators during the collaboration. A 2021 paper [48] provides a comprehensive review of control strategies based on depth sensing to avoid collision.

Automatic Toolpath Generation The depth information has been used to generate robots' toolpaths automatically. Zaki [49] presented a toolpath generation workflow consisting of 3D scanning using a depth camera, surface reconstruction, and automatic toolpath generation. Gómez-Espinosa *et al.* [13] proposed and developed an automatic toolpath generation for welding robots. The geometric features, such as lines and curves, are extracted for the welding robot toolpaths, taking the depth information of the 3D part as the input. Another related work [6] also focuses on generating toolpaths for welding based on depth information but incorporates the human operator's input with the generated toolpaths in a teleoperation setup.

Grasp Generation Recent advancement in computer vision enable automatic grasp plan generation based on the RBG-D information. Mousavian *et al.* proposed GraspNet [50], In this work, a grasp generation system based on a variational auto-encoder and grasp evaluator model achieves an 88% success rate on diverse objects using 3D point clouds from a depth camera, trained purely in simulation and working directly in the real world. A follow-up work [51], proposed by Fang *et al.*, established a large-scale grasp pose detection dataset with more than 97,280 RGB-D images of

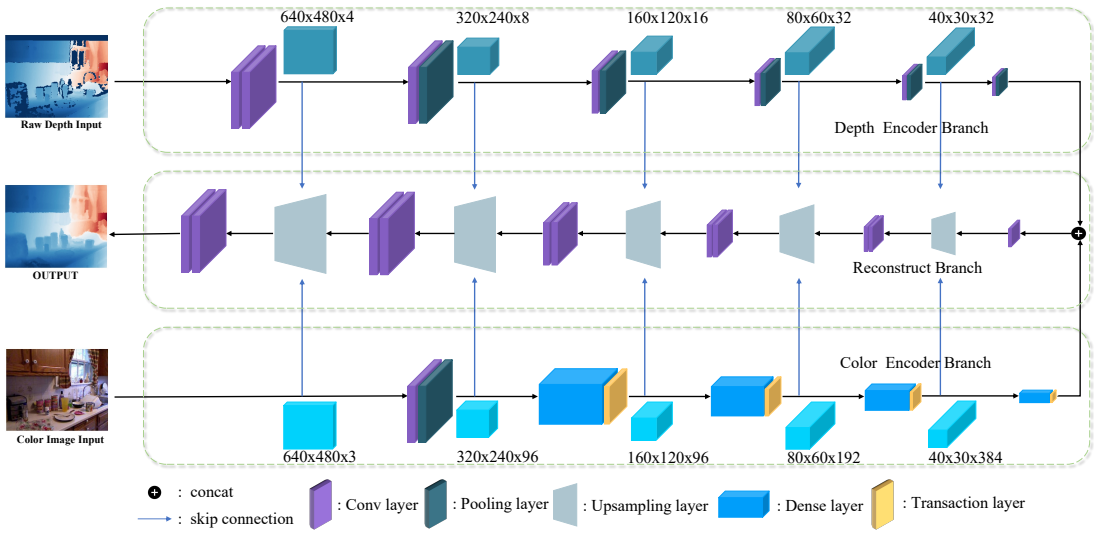

Fig. 2 An overview of the proposed framework. The network is based on an encoder-decoder framework with a multi-scale feature fusion strategy. From left to right, there are two branches, respectively, for the progressive feature extraction of the color map and the depth image with multiple-level receptive fields. In the Reconstruct branch, the features of the color map and the previous hierarchical features continue to guide the restoration of the depth image.

objects for training and testing automatic grasp algorithms. They also proposed and implemented a grasp generation algorithm as a benchmark for others to compare with.

Human-robot Interaction The depth information enables new affordances for more intuitive and efficient interaction between human operators and collaborative robots. The gesture-based interface has been developed by Magrini *et al.* [5] to enable the control of collaborative robots, which is based on the RGB-D information of the operator's hand. Several AR-based interfaces [4, 16–18] are also proposed to provide intuitive robot programming methods with AR visualization. The depth of information is critical to providing a correct visual perception since successful handling of occlusion is achieved by comparing the depth of the real surroundings with the rendered virtual model.

3 Proposed Method

We propose a consistent network for both depth image completion and super-resolution (SR). As illustrated in Fig. 2, our proposed model takes a raw (incomplete or low-resolution) depth image and its corresponding RGB image as input and outputs a complete or high-resolution depth image. It mainly consists of three branches: the color encoder branch, the depth encoder branch, and the reconstruct branch. The color encoder branch utilizes several dense backbone blocks and a residual learning strategy [52] to fully extract textural and structural features from an RGB image. The depth encoder branch employs a custom CNN network to extract information from the raw depth image. To fuse the representations between the two encoder branches, the reconstruct branch introduces a fusion strategy to merge and restore the features from different levels in the output of former encoder branches.

3.1 Network Architecture

Color Encoder Branch To fully extract the hidden features of the RGB image, we use several convolution layers and dense layers to mine hierarchical features. As shown in Fig. 2 (Down), the color encoder branch of our approach consists of six blocks. These blocks are utilized to generate feature maps with dimensions of 1/1, 1/2, 1/4, 1/8, 1/16, and 1/32 of the original image size. We firstly employ two 3×3 convolutions and a max-pooling layer with stride 2 to downsample the original RGB image. Then we use several dense layers to mine hierarchical features. Following each dense layer, we utilize the transition layers, which increase sensitivity of the network to lower-level features. The operations in the color

encoder branch can be described as follows:

$$\begin{aligned} I_{conv}^{1,I} &= \sigma\left(\mathbf{W}^{1,D} * I^{1,I} + \mathbf{b}^{1,D}\right) \\ F_{down}^{1,I} &= maxpool(I_{conv}^{1,I}) \\ F_{down}^{i+1,I} &= transition(Dense(F_{down}^{i,I})) \end{aligned} \tag{1}$$

where $I^{1,I}$ is the RGB image, $i \in \{1,2,3,4\}$ represents the i-th layer, $\mathbf{W}^{1,D}$ and $\mathbf{b}^{1,D}$ are weight and bias in the first Conv layer, $*$ represents the convolution operation, and σ denotes the element-wise activation function, which adopts the rectified linear unit (ReLU). $F_{down}^{1,I}$ is the first output in the color encoder branch. *Dense* and *transition* denote dense blocks and transition layers in DenseNet121 [52]. The dense block enables full feature extraction across various scales and employs a feature reuse mechanism that reduces parameters of the network. This, in turn, enhances the guidance of the subsequent reconstruction.

Depth Encoder Branch The depth encoder branch follows a similar structure to our color encoder branch, with the distinction that we substitute the dense layers with conventional convolution layers. This modification is due to the fact that depth images contain less channel information compared to RGB images. Employing a complex feature extraction mechanism in this scenario may result in network overfitting. As illustrated in Fig. 2 (Top), we utilize two traditional convolution layers with a 3×3 convolution kernel to produce the input feature map. Then, we apply several down-sampling modules to extract multi-level features, which can be mathematically represented as follows:

$$\begin{aligned} F_{conv}^{i+1,D} &= \sigma\left(\mathbf{W}^{i,D} * F_{down}^{i,D} + \mathbf{b}^{1,D}\right) \\ F_{down}^{i+1,D} &= convpool(F_{conv}^{i,D}) \end{aligned} \tag{2}$$

where $i \in \{1,2,3,4,5\}$, and *convpool* represents the Conv and Pooling layers.

Reconstruct Branch We design the reconstruct branch to progressively refine the raw depth image with the hierarchical features generated from the other two branches. As shown in Fig. 3 (Middle), the highest-level input is obtained by directly concatenating the feature maps of the last layer from both the color and depth branches. In each step of the subsequent reconstruction, we further integrate the features between different branches through the fusion module to learn the consistency of features in different domains. The fusion modules can be expressed as:

$$\begin{aligned} F_f^{i+1,R} &= [F_{up}^{i,R}, F_{down}^{m,I}, F_{down}^{n,D}] \\ F_{conv}^{i+1,R} &= \sigma\left(\mathbf{W}_f^{i+1,R} * F_f^{i+1,R} + \mathbf{b}_f^{i+1,R}\right) \\ F_{up}^{i+1,R} &= \sigma\left(\mathbf{W}^{i+1,R} * F_{conv}^{i+1,R} + \mathbf{b}^{i+1,R}\right) \end{aligned} \tag{3}$$

where $i \in \{1,2,3,4\}$, $m = k - i - 2$ and $n = k - i - 1$. $k = 7$ represents the maximum number of modules in the three branches. The $F_{down}^{m,D}$, $F_{down}^{n,C}$ are obtained from the color and depth encoder branches. $F_{up}^{i,R}$ denotes the features in the previous step of the reconstruct branch. $\mathbf{W}_f^{i+1,R}$ and $\mathbf{b}_f^{i+1,R}$ are the convolution parameters corresponding to $F_f^{i+1,R}$.

3.2 Loss Function

To optimize the network parameters for the specific tasks of depth completion and super-resolution, we employ separate loss functions for each task. These loss functions capture different aspects of the desired output and guide the network to generate high-quality depth images and super-resolved images.

Depth Completion To facilitate our depth completion task, we define three types of loss functions: $L1$ loss, *grad* loss, and *structure* loss.

$L1$ **Loss** $L1$ loss is an element-wise loss, which can be defined as:

$$L_{\text{point}} = \frac{1}{W * H} \sum_{i=1}^{W} \sum_{j=1}^{H} \left\| I_{i,j}^{pred} - I_{i,j}^{gt} \right\|_1 \tag{4}$$

where L_{point} denotes the $L1$ loss, and W, H denotes the width and height of the predicted depth image. $I_{i,j}^{pred}, I_{i,j}^{gt}$ denote the value of pixels in the predicted depth image and ground truth.

Gradient Loss The gradient is close to zero in the area of continuous depth. In contrast, the gradient is often very large in areas of discontinuous depth. This phenomenon indicates that the edge of the depth image is strongly related to the gradient. As a result, we define the gradient loss as the mean absolute error between the predicted depth image and the ground truth. When we get the predicted depth image, the gradient loss is defined as:

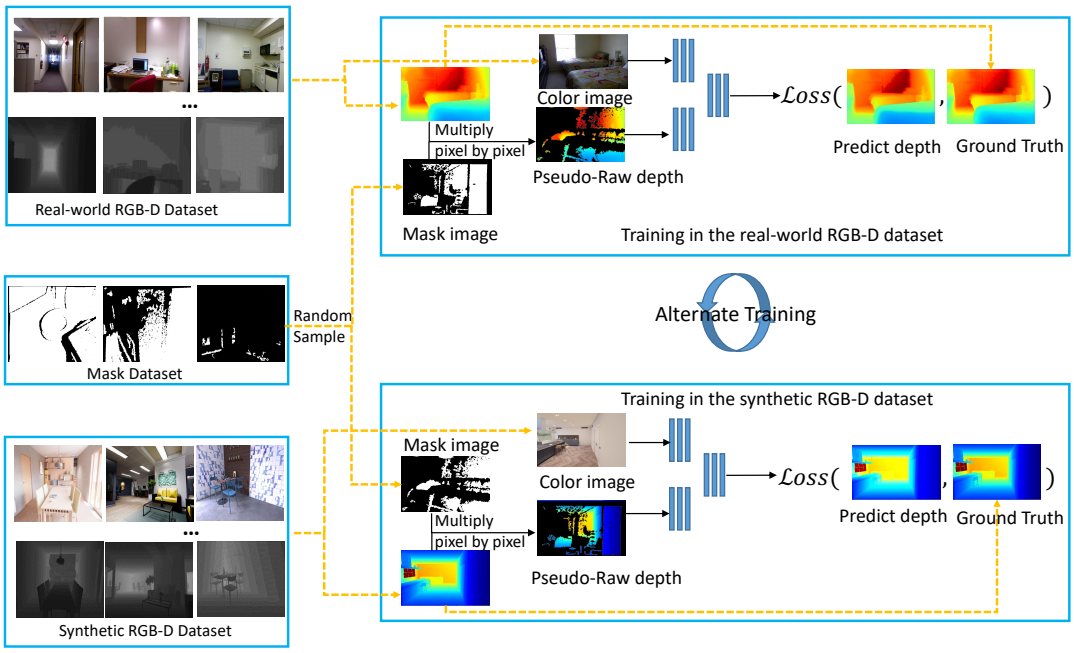

Fig. 3 An example of our training dataset based on the mask dataset, real-world, and synthetic RGB-D datasets. By combining mask images, real-world RGB-D images, and synthetic RGB-D images randomly sampled from the datasets, we can achieve alternate training in the real-world and synthetic RGB-D datasets.

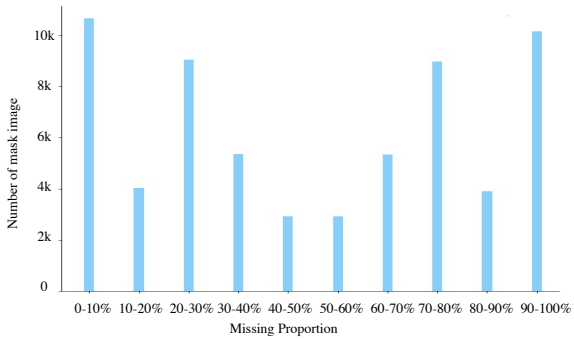

Fig. 4 The distribution of missing proportions for the mask dataset.

$$L_{\text{grad}} = \frac{1}{W*H}\sum_{i=1}^{W}\sum_{j=1}^{H} \left\|\nabla_x\left(I_{i,j}^{pred}\right) - \nabla_x\left(I_{i,j}^{gt}\right)\right\|_1 + \frac{1}{W*H}\sum_{i=1}^{W}\sum_{j=1}^{H}\left\|\nabla_y\left(I_{i,j}^{pred}\right) - \nabla_y\left(I_{i,j}^{gt}\right)\right\|_1 \tag{5}$$

where L_{grad} represents the gradient loss, and W, H denote the width and height of the predicted depth image. $I_{i,j}^{pred}, I_{i,j}^{gt}$ denote the predicted depth image and ground truth. ∇_x, ∇_y denote the vertical and horizontal gradient.

Structure Loss Even with the $L1$ loss and *Gradient* loss, the predicted depth value may still be inconsistent in a certain area, which makes the reconstructed point cloud result worse. Thus, we adopt the structure loss [53, 54] to favor some local consistency, which can be expressed as:

$$L_{ssim} = \frac{1}{W*H}\sum_{i=1}^{W}\sum_{j=1}^{H}\text{SSIM}\left(I_{i,j}^{pred}, I_{i,j}^{gt}, K\right) \tag{6}$$

where L_{ssim} represents the structure loss. k is the sliding window size used to calculate the SSIM, and we set $k = 11$ in our implementation.

The overall loss function is defined as:

$$L_{\text{total}} = \lambda_1 L_1 + \lambda_2 L_{\text{grad}} + \lambda_3 L_{ssim}. \tag{7}$$

Using empirical rules, we find that both the *structure* loss and $L1$ loss are too strong. To address this issue, we conduct several experiments to determine the weights and ultimately set the values of λ_1, λ_2, and λ_3 to 0.1, 1.0, and 0.01, respectively. This helps balance the three losses and ensures optimal performance across all experiments.

Super Resolution We define three types of losses for optimizing the generated super-resolution depth image: $L1$ loss, *Edge* loss, and *Structure* loss.

Edge Loss We define *Edge* loss on the depth edge to obtain the boundaries with more details. It is an element-wise edge loss based on gradient, and can be defined as follows:

$$L_{edge} = \frac{1}{N} \left\| sobel(\mathcal{F}\left(\mathbf{D}^{LR}, \mathbf{I}^{HR}\right)) - sobel(\mathbf{D}^{HR}) \right\|_1 \tag{8}$$

where L_{edge} represents the edge loss, D^{LR} is the low resolution depth, D^{HR} is the high resolution depth, I^{HR} is the high resolution RGB image, and $F(D^{HR}, I^{HR})$ represent the predict high resolution depth, and *sobel* is a boundary operator in the image processing field. It should be noted that this operator requires a hyper-parameter k, namely the size of the sliding window. In our experiments, $k = 5$.

The final loss is a weighted combination of the three losses above, as follows:

$$L_{\text{total}} = \lambda_1 L_1 + \lambda_2 L_{\text{edge}} + \lambda_3 L_{\text{ssim}}. \tag{9}$$

In this work, we utilize the same approach as before to determine the weight of each loss. $\lambda_1 = 0.1$, $\lambda_1 = 1$, and $\lambda_3 = 1$ are set to balance the losses for all the experiments.

4 Experiments

4.1 Datasets

We conduct experiments on several datasets.

- NYU-Depth Raw [55] is one of the largest and most diverse indoor RGB-D datasets, containing over 500 indoor scenes and a total of over 500,000 pairs of RGB-D images with the resolution of 640×480 pixels.
- NYUv2 [56] is captured by both the RGB and depth cameras from Kinect for indoor scenes. It has 1449 RGB-D images in 3 cities and 464 scenes with the resolution of 640×480 pixels.
- ScanNet [57] is also captured from Kinect for indoor scenes. It has 2.5 million RGB-D images in 1,513 scans acquired in 707 distinct spaces with the resolution of 640×480 pixels.
- IRS [58] is a large-scale synthetic but naturalistic indoor robotics stereo (IRS) dataset. It contains more than 100,000 RGB-D images and high-quality surface normal maps with over 100K stereo with the resolution of 960×540 pixels.
- Matterport3D [59] is an RGB-D dataset containing 10,800 panoramic views from 194,400 RGB-D images of 90 building-scale scenes with the resolution of 1280×1024 pixels.
- Middlebury [60–62] consists of high-resolution stereo sequences with complex geometry and pixel-accurate ground-truth disparity data. The image resolution in this dataset is 640×480 pixels.
- MPI Sintel [63] has 1064 synthesized stereo images and ground truth data for the disparity. The image resolution in this dataset is 1024×436 pixels.

4.2 Depth Completion Experiments

Hybrid Training Strategy for Real-Synthetic Data The primary hurdle faced by deep learning-based algorithms for depth completion is the limited availability of high-quality datasets. There are only two types of datasets currently employed for this task: real-world scene datasets and synthetic datasets. However, the real-world scene datasets are incomplete in terms of depth images, and this leads to a significant portion of RGB-D datasets obtained from large-scale 3D sensors with missing depth information being underutilized by deep learning-based methods. It has been shown that using a combination of synthetic and real-world datasets for training models can improve scene understanding and produce superior results, as illustrated in [64]. To overcome this challenge, we propose a hybrid training strategy that combines real and synthetic data, as illustrated in Fig. 3.

We collect 100,000 missing mask samples from a real-world scene dataset [56, 59]. To construct a mask dataset with relatively balanced missing values, we use various data augmentation methods, including 0–1 reversal, horizontal flip, vertical flip, and rotation by 90 and 270 degrees. The resulting mask dataset, shown in Fig. 4, covers a wide range of missing proportions from 0-100%. All mask images are resized to 480×640 pixels. As a result, we obtain a mask dataset with over 600,000 samples exhibiting diverse missing patterns. To create the training data, we utilize 70000 pairs of

RGB-D images from NYU-Depth Raw [55], 40000 pairs from the ScanNet [57] as real-world RGB-D datasets, and 50000 pairs from the IRS [58] as synthetic RGB-D datasets. We randomly select masks and depth images and perform pixel-level multiplication to get incomplete depth images. Notably, in order to make full use of the large-scale real-world RGB-D dataset, we train our proposed network alternately using real-world and synthetic RGB-D datasets with a 1:1 ratio.

Implementation We train our network using the proposed hybrid Training Strategy for Real-Synthetic Data. To optimize the network parameters, we utilize the Adam optimizer [65] and set the learning rate to $1e^{-4}$ for the entire training process. The training is performed using PyTorch on a computer with an i7-9700k CPU, 16 GB of RAM, and a GTX 1080Ti GPU.

Evaluation Metrics We utilize three evaluation metrics in our study, which include Mean Relative Error (REL), Root Mean Squared Error (RMSE), and Threshold Accuracy (δ_i). These metrics are defined as follows:

Mean Relative Error(REL):

$$\frac{1}{n}\sum_{p}^{n}\frac{|y_p - \hat{y}_p|}{y_p} \tag{10}$$

Root Mean Squared Error(RMSE):

$$\sqrt{\frac{1}{n}\sum_{p}^{n}\|y_p - \hat{y}_p\|^2} \tag{11}$$

Threshold Accuracy(δ_i):

$$\max\left(\frac{y_p}{\hat{y}_p}, \frac{\hat{y}_p}{y_p}\right) = \delta < thr \text{ for } thr = 1.05, 1.10, 1.25, 1.25^2, 1.25^3; \tag{12}$$

where y_p and $\hat{y}_p$ are respectively the ground truth and the prediction of the depth image, and n is the total number of pixels for each depth image.

Evaluation on NYUv2 Following the work of Senushkin *et al.* [25], we cut off black borders (45, 15, 45, 40 pixels from the top, bottom, left, and right sides) from the RGB-D images and resize them into 320×240 pixels for a fair comparison. As illustrated in Table 1, our method outperforms all other methods across the metrics, especially on RMSE and REL. We also test our proposed hybrid training strategy for Real-Synthetic Data on the work of Senushkin *et al.* [25] labeled with (our) in Table 1. Moreover, we visualize some results in Fig. 5 to demonstrate the effectiveness of our method in generating accurate depth predictions. As we can see, our model produces sharper object boundaries and achieves higher accuracy near the edges of objects.

Evaluation on the Matterport3D We train our network with our proposed training strategy and test it on the Matterport3D test dataset to make a fair comparison with other methods. To account for any potential impact of using different training datasets, we also train the work proposed by Senushkin *et al.* [25], which is labeled (our). As shown in Table 2, our proposed method achieves superior performance compared to state-of-the-art (SOTA) methods in terms of the root mean squared error (RMSE), which is the primary metric for evaluating the accuracy of depth completion. While our method may not always achieve the best performance in other evaluation metrics besides RMSE, we think that it may come from the incomplete problem of Matterport3D. As depicted by Zhang et al. [20], only 64.6% of the pixels missing from the raw depth images in Matterport3D are filled. Another reason is that we use different datasets to train our models, which may lead to some domain problems. According to Table 2, the model trained with our dataset does not outperform the model trained with Matterport3D.

Evaluation for Sparse Depth Completion The difference between LiDAR-based outside depth completion and our indoor scenario depth completion is that the former uses a sparse depth image as input, while the latter uses a semi-dense depth image. To obtain a sparse depth image, we randomly chose 500 points from each raw semi-depth image on the NYUv2 dataset. We then test our model over the dataset to verify its generalization ability for sparse depth completion. Notably, we compare our method with both indoor-based and outdoor-based methods.

Quantitative results are shown in Table 3. Compared to the indoor-based method [25], we get much better results in terms of all metrics.

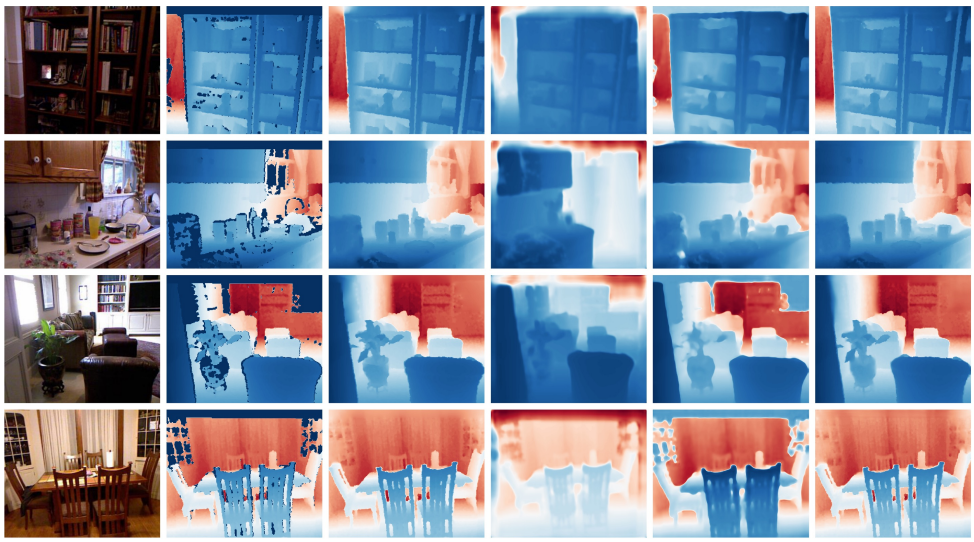

Fig. 5 The visualization result of the NYUv2 Test dataset. From the visualization result, our result has more accuracy than others and the boundary is clearer than others. GT refers to the ground truth depth image, while Sensor refers to the incomplete depth image obtained from RGB-D cameras.

Table 1 NYUv2 Test. We use the results for Senushkin *et al.* [25] and Huang *et al.* [24] reported in [25]. Methods labeled (our) are trained by our proposed training dataset. In the top two rows, the models are trained with the Matterport3D dataset, while in the bottom two rows, the models are trained using our proposed training strategy.

Methods	$RMSE \downarrow$	$REL \downarrow$	$\delta_{1.25^1} \uparrow$	$\delta_{1.25^2} \uparrow$	$\delta_{1.25^3} \uparrow$
Huang *et al.* [24]	0.271	0.016	0.981	0.991	0.994
Senushkin *et al.* [25]	0.205	0.016	0.988	0.996	0.999
Senushkin *et al.* [25] (our)	0.847	0.009	0.833	0.858	0.888
Our Work	**0.106**	**0.002**	**0.993**	**0.999**	**1.000**

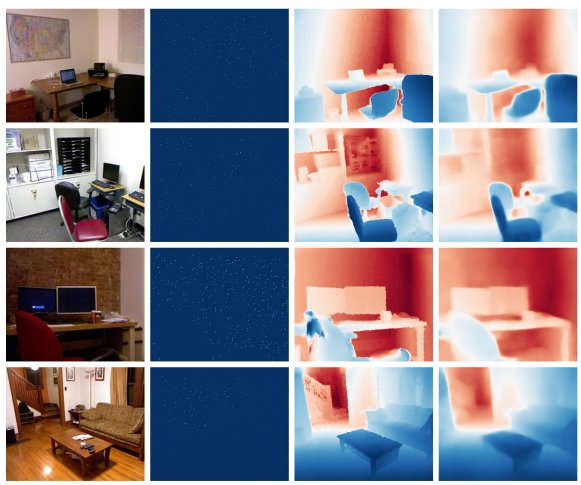

Fig. 6 The visualization result of the NYUv2 Test dataset for sparse depth completion. From the visualization result, we can see that our model can effectively generate a completed depth image.

For outdoor-based methods [23, 66, 68, 69], which trained on the outdoor RGB-D datasets, we may not be able to reach state-of-the-art performance, but we can still outperform some of them [23, 66]. The results of our method are visualized in Fig. 6. It can be observed that our model generates complete depth images with clear boundaries for most parts of the input sparse depth images.

Comparison with the Former Adaptive Convolution Method Table 4 clearly shows that our new network is capable of producing depth images with higher accuracy and efficiency. The results demonstrate that our network is better at fusing color and depth image features than the approach using adaptive convolution [70]. Moreover, with densenet as the backbone, our network can extract features from the color branch more quickly.

Table 2 Matterport3D Test. We use the results for Senushkin *et al.* [25], Zhang *et al.* [20] and Huang *et al.* [24] reported in [25]. Methods labeled "our" are trained using our proposed training dataset. The top three rows are the methods trained with the Matterport3D dataset. The bottom two rows represent models trained using our proposed training strategy.

Methods	$RMSE \downarrow$	$\delta_{1.05} \uparrow$	$\delta_{1.10} \uparrow$	$\delta_{1.25^1} \uparrow$	$\delta_{1.25^2} \uparrow$	$\delta_{1.25^3} \uparrow$
Zhang *et al.* [20]	1.316	0.657	0.708	0.781	0.851	0.888
Huang *et al.* [24]	1.092	0.661	0.750	0.850	0.911	0.936
Senushkin *et al.* [25]	0.961	**0.726**	**0.813**	**0.890**	**0.933**	**0.949**
Senushkin *et al.* [25] (our)	0.970	0.615	0.701	0.802	0.871	0.904
Our Work	**0.933**	0.663	0.726	0.799	0.866	0.902

Table 3 NYUv2 Test for sparse depth completion in the outdoor scenario. We use the results that are reported in their paper. Each raw image is generated by a random-sampling strategy for 500 points.

Methods	$RMSE \downarrow$	$REL \downarrow$	$\delta_{1.25^1} \uparrow$	$\delta_{1.25^2} \uparrow$	$\delta_{1.25^3} \uparrow$
Gansbeke *et al.* [66]	0.344	0.042	0.961	0.985	0.995
Li *et al.* [23]	0.272	0.034	0.973	0.992	0.997
Senushkin *et al.* [25]	0.263	0.035	0.975	0.993	0.998
Our work	0.218	0.028	0.979	0.995	0.999
CSPN [67]	0.117	0.016	0.992	0.999	1.000
Park *et al.* [68]	0.092	**0.012**	**0.996**	**0.999**	**1.000**
Huynh *et al.* [69]	**0.090**	0.014	**0.996**	0.999	1.000

4.3 Super Resolution Experiments

Implementation We conduct our experiments on four datasets: NYU-Depth Raw [55], NYUv2 [56], Middlebury [60–62] and MPI Sintel [63]. The experiments on the above three datasets are conducted in two types. In the first type, we train the model on NYU-Depth Raw with batch size 8 for 5 epochs and test it on NYUv2. In the second type, we select 34 RGB-D images from the Middlebury and 58 RGB-D images from the MPI Sintel depth following the work in [34]. We utilize 82 images for training and 10 images for validation. In order to generate low-level inputs for super-resolution tasks, we begin by cropping the high-resolution (HR) depth image to a size of 128×128 pixels. We then employ nearest neighbor sampling to downsample the image and subsequently utilize Bicubic interpolation to upsample it back to its original size. The downsampling is carried out at sampling rates of 2, 4, 8, and 16, respectively. This is a standard approach since the Middlebury dataset has one type of image and related depth images.

4.3.1 Evaluations

Evaluation metrics. We use two evaluation metrics, which are Root Mean Squared Error (RMSE) and Peak Signal-to-Noise Ratio (PSNR). The Root Mean Squared Error has been introduced in the Deep Completion experiment.

Peak Signal-to-Noise Ratio (PSNR):

$$PSNR = 10 \log_{10} \left(\frac{MAX^2}{MSE} \right) \tag{13}$$

where MAX is the maximum of the image color and MSE is the Mean Square Error for the input image and output image.

Evaluation on NYUv2 To evaluate our method, we train GF [28], TGV [71], RDN [72], DepthSR [44], and SRFBN [73] on NYUv2 using the codes provided by their authors. For MSG [34], DU-DEAL [45] and GbFT [74], we directly use the released models to generate a super-resolution depth image. As shown in Table 5, the average RMSE/PSNR of our results are better than most current SOTA methods. Our model has the best performance for the scenarios with downsampling rates of 2, 4, and 16. We also compare

Table 4 Comparison with the former study of adaptive convolution. We compare the results of our network with those of the former study using adaptive convolution. Max Err and Min Err refer to the maximum and minimum difference between the predicted and ground truth depth values.

Methods	Average Err(m)	Max Err(m)	Min Err(m)	Avg. Time (second)
Zhang *et al.* [20]	0.170	0.329	0.085	13.450
Xian *et al.* [70]	0.119	0.261	0.037	0.045
Our work	0.107	0.440	0.025	0.021

Table 5 Quantitative comparisons of four scales on the NYUv2 Test Dataset in terms of average RMSE/PSNR values. The lower the RMSE or the higher the PSNR, the better the performance.

Method	**Average RMSE ↓**				**Average PSNR ↑**			
	2x	**4x**	**8x**	**16x**	**2x**	**4x**	**8x**	**16x**
Bibcubic	4.20	4.38	6.11	7.38	39.03	36.61	33.86	31.37
GF [28]	5.41	6.07	12.64	17.18	38.03	36.23	32.31	29.25
TGV [71]	3.20	5.18	10.11	18.09	40.05	35.91	32.17	28.17
RDN [72]	4.83	5.62	7.58	-	36.52	35.10	32.42	-
SRFBN [73]	2.91	3.79	10.82	-	41.03	38.61	35.16	-
DU-REAL [45]	3.08	4.47	7.19	10.32	45.47	40.71	35.82	31.10
DepthSR [44]	4.23	5.20	5.53	7.90	40.34	37.85	37.40	34.05
GbFT [74]	-	2.14	3.86	5.86	-	41.94	36.73	33.16
Xian [70]	-	**1.56**	8.32	-	-	**51.32**	33.13	-
Ours	**2.10**	3.57	**3.83**	**5.83**	**47.35**	41.06	**38.82**	**34.82**

Table 6 The average running time (seconds/frame) with different scales on the NYUv2 Test datasets.

Method	**2x**	**4x**	**8x**	**16x**
Bibcubic	0.01	0.01	0.01	0.01
GF [28]	2.75	2.89	2.81	3.21
TGV [71]	29.57	29.42	29.28	36.2
RDN [72]	3.46	2.35	2.29	\
SRFBN [73]	0.50	0.23	0.24	\
MSG [34]	0.30	0.32	0.38	0.44
DU-REAL [45]	0.24	0.24	0.22	0.22
DepthSR [44]	1.84	1.85	1.86	1.85
Ours	**0.18**	**0.17**	**0.15**	**0.16**

with our previous work [70]. This method outperforms it for input depth images with low densities. Considering that capturing depth images in real-world circumstances usually has a low density, our proposed method is more suitable and practical for real-world applications (e.g., AR-based HRI applications). Furthermore, we analysis the performance of our method visually. As shown in Fig. 7, our method not only keeps the boundary details correct when up-sampling the depth but also makes the restored depth more consistent and reasonable. Additionally, our proposed model can also recover super-resolution depth at a higher sampling ratio. That is, the details of the super-resolution depth image can still be retained under the 8x ratio, which is shown in Fig. 8. On the NYUv2, Bicubic, GF [28], and TGV [71] generate results with some artifacts or noises. RDN [72] and SRFBN [73] can't generate particularly good details when restoring depth images. DU-REAL [45], GbFT [74] and DepthSR [44] produce competitive results, which generate more details than other work [28, 71–73]. In contrast, our method generates the depth boundaries with more details.

Evaluation on Middlebury The quantitative results are shown in Table 7. In general, our proposed method exhibits lower RMSE values compared to other methods when evaluated on six test datasets (namely, Art, Books, Dolls, Laundry, Moebius, and Reindeer), which indicates that our method is better than other depth image super-resolution methods, especially for 4x and 8x cases.

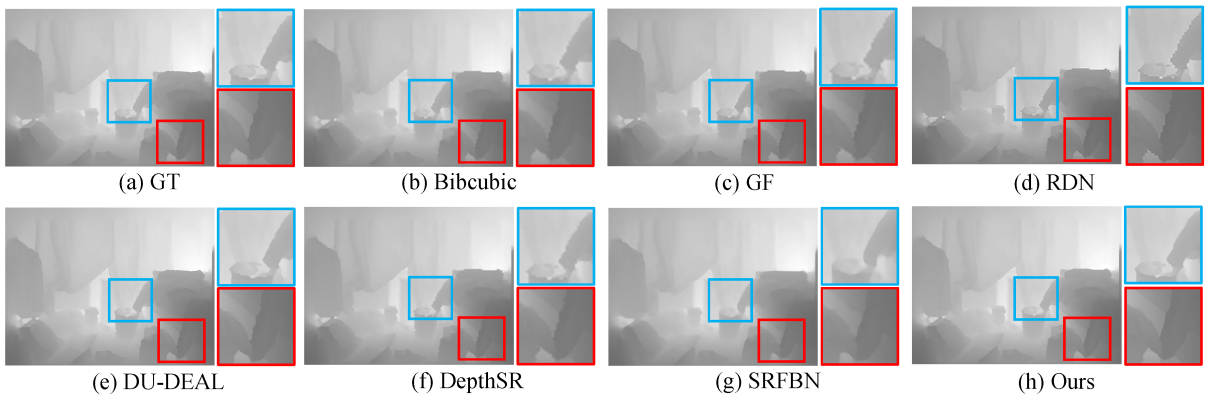

Fig. 7 Visual Depth Super Resolution comparison results for 4× on NYUv2 Test datasets.

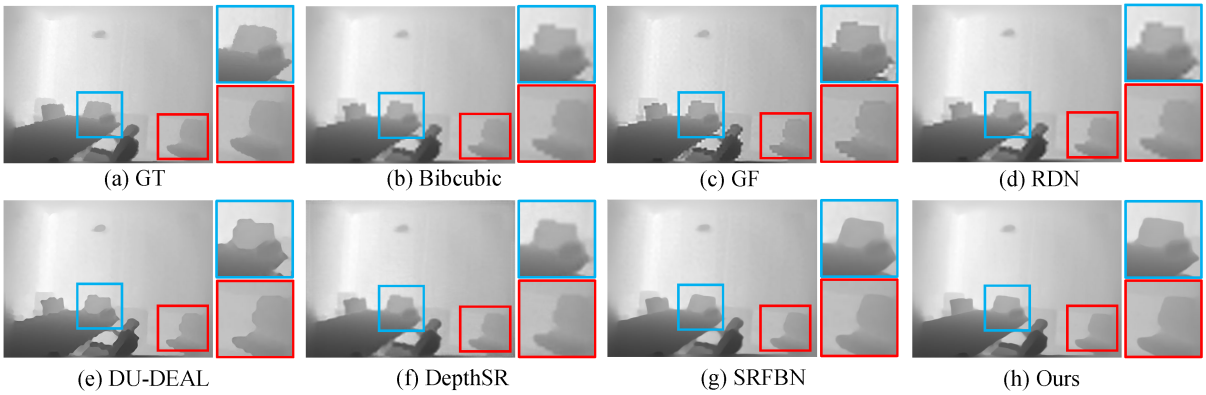

Fig. 8 Visual Depth Super Resolution comparison results for 8× on NYUv2 Test datasets. From (a) to (h), they are the upsampling results of Bibubic, GF [28], RDN [72], DU-REAL [45], DepthSR [44], SRFBN [73], and Ours. The details are drawn inside the box.

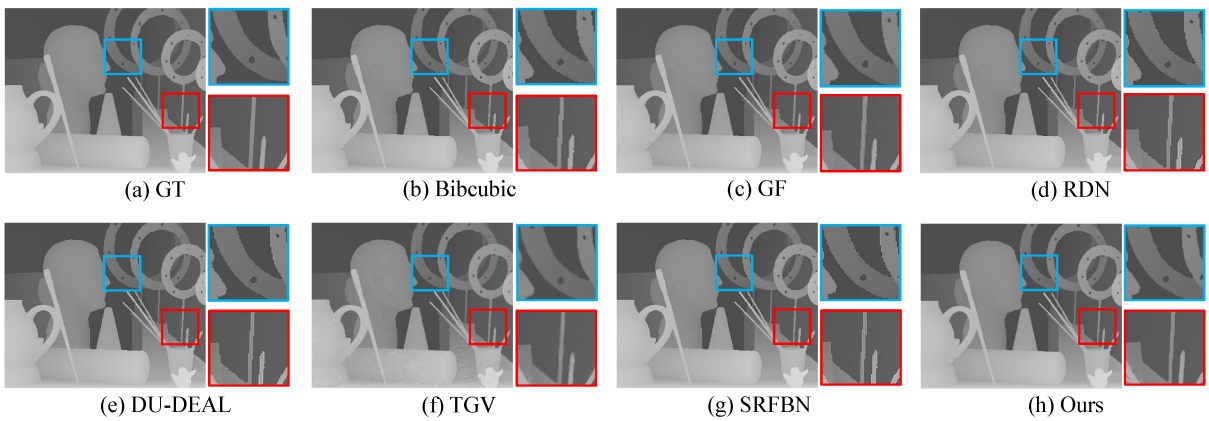

Fig. 9 Visual Depth Super Resolution comparison results for 4× on Middlebury datasets. From (a) to (h), they are the high-resolution depth images and the results of Bicubic, GF [28], RDN [72], DU-REAL [45], TGV [71], SRFBN [73], and Ours.

Table 7 Qualitative analysis results on four scales in terms of RMSE (↓) values.

Method	Art				Books				Dolls			
	2x	**4x**	**8x**	**16x**	**2x**	**4x**	**8x**	**16x**	**2x**	**4x**	**8x**	**16x**
Bibcubic	3.53	3.84	4.47	5.72	1.31	1.61	2.34	3.34	3.28	3.34	3.47	3.72
GF [28]	2.75	3.91	5.32	8.36	1.36	1.76	2.10	3.36	1.23	2.48	3.97	4.86
TGV [71]	3.03	3.78	7.08	11.59	1.29	1.61	2.15	3.05	1.63	1.96	2.62	4.08
RDN [72]	2.61	3.82	5.87	-	1.46	2.01	3.08	-	1.25	1.70	2.22	-
SRFBN [73]	1.99	3.02	3.58	-	0.54	1.22	1.51	-	1.04	1.81	2.06	-
MSG [34]	0.66	1.47	2.45	4.57	0.37	0.67	1.03	1.60	0.34	0.69	1.05	1.60
DU-REAL [45]	0.62	1.15	2.15	4.32	0.34	0.57	1.01	1.54	0.31	0.65	0.98	1.42
DepthSR [44]	0.53	1.20	2.22	3.91	0.31	0.60	0.89	1.51	0.32	0.62	0.85	1.48
Ours	**0.51**	**1.05**	**2.08**	**3.87**	**0.30**	**0.55**	**0.83**	**1.47**	**0.29**	**0.58**	**0.80**	**1.35**

Method	Laundry				Moebius				Reindeer			
	2x	**4x**	**8x**	**16x**	**2x**	**4x**	**8x**	**16x**	**2x**	**4x**	**8x**	**16x**
Bibcubic	3.35	3.49	3.77	4.35	3.28	3.36	3.50	3.81	3.40	3.52	3.83	5.82
GF [28]	2.26	2.67	3.84	5.23	1.92	2.29	3.85	5.22	2.27	2.89	3.98	5.85
TGV [71]	2.15	2.51	3.82	6.41	1.21	1.65	2.13	2.73	2.41	2.71	3.79	7.27
RDN [72]	2.53	3.22	4.65	-	1.22	1.61	2.39	-	3.32	2.93	4.41	-
SRFBN [73]	1.67	2.13	2.28	-	1.12	1.43	1.52	-	1.63	2.07	2.15	-
MSG [34]	0.37	0.79	1.51	2.62	0.35	0.66	1.02	1.63	0.42	0.98	1.76	2.91
DU-REAL [45]	0.35	0.76	1.49	2.56	0.34	0.62	0.97	1.54	0.39	0.95	1.61	2.53
DepthSR [44]	0.34	0.78	1.32	2.26	0.32	0.59	0.92	1.51	0.39	0.96	1.57	2.47
Ours	**0.32**	**0.71**	**1.21**	**2.15**	**0.29**	**0.56**	**0.85**	**1.42**	**0.35**	**0.82**	**1.45**	**2.21**

We also show the visual comparison results in Fig. 9. Our method produces better results in the details (in the red box) and boundaries (in the blue box). Compared with traditional methods [28, 71], our method produces less noise as well. Compared with current learning-based methods [34, 44, 70, 72–74], our method can produce more clear and reasonable super-resolution results for depth images.

Running Time To compare the computational efficiency of our method with other methods, we conduct tests on the NYUv2 Test datasets, which has a resolution of 640x480. The implementation codes of Bicubic, GF, RDN, SRFBN, and DepthSR are written in Python (implemented using TensorFlow or PyTorch) with GPU acceleration, while TGV and DU-REAL are implemented with Matlab. MSG is implemented with Caffe. The code of our method is implemented with PyTorch and accelerated by the GPU. We calculate the average running time for the entire dataset, and the results are presented in Table 6. The running time of our method is faster than other methods except bicubic at different scales.

4.4 Human-robot-collaboration Applications in Manufacturing

To evaluate the proposed method in real-life HRI applications, we set up three scenarios for experiments: 1) a remote-controlled robot based on an unmanned ground vehicle (UGV) (see Fig. 10), 2) an AR-based toolpath planning (Fig. 11), and 3) an automatic toolpath extraction (Fig. 12). For the first two scenarios, we use a ZED mini depth camera to capture depth images with 1280×720 resolution at 30 frames per second (FPS), while for the third scenario, we use a Microsoft Kinect depth camera with 640×480 resolution at 30 FPS.

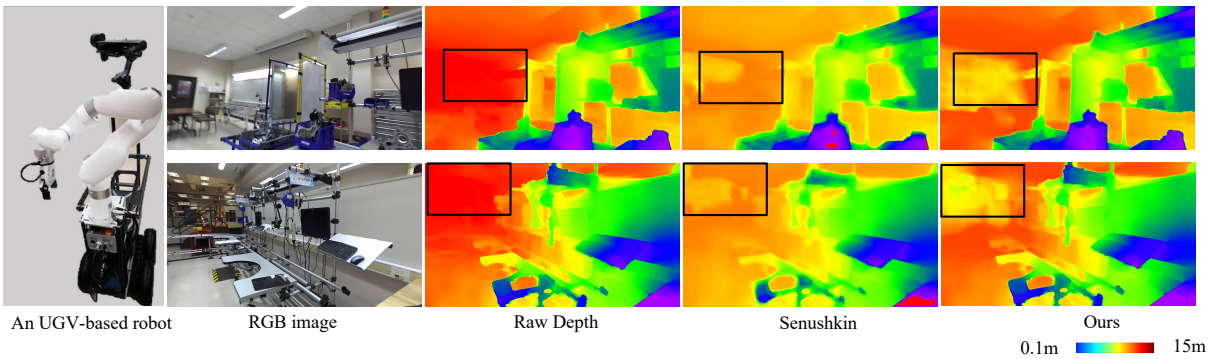

Fig. 10 A remote-controlled robot based on an unmanned ground vehicle (UGV) in a production line environment. Black boxes are used to highlight the areas of depth images that have been fully completed.

We apply the proposed completion and super-resolution methods to the three scenarios without an additional training step. All experiments are conducted in the same setting as the training stages, as indicated in the implementation part of sections 3.2 and 3.3. The details of the experiments are presented as follows.

A Remote-controlled Robot based on A UGV As shown in Fig. 10, a UGV equipped with a robot manipulator and a depth camera is controlled remotely by an operator through networks. This system enables exploration and operation in a production line environment, which heavily relies on depth information for UGV's navigation and accurate robot operations. We apply our methods to the captured raw RGB-D images to fill in the missing depth information. Since the ground truth of the depth is not available, we visualize the results in Fig. 10, where the black areas in the raw depth images represent missing depth regions. Our model is successful in restoring and completing the depth maps. Moreover, our method exhibits superior performance in the black box regions when compare to the work of Senushkin et al. [25], which is regarded as one of the best methods in depth completion.

AR-based Toolpath Planning We apply the proposed method to an AR-based HRI interface. For such an interface, a video see-through-based AR headset is utilized to display the virtual contents in the real environment for toolpath planning [4, 18] or pick-and-place tasks [75]. The general procedure for users of these interfaces consists of three steps (see Fig. 11 (a)): 1) define and edit waypoints for the toolpath; 2) check the simulation of the virtual robot traveling through the planned path; and 3) go back to step 1 if there are collisions during the simulation; otherwise, let the physical robot execute the motion following the planned path. For these steps, it is essential to have a visualization of virtual contents with correct occlusion information (i.e., whether the virtual content is on top of or behind a physical object). The accurate and high-quality depth image of the physical environment is utilized to compute the occlusion with the virtual contents. We implement the same AR rendering and occlusion algorithm [76] based on [4, 18], and test it with the raw depth image before and after our method. As shown in Fig. 11 (b), the missing depth information of the frame structure (the white corner of the frame shown in the left figure of Fig. 11 (b)) affects the occlusion computation and leads to the incorrect rendering of the virtual robot. After being processed by our method, the depth images are completed and refined (see the left figure of Fig. 11 (c)), and therefore the virtual robot is rendered correctly with proper occlusion.

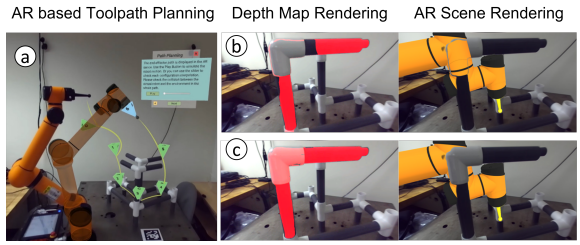

Fig. 11 The AR-based toolpath planning interface allows users to define the toolpath through an AR rendering of the virtual robot traveling along the toolpath (a). The depth image before processing causes an occlusion error in the rendering due to the depth image incompletion and low quality (b), while the AR rendering works properly with the processed depth by our method (c).

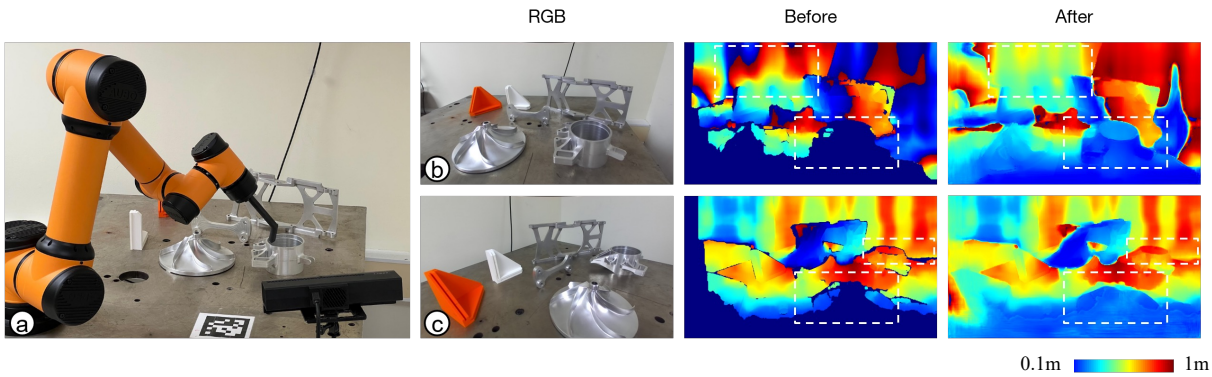

Fig. 12 The automatic toolpath planning algorithm can extract the toolpaths from the captured depth camera, and an experimental setup is shown in (a). The raw depth images with missing depth information (the "Before" column of (b) and (c)) are completed and refined utilizing the high-resolution RGB images (the "RGB" column of (b) and (c)). The resultant depth images show a clear improvement in the completion of the missing depth information and the accuracy of the boundaries (see the "After" column of (b) and (c)).

The high efficiency of our method is an important feature for the real-time performance of AR-based interfaces.

Automatic Toolpath Extraction Another application is the automatic toolpath path extraction for efficient HRI [3]. We follow Ni *et al.*'s work [3] to set up our experiment (see Fig. 12 (a)). The mechanical parts are scanned by a Microsoft Kinect depth camera, and there is missing depth information and inaccurate boundaries between different parts (as shown in the middle of Fig. 12 (b) and (c)). Such depth images will cause difficulty for the toolpath extraction algorithms and, therefore, need to be improved. We apply the proposed completion and super-resolution methods for two depth images utilizing the high-resolution RGB images (see the leftmost of Fig. 12 (b) and (c)). The resultant depth images show a clear improvement in regions with missing depth information and boundaries. Please refer to the right column of Fig. 12 (c) to see the completed and refined boundaries (regions in the white rectangles).

5 Conclusions

In this paper, we propose a multi-scale progressive network for efficient depth image repair and super-resolution with the help of a fusion strategy. Our method consists of three branches: the color feature encoder branch, the depth feature encoder branch, and the reconstruct branch based on the fusion module. The color feature encoder branch uses a multi-scale network based on DenseNet to extract the color feature. The depth feature encoder branch is used to extract the geometric features of the input depth image. Then, the reconstruct branch fuses these multi-scale features at different levels to reconstruct the high-quality depth image. We also propose a joint training strategy combining random masks and 'real-synthetic' data to generate a large number of large-scale RGB-D datasets. Numeric results on public datasets and our dataset demonstrate the effectiveness and efficiency of our method. We also apply the proposed methods to three human-robot interaction applications, including a remote-controlled robot based on a UGV, AR-based toolpath planning, and automatic toolpath extraction. The testing results indicate that our methods can effectively complete and improve the depth images and, therefore, have the potential to benefit downstream HRI applications.

6 Declarations

Funding Chuhua Xian and Jun Zhang are supported by the Natural Science Fund of Guangdong Province under Grant No. 2021A1515011849. Wenhao Yang is partially supported by the National Science Foundation under Grant No. DGE-2125362. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Author Contribution Yunbo Zhang and Chuhua Xian contributed to the study's conception and design. Chuhua Xian and Jun Zhang implemented the proposed method, conducted experiments, collected data, and performed an analysis of the results. Jun Zhang and Wenhao Yang conducted the experiment, collected data, and analyzed the results for human-robot collaboration applications in manufacturing. The first draft of the manuscript was written by Chuhua Xian and Jun Zhang, and all authors commented on previous versions of the manuscript. Yunbo Zhang supervised the whole project and edited the manuscript.

Data Availability Statement The data that support the findings of this study are openly available in NYU-Depth Raw at https://cs.nyu.edu/~silberman/datasets/nyu_depth_v1.html, NYUv2 at https://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html, ScanNet at http://www.scan-net.org/, IRS at https://github.com/HKBU-HPML/IRS, Matterport3D at https://niessner.github.io/Matterport/, Middlebury at https://vision.middlebury.edu/stereo/data/, and MPI Sintel at http://sintel.is.tue.mpg.de/.

References

[1] www.automate.org: What Are Collaborative Robots? https://www.automate.org/a3-content/what-are-collaborative-robots

[2] Nikolaidis, S., Ramakrishnan, R., Gu, K., Shah, J.: Efficient model learning from joint-action demonstrations for human-robot collaborative tasks. In: 2015 10th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pp. 189–196 (2015). https://doi.org/10.1145/2696454.2696455. IEEE

[3] www.fortunebusinessinsights.com: Collaborative Robots Market. https://www.fortunebusinessinsights.com/industry-reports/collaborative-robots-market-101692

[4] Yang, W., Xiao, Q., Zhang, Y.: An augmented-reality based human-robot interface for robotics programming in the complex environment. In: International Manufacturing Science and Engineering Conference, vol. 85079, pp. 002–07003 (2021). https://doi.org/10.1115/MSEC2021-62468. American Society of Mechanical Engineers

[5] Magrini, E., Ferraguti, F., Ronga, A.J., Pini, F., De Luca, A., Leali, F.: Human-robot coexistence and interaction in open industrial cells. Robotics and Computer-Integrated Manufacturing **61**, 101846 (2020). https://doi.org/10.1016/j.rcim.2019.101846

[6] Ni, D., Yew, A., Ong, S., Nee, A.: Haptic and visual augmented reality interface for programming welding robots. Advances in Manufacturing **5**(3), 191–198 (2017). https://doi.org/10.1007/s40436-017-0184-7

[7] Hentout, A., Aouache, M., Maoudj, A., Akli, I.: Human–robot interaction in industrial collaborative robotics: a literature review of the decade 2008–2017. Advanced Robotics **33**(15-16), 764–799 (2019). https://doi.org/10.1080/01691864.2019.1636714

[8] Costanzo, M., De Maria, G., Lettera, G., Natale, C.: A multimodal approach to human safety in collaborative robotic workcells. IEEE Transactions on Automation Science and Engineering **19**(2), 1202–1216 (2021). https://doi.org/10.1109/TASE.2020.3043286

[9] Ragaglia, M., Zanchettin, A.M., Rocco, P.: Trajectory generation algorithm for safe human-robot collaboration based on multiple depth sensor measurements. Mechatronics **55**, 267–281 (2018). https://doi.org/10.1016/j.mechatronics.2017.12.009

[10] Scimmi, L.S., Melchiorre, M., Troise, M., Mauro, S., Pastorelli, S.: A practical and effective layout for a safe human-robot collaborative assembly task. Applied Sciences **11**(4), 1763 (2021). https://doi.org/10.3390/app11041763

[11] Gualtieri, L., Rauch, E., Vidoni, R.: Emerging research fields in safety and

ergonomics in industrial collaborative robotics: A systematic literature review. Robotics and Computer-Integrated Manufacturing **67**, 101998 (2021). https://doi.org/10.1016/j.rcim.2020.101998

[12] Yeamkuan, S., Chamnongthai, K., Pichitwong, W.: A 3d point-of-intention estimation method using multimodal fusion of hand pointing, eye gaze and depth sensing for collaborative robots. IEEE Sensors Journal **22**(3), 2700–2710 (2021). https://doi.org/10.1109/JSEN.2021.3133471

[13] Gómez-Espinosa, A., Rodríguez-Suárez, J.B., Cuan-Urquizo, E., Cabello, J.A.E., Swenson, R.L.: Colored 3d path extraction based on depth-rgb sensor for welding robot trajectory generation. Automation **2**(4), 252–265 (2021). https://doi.org/10.3390/automation2040016

[14] Konam, S., Rosenthal, S., Veloso, M.: Uav and service robot coordination for indoor object search tasks. arXiv preprint arXiv:1709.08831 (2017). https://doi.org/10.48550/arXiv.1709.08831

[15] Avizzano, C.A.: Human-robot interactions in future military operations. Industrial Robot: An International Journal **38**(5) (2011). https://doi.org/10.1108/ir.2011.04938eaa.010

[16] Ong, S., Yew, A., Thanigaivel, N., Nee, A.: Augmented reality-assisted robot programming system for industrial applications. Robotics and Computer-Integrated Manufacturing **61**, 101820 (2020). https://doi.org/10.1016/j.rcim.2019.101820

[17] Pan, Y., Chen, C., Li, D., Zhao, Z., Hong, J.: Augmented reality-based robot teleoperation system using rgb-d imaging and attitude teaching device. Robotics and Computer-Integrated Manufacturing **71**, 102167 (2021). https://doi.org/10.1016/j.rcim.2021.102167

[18] Yang, W., Xiao, Q., Zhang, Y.: HAR^{2}bot: a human-centered augmented reality robot programming method with the awareness of cognitive load. Journal of Intelligent Manufacturing, 1–19 (2023). https://doi.org/10.1007/s10845-023-02096-2

[19] Vodrahalli, K., Bhowmik, A.K.: 3d computer vision based on machine learning with deep neural networks: A review. Journal of the Society for Information Display **25**(11), 676–694 (2017). https://doi.org/10.1002/jsid.617

[20] Zhang, Y., Funkhouser, T.: Deep depth completion of a single rgb-d image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 175–185 (2018). https://doi.org/10.48550/arXiv.1803.09326

[21] Tang, J., Tian, F.-P., Feng, W., Li, J., Tan, P.: Learning guided convolutional network for depth completion. IEEE Transactions on Image Processing **30**, 1116–1129 (2020). https://doi.org/10.1109/TIP.2020.3040528

[22] Xiong, X., Xiong, H., Xian, K., Zhao, C., Cao, Z., Li, X.: Sparse-to-dense depth completion revisited: Sampling strategy and graph construction. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16, pp. 682–699 (2020). https://doi.org/10.1007/978-3-030-58589-1_41. Springer

[23] Li, A., Yuan, Z., Ling, Y., Chi, W., Zhang, C., *et al.*: A multi-scale guided cascade hourglass network for depth completion. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 32–40 (2020)

[24] Huang, Y.-K., Wu, T.-H., Liu, Y.-C., Hsu, W.H.: Indoor depth completion with boundary consistency and self-attention. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, pp. 0–0 (2019)

[25] Senushkin, D., Romanov, M., Belikov, I., Konushin, A., Patakin, N.: Decoder modulation for indoor depth completion. arXiv preprint arXiv:2005.08607 (2020). https://doi.org/10.1109/IROS51168.2021.9636870

[26] Diebel, J., Thrun, S.: An application of markov random fields to range sensing. In:

Advances in Neural Information Processing Systems, pp. 291–298 (2006)

[27] Yang, Q., Yang, R., Davis, J., Nistér, D.: Spatial-depth super resolution for range images. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2007). https://doi.org/10.1109/CVPR.2007.383211. IEEE

[28] He, K., Sun, J., Tang, X.: Guided image filtering. In: European Conference on Computer Vision, pp. 1–14 (2010). Springer

[29] Liu, M.-Y., Tuzel, O., Taguchi, Y.: Joint geodesic upsampling of depth images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 169–176 (2013)

[30] Petschnigg, G., Szeliski, R., Agrawala, M., Cohen, M., Hoppe, H., Toyama, K.: Digital photography with flash and no-flash image pairs. ACM transactions on graphics (TOG) **23**(3), 664–672 (2004). https://doi.org/10.1145/1015706.1015777

[31] Kopf, J., Cohen, M.F., Lischinski, D., Uyttendaele, M.: Joint bilateral upsampling. ACM Transactions on Graphics (ToG) **26**(3), 96 (2007). https://doi.org/10.1145/1276377.1276497

[32] Hornacek, M., Rhemann, C., Gelautz, M., Rother, C.: Depth super resolution by rigid body self-similarity in 3d. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1123–1130 (2013)

[33] Li, Y., Huang, J.-B., Ahuja, N., Yang, M.-H.: Deep joint image filtering. In: European Conference on Computer Vision, pp. 154–169 (2016). https://doi.org/10.1007/978-3-319-46493-0_10. Springer

[34] Hui, T.-W., Loy, C.C., Tang, X.: Depth map super-resolution by deep multi-scale guidance. In: European Conference on Computer Vision, pp. 353–369 (2016). https://doi.org/10.1007/978-3-319-46487-9_22. Springer

[35] Peng, S., Haefner, B., Quéau, Y., Cremers, D.: Depth super-resolution meets uncalibrated photometric stereo. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 2961–2968 (2017). https://doi.org/10.48550/arXiv.1708.00411

[36] Voynov, O., Artemov, A., Egiazarian, V., Notchenko, A., Bobrovskikh, G., Burnaev, E., Zorin, D.: Perceptual deep depth super-resolution. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5653–5663 (2019)

[37] Wang, Z., Chen, J., Hoi, S.C.: Deep learning for image super-resolution: A survey. IEEE transactions on pattern analysis and machine intelligence (2020). https://doi.org/10.1109/TPAMI.2020.2982166

[38] Botach, A., Feldman, Y., Miron, Y., Shapiro, Y., Di Castro, D.: Bidcd–bosch industrial depth completion dataset. arXiv preprint arXiv:2108.04706 (2021). https://doi.org/10.48550/arXiv.2108.04706

[39] Tan, J., Lin, W., Chang, A.X., Savva, M.: Mirror3d: Depth refinement for mirror surfaces. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 15990–15999 (2021). https://doi.org/10.48550/arXiv.2106.06629

[40] Choi, O., Jung, S.-W.: A consensus-driven approach for structure and texture aware depth map upsampling. IEEE transactions on image processing **23**(8), 3321–3335 (2014). https://doi.org/10.1109/TIP.2014.2329766

[41] Xie, J., Feris, R.S., Sun, M.-T.: Edge-guided single depth image super resolution. IEEE Trans. Image Processing **25**(1), 428–438 (2016). https://doi.org/10.1109/TIP.2015.2501749

[42] Li, Y., Min, D., Do, M.N., Lu, J.: Fast guided global interpolation for depth and motion. In: European Conference on Computer Vision, pp. 717–733 (2016). https://doi.org/10.1007/978-3-319-46487-9_44. Springer

[43] Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. IEEE transactions on pattern analysis and machine intelligence **38**(2), 295–307 (2015). https://doi.org/10.1109/TPAMI.2015.2439281

[44] Guo, C., Li, C., Guo, J., Cong, R., Fu, H., Han, P.: Hierarchical features driven residual learning for depth map super-resolution. IEEE Transactions on Image Processing **28**(5), 2545–2557 (2018). https://doi.org/10.1109/TIP.2018.2887029

[45] Wang, Z., Ye, X., Sun, B., Yang, J., Xu, R., Li, H.: Depth upsampling based on deep edge-aware learning. Pattern Recognition **103**, 107274 (2020). https://doi.org/10.1016/j.patcog.2020.107274

[46] Calcagni, M.T., Scoccia, C., Battista, G., Palmieri, G., Palpacelli, M.: Collaborative robot sensorization with 3d depth measurement system for collision avoidance. In: 2022 18th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA), pp. 1–6 (2022). https://doi.org/10.1109/MESA55290.2022.10004475. IEEE

[47] Dumonteil, G., Manfredi, G., Devy, M., Confetti, A., Sidobre, D.: Reactive planning on a collaborative robot for industrial applications. In: 2015 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO), vol. 2, pp. 450–457 (2015). IEEE

[48] Cherubini, A., Navarro-Alarcon, D.: Sensor-based control for collaborative robots: Fundamentals, challenges, and opportunities. Frontiers in Neurorobotics, 113 (2021). https://doi.org/10.3389/fnbot.2020.576846

[49] Zaki, A., AHMED, M., et al.: Trajectory planning of collaborative robotic contact-based applications (2023)

[50] Mousavian, A., Eppner, C., Fox, D.: 6-dof graspnet: Variational grasp generation for object manipulation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2901–2910 (2019)

[51] Fang, H.-S., Wang, C., Gou, M., Lu, C.: Graspnet-1billion: A large-scale benchmark for general object grasping. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11444–11453 (2020)

[52] Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708 (2017). https://doi.org/10.48550/arXiv.1608.06993

[53] Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004). https://doi.org/10.1109/TIP.2003.81986

[54] Zhao, H., Gallo, O., Frosio, I., Kautz, J.: Loss functions for neural networks for image processing. arXiv preprint arXiv:1511.08861 (2015). https://doi.org/10.48550/arXiv.1511.08861

[55] Silberman, N., Fergus, R.: Indoor scene segmentation using a structured light sensor. In: 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), pp. 601–608 (2011). https://doi.org/10.1109/ICCVW.2011.6130298. IEEE

[56] Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: European Conference on Computer Vision, pp. 746–760 (2012). https://doi.org/10.1007/978-3-642-33715-4_54. Springer

[57] Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5828–5839 (2017). https://doi.org/10.48550/arXiv.1702.04405

[58] Wang, Q., Zheng, S., Yan, Q., Deng, F., Zhao,

K., Chu, X.: Irs: A large synthetic indoor robotics stereo dataset for disparity and surface normal estimation. arXiv e-prints, 1912 (2019). https://doi.org/10.1109/ICME51207.2021.9428423

[59] Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3d: Learning from rgb-d data in indoor environments. arXiv preprint arXiv:1709.06158 (2017). https://doi.org/10.48550/arXiv.1709.06158

[60] Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International journal of computer vision **47**(1-3), 7–42 (2002). https://doi.org/10.1023/A:1014573219977

[61] Scharstein, D., Pal, C.: Learning conditional random fields for stereo. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2007). https://doi.org/10.1109/CVPR.2007.383191. IEEE

[62] Scharstein, D., Hirschmüller, H., Kitajima, Y., Krathwohl, G., Nešić, N., Wang, X., Westling, P.: High-resolution stereo datasets with subpixel-accurate ground truth. In: German Conference on Pattern Recognition, pp. 31–42 (2014). https://doi.org/10.1007/978-3-319-11752-2_3. Springer

[63] Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A naturalistic open source movie for optical flow evaluation. In: European Conference on Computer Vision, pp. 611–625 (2012). https://doi.org/10.1007/978-3-642-33783-3_44. Springer

[64] Handa, A., Patraucean, V., Badrinarayanan, V., Stent, S., Cipolla, R.: Understanding real world indoor scenes with synthetic data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4077–4085 (2016)

[65] Diederik, K., Jimmy, B., et al.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 273–297 (2014). https://doi.org/10.48550/arXiv.1412.6980

[66] Van Gansbeke, W., Neven, D., De Brabandere, B., Van Gool, L.: Sparse and noisy lidar completion with rgb guidance and uncertainty. In: 2019 16th International Conference on Machine Vision Applications (MVA), pp. 1–6 (2019). https://doi.org/10.23919/MVA.2019.8757939. IEEE

[67] Cheng, X., Wang, P., Yang, R.: Depth estimation via affinity learned with convolutional spatial propagation network. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 103–119 (2018). https://doi.org/10.48550/arXiv.1808.00150

[68] Park, J., Joo, K., Hu, Z., Liu, C.-K., So Kweon, I.: Non-local spatial propagation network for depth completion. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16, pp. 120–136 (2020). https://doi.org/10.1007/978-3-030-58601-0_8. Springer

[69] Huynh, L., Pedone, M., Nguyen, P., Matas, J., Rahtu, E., Heikkilä, J.: Monocular depth estimation primed by salient point detection and normalized hessian loss. In: 2021 International Conference on 3D Vision (3DV), pp. 228–238 (2021). https://doi.org/10.1109/3DV53792.2021.00033. IEEE

[70] Xian, C., Zhang, D., Dai, C., Wang, C.C.L.: Fast generation of high-fidelity rgb-d images by deep learning with adaptive convolution. IEEE Transactions on Automation Science and Engineering **18**(3), 1328–1340 (2021). https://doi.org/10.1109/TASE.2020.3002069

[71] Mandal, S., Bhavsar, A., Sao, A.K.: Depth map restoration from undersampled data. IEEE Transactions on Image Processing **26**(1), 119–134 (2017). https://doi.org/10.1109/TIP.2016.2621410

[72] Zhang, Y., Tian, Y., Kong, Y., Zhong, B., Fu, Y.: Residual dense network for image super-resolution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2472–2481 (2018)

[73] Li, Z., Yang, J., Liu, Z., Yang, X., Jeon, G., Wu, W.: Feedback network for image super-resolution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3867–3876 (2019)

[74] AlBahar, B., Huang, J.-B.: Guided image-to-image translation with bi-directional feature transformation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9016–9025 (2019)

[75] Cao, Y., Xu, Z., Li, F., Zhong, W., Huo, K., Ramani, K.: V. ra: An in-situ visual authoring system for robot-iot task planning with augmented reality. In: Proceedings of the 2019 on Designing Interactive Systems Conference, pp. 1059–1070 (2019). https://doi.org/10.1145/3322276.3322278

[76] Getting Started with Unity and ZED. https://www.stereolabs.com/docs/unity/. Accessed: 20121-11-10