Hindawi
Wireless Communications and Mobile Computing
Volume 2023, Article ID 9858963, 1 page
https://doi.org/10.1155/2023/9858963

Retraction

Retracted: Improved Multiview Decomposition for Single-Image High-Resolution 3D Object Reconstruction

Wireless Communications and Mobile Computing

Received 17 October 2023; Accepted 17 October 2023; Published 18 October 2023

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope
(2) Discrepancies in the description of the research reported
(3) Discrepancies between the availability of data and the research described
(4) Inappropriate citations
(5) Incoherent, meaningless and/or irrelevant content included in the article
(6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] J. Peng, K. Fu, Q. Wei, Y. Qin, and Q. He, "Improved Multiview Decomposition for Single-Image High-Resolution 3D Object Reconstruction," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 8871082, 14 pages, 2020.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8871082, 14 pages
https://doi.org/10.1155/2020/8871082

Research Article

Improved Multiview Decomposition for Single-Image High-Resolution 3D Object Reconstruction

Jiansheng Peng, Kui Fu, Qingjin Wei, Yong Qin, and Qiwen He

School of Physics and Mechanical and Electronic Engineering, Hechi University, Yizhou 546300, China

Correspondence should be addressed to Jiansheng Peng; 1692759628@qq.com

Received 3 September 2020; Revised 3 December 2020; Accepted 13 December 2020; Published 28 December 2020

Academic Editor: Shaohua Wan

As a representative technology of artificial intelligence, 3D reconstruction based on deep learning can be integrated into the edge computing framework to form an intelligent edge and then realize the intelligent processing of the edge. Recently, high-resolution representation of 3D objects using multiview decomposition (MVD) architecture is a fast reconstruction method for generating objects with realistic details from a single RGB image. The results of high-resolution 3D object reconstruction are related to two aspects. On the one hand, a low-resolution reconstruction network represents a good 3D object from a single RGB image. On the other hand, a high-resolution reconstruction network maximizes fine low-resolution 3D objects. To improve these two aspects and further enhance the high-resolution reconstruction capabilities of the 3D object generation network, we study and improve the low-resolution 3D generation network and the depth map superresolution network. Eventually, we get an improved multiview decomposition (IMVD) network. First, we use a 2D image encoder with multifeature fusion (MFF) to enhance the feature extraction capability of the model. Second, a 3D decoder using an effective subpixel convolutional neural network (3D ESPCN) improves the decoding speed in the decoding stage. Moreover, we design a multiresidual dense block (MRDB) to optimize the depth map superresolution network, which allows the model to capture more object details and reduce the model parameters by approximately 25% when the number of network layers is doubled. The experimental results show that the proposed IMVD is better than the original MVD in the 3D object superresolution experiment and the high-resolution 3D reconstruction experiment of a single image.

1. Introduction

The three-dimensional reconstruction of a single image is a hotspot and a difficult point in the field of computer vision. The purpose of the three-dimensional reconstruction of a single image is to reconstruct the corresponding 3D model structure from a single RGB image or a single depth image. The early 3D reconstruction of objects used the multiview geometry (MVG) method, which mainly studied structure-from-motion (SFM) [1, 2] recovery and simultaneous localization and mapping (SLAM) [3]. In addition, 3D object reconstruction also has methods based on prior knowledge [4, 5]. These traditional methods are often limited to a certain class of object in the 3D reconstruction of a single image, or it is difficult to generate a 3D object with better precision. With the continuous development of deep learning technology, the technology has been widely used in recent years [6–14], such as video analysis [8], image processing [9–11], medical diagnosis and service [12, 13], and target recognition [14]. Applying these to actual scenarios will encounter problems of large energy consumption and long response time. Using edge computing can effectively solve these problems. In the era of big data, data generated at the edge (e.g., images) also requires artificial intelligence technology to release its potential. Some research attempts to combine edge computing and deep learning include intelligent video surveillance [15], food recognition systems, [16], and self-driving cars [17]. At present, most of the research on edge computing and deep learning focuses on object recognition in two-dimensional space. However, for applications such as self-driving and

virtual reality, 3D reconstruction is the core technology. In the 3D reconstruction of objects, many methods try to extend the convolution operation in the two-dimensional space to the three-dimensional space to generate 3D shapes [18–20] and have achieved good research results. These methods all use a convolution operation based on dense voxels. As the running time and memory consumed increase cubically with the improvement of voxel resolution, the resolution of the generated models is limited to $64 \times 64 \times 64$. In order to solve the problem that the model generated by this method is limited to low resolution, some studies have proposed a sparse 3D reconstruction method using octrees [21–23]. Recently, the generative adversarial network (GAN) has shown great potential in image generation, and Yu et al. [24] also extended it to the 3D reconstruction of a single image. For the 3D reconstruction of a single image using GAN, this method consumes huge computing resources and also has a long training time. At present, the application of edge computing [25, 26] may be a feasible solution to this problem. Applying edge computing to traditional 3D reconstruction can generate 3D shapes faster, but the selection and processing of images may be a problem [27]. Therefore, combining edge computing and deep learning to achieve real-time 3D reconstruction of a single image may be a solution. In addition to the direct use of voxel methods to generate 3D shapes, other studies have used different three-dimensional representations, such as point clouds [28–30], meshes [31–33], primitives [34, 35], and implicit surfaces [36, 37]. Most of these methods can reconstruct three-dimensional objects with high resolution and are not limited by memory requirements. However, most of these methods need to solve the inherent defects of the model, such as using the point cloud method to reconstruct the surface details of the object and solving the genus problem of the mesh method to reconstruct the object.

For the voxel-based 3D object reconstruction method, it is robust to input. This method has the ability to adapt to 3D CNN and generate arbitrary topological structures. However, this method requires a huge amount of memory and calculations, and these factors make the resolution of the generated 3D shape too low. Therefore, how to solve the drawbacks of voxel-based 3D reconstruction is a premise for this method to generate high-resolution 3D shapes. At present, there are several methods for generating high-resolution 3D objects using voxel-based methods. As mentioned above, one of the methods is to use the sparse three-dimensional representation of the octree to generate high-resolution 3D shapes. It is also a method to transfer high-resolution 3D shape reconstruction to 2D space for implementation. Specifically, the method first uses the traditional 2D encoder-3D decoder architecture to generate a 3D object with low resolution from the input image. Then, superresolution reconstruction is performed on the 2D depth images of the low-resolution 3D object. Finally, the generated superresolution depth images are used for the reconstruction of a single high-resolution 3D object. In order to avoid directly manipulating voxels in a three-dimensional space, Richter and Roth [38] first predicted 6 depth maps of a 3D shape. They are then fused into a single reconstructed 3D shape. Smith et al. [39] also adopted a similar idea in the proposed MVD. They first used an encoder-decoder network to reconstruct the low-resolution 3D volume of a single image. Then, six orthographic depth maps of the low-resolution 3D object are obtained for superresolution reconstruction. Finally, the generated superresolution images are used to carve the upsampled low-resolution 3D shape to generate a high-resolution 3D object. This method can quickly accomplish high-resolution 3D object reconstruction of a single image.

However, the MVD method uses a traditional encoder-decoder network to generate low-resolution 3D shapes. This method has limited ability to extract image features in the 2D encoding stage, and the decoding speed in the 3D decoding stage is slow. In addition, the residual blocks (RB) used by MVD in depth image superresolution reconstruction do not fully utilize the features of different layers. This paper studies and improves these aspects to enhance the overall 3D reconstruction capabilities of the model. First, we improve the 2D encoder in the low-resolution 3D generation network into a 2D encoder with multifeature fusion to enhance the image feature extraction capability of the model. Then, we extend 2D ESPCN [40] to 3D ESPCN in the decoder stage to increase the speed of the decoder to generate 3D shapes. Second, this paper first introduces a single residual dense network (SRDN) on the basis of the residual network and dense network to improve. The residual network is then improved in a densely connected manner to maximize the reuse of features. Then, we obtain a multiresidual dense network (MRDN) to enhance the depth map superresolution network, which makes the network structure deeper and maximizes the information transfer between different convolutional layers. The experimental results show that the improved multiview decomposition (IMVD) structure performs better. First, the decoder using 3D ESPCN can increase the decoding speed of the model without degrading the performance of the model. Second, when the number of MRDB network layers is doubled compared to the number of RB network layers, the total model parameters and size are reduced by approximately 25%, respectively. Then, when the reconstructed object is in a relatively thin part, the reconstruction results of the MVD method are often broken. But our IMVD method can avoid this situation to some extent. In addition, the network that combines MFF and MRDB can capture more local features. The following sections are organized as follows. In Related Work, the current work related to this research is introduced. In Method, the improved MRDB and the low-resolution 3D object reconstruction network are introduced, respectively. In Experiment, the experiment is introduced, which includes the establishment of the dataset, the details of the training, and the relevant experimental results of each improvement component. In Conclusion, this paper is summarized.

The main contributions of this paper are summarized as follows:

(i) We propose an image encoder with multifeature fusion, which extracts the feature information of each layer to enhance the representation of the local details of the 3D shape. Compared with the traditional image encoder, the encoder with MFF is relatively more advantageous in capturing the detailed parts of 3D objects

(ii) We propose a 3D ESPCN operation to improve the traditional 3D decoder based on voxel representation, which reduces the time for the model to generate 3D shapes. Using 3D ESPCN can generate 3D shapes in lower resolution 3D volume spaces than traditional 3D decoders in the last step of the 3D decoding stage. This reduces the time required for the model to generate 3D shapes

(iii) We propose a multiresidual dense network to make full use of the features extracted from the residual network and the dense network. We connect the residual network in a dense manner and send the extracted features into the densely connected network. Model expression ability is improved by maximizing the reuse of features of each layer

2. Related Work

The goal of our work is to enhance its ability to generate high-resolution 3D objects from a single RGB image by improving the original MVD network. Wu et al. [18] earlier proposed the use of neural networks to recover the 3D shape of objects from 2.5D depth maps. Girdhar et al. [19] proposed a TL-embedding network. The network can complete the reconstruction from the RGB image to the 3D shape after training. These studies all apply a traditional encoder-decoder architecture, which uses progressive 2D convolution and 3D deconvolution for processing. Smith et al. [39] also used a similar structure to generate 3D shapes from 2D images. As we all know, in 2D image processing, the network layer that is too deep will cause the problem of gradient dispersion. When a network that is too deep can converge, its accuracy will also degrade. However, the deeper the network has also been proven to improve its performance. Therefore, it is an instinctive idea to introduce residual learning in the 3D reconstruction of a single image. Inspired by the residual network [41], Choy et al. [20] introduced a residual structure to design a deeper 3D object generation network. Their experimental results show that the network has a lower loss value in the training stage and can generate better 3D shapes than traditional 3D object generation networks. Similarly, Wu et al. [42] applied a similar residual structure in the 2D encoder. In addition, Soltani et al. [43] merged the residual block into the network to improve the performance of the model.

In the image superresolution, Dong et al. [44] first used convolutional neural networks to achieve superresolution reconstruction of low-resolution images. The input of this method is a high-resolution image after upsampling the low-resolution image. This superresolution method is complicated in operation and has a large amount of calculation. Subsequently, Shi et al. [40] proposed ESPCN. Different from upsampling input images to target resolution images for processing, they first use neural networks to extract features from low-resolution images. Then, the extracted features are recalculated using ESPCN operations to obtain high-resolution images. Since the feature extraction stage is performed on a lower resolution space, this method reduces the computational complexity of the entire superresolution process. Inspired by this, we first use a traditional 3D deconvolution operation to generate multiple low-resolution 3D volumes from the feature vector. Then, we expand ESPCN from 2D space to 3D space to generate a higher resolution 3D volume from these 3D volumes.

Recently, different network structures have appeared in image classification, such as the residual network (ResNet) [41] and the densely connected network (DenseNet) [45]. The purpose of introducing a residual network or densely connected network is to solve the problem of model degradation caused by designing a deeper network structure, and the deeper the network can extract more features to enhance the expression ability of the model. To reuse the feature information between more layers, a densely connected network is designed to solve the problem of gradient disappearance. Besides, the network structure designed in this way has a smaller model and requires less computation. Based on the above research, after analyzing the advantages and disadvantages of the residual block and the dense block, the Dual Path Network (DPN) [46] combines both to reduce the model parameters and to improve the training speed. Finally, better results were obtained in image classification, object detection, and semantic segmentation experiments. The relevant experimental results show that different structures have different benefits to the performance, parameter size, and computational complexity of the model.

Later on, various extended feature extraction structures were gradually introduced in the experiment of image superresolution reconstruction [47], such as the deep residual recurrent network (DRRN) [48] and the residual block [49]. In the superresolution experiment of 2D images, a multilayer feature concatenation method is often introduced to obtain more image feature information. Zhang et al. [50] proposed a residual dense network (RDN) after studying the residual block and the dense block. The output of each residual dense block (RDB) is processed through local feature fusion and global feature fusion. They further explore how to make full use of the features of different convolutional layers through this multifusion method. Wang et al. [51] introduced the residual-in-residual dense block (RRDB) to connect different network layers to make the model achieve better performance. Inspired by these studies, we study a multiresidual dense block to make full use of the features of each convolutional layer.

3. Method

In this section, we introduce an improved multiview decomposition (IMVD) network, as shown in Figure 1. The goal of this paper is to improve the MVD network to enhance the expression ability of the model and raise the quality of 3D object reconstruction. In the following content, we first describe the improved multiresidual dense block (MRDB) network. Second, a 2D encoder with multilayer feature fusion is described. Finally, we briefly introduce the 3D subpixel convolutional layer (3D SPCL) in 3D ESPCN.

3.1. Multiresidual Dense Network. The depth map superresolution network of MVD is based on the residual block in the

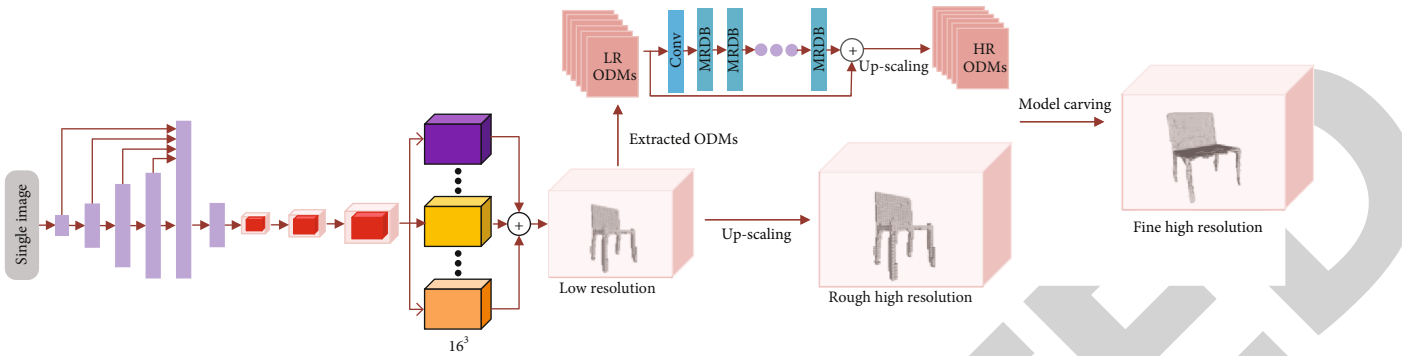


Figure 1: Improved single-RGB image high-resolution 3D reconstruction network structure. We apply the basic architecture of MVD [39]. The improved multiresidual dense block superresolution network processes six axis-aligned orthographic depth maps (ODMs). High-resolution depth maps and silhouette maps are estimated from the low-resolution ODMs, respectively. The improved multifeature fusion 2D encoder encodes the input RGB image into a 1024-dimensional latent vector and decodes it through 3D ESPCN.

generator of SRGAN [49]. Our improved superresolution network is based on a combination of the residual network and dense network. This improvement is to increase the connections between the convolutional layers to obtain more feature information and to design deeper and more complex structures.

Recent experiments have shown that connecting more layers in a network structure can further improve the performance of the model. Similarly, the use of denser connections in 2D images has also proved to enhance the performance of the model. Chen et al. [46] demonstrated that a single residual network has less redundancy in reusing features, and this shared information strategy makes it difficult to learn new features. However, a single densely connected network will lead to high redundancy while learning multiple new features. Finally, they designed a DPN with the advantages of the residual network and the densely connected network. In addition, Zhang et al. [50] also explored the combination of the residual network and the dense network. Their experimental results showed that the combination of both is beneficial. Similarly, we also take both into consideration. First, we introduce a single residual dense block (SRDB) [50]. Then, we improve on the basis of a single residual dense block and design a new multiresidual dense block (MRDB) by connecting the residual learning in a dense manner, as shown in Figure 2.

The MVD basic architecture uses sixteen residual blocks as shown in Figure 2(a). We maintain the basic architecture of MVD. We apply L multiresidual dense blocks as shown in Figure 2(c). The basic structure of the multiresidual dense network is shown in Figure 1. First, we consider a single image x_0 as the input of the superresolution network. Each layer of the network input consists of one or more components: batch normalization (BN) and convolution (Conv), and we represent these nonlinear transformations as $H_l(\cdot)$, where l indexes the layer. Then, $H_l(\cdot)$ in Figure 2 is in the form of Conv-BN-Conv-BN. Then, T denotes a transition layer consisting of a 1×1 convolution layer and batch normalization.

3.1.1. ResNet. Compared with the traditional CNN, inserting shortcut connections between different convolutional layers can convert it into a residual network, as shown in Figure 2(a). When the input and output dimensions of different convolutional layers are the same, the identity shortcut connection can be used to directly add its output to the output of the subsequent layer. When using the identity shortcut connection method, this connection method neither adds new parameters nor increases the computational complexity. For the residual network of Figure 2(a), the output x_{l-1} from the $(l-1)$th layer bypasses the nonlinear transformations with an identity function, and the results are added as the lth layer input. The residual network can be expressed as follows:

$$x_l = H_l(x_{l-1}) + x_{l-1}. \tag{1}$$

3.1.2. Single Residual Dense Network (SRDN). ResNet uses shortcut connections to solve the problem of model degradation to a certain extent. However, the connection between different layers of ResNet is a sparse connection. In order to make full use of the features of different layers, DenseNet uses the output of each layer as the input of each subsequent layer. This densely connected approach allows the model to achieve better performance than ResNet with fewer parameters and computational costs. In the single residual dense block of Figure 2(b), the input of the lth layer is derived from the output features of the previous 0th, 1th, $\cdots$, $(l-1)$th layers, x_0, $x_1, \cdots, x_{l-1}$:

$$x_l = H_l([x_0, x_1, \cdots, x_{l-1}]), \tag{2}$$

where $[x_0, x_1, \cdots, x_{l-1}]$ represents the concatenation operation. Equation (2) is also known as densely connected network output. Finally, a SRDB result consists of the input x_0 summed with the T output by a shortcut connection. We call this network SRDN, and its output can be expressed as

$$x_{\text{SRDB}} = T(x_l) + x_0. \tag{3}$$

3.1.3. Multiresidual Dense Network (MRDN). In each SRDB, DenseNet is applied to extract the features of different layers for fusion, and single residual learning is introduced to improve the information flow. It should be noted that residual learning in SRDB is not closely combined with DenseNet. In order to further improve the information flow, we fuse the residual learning of different layers with DenseNet. Now we

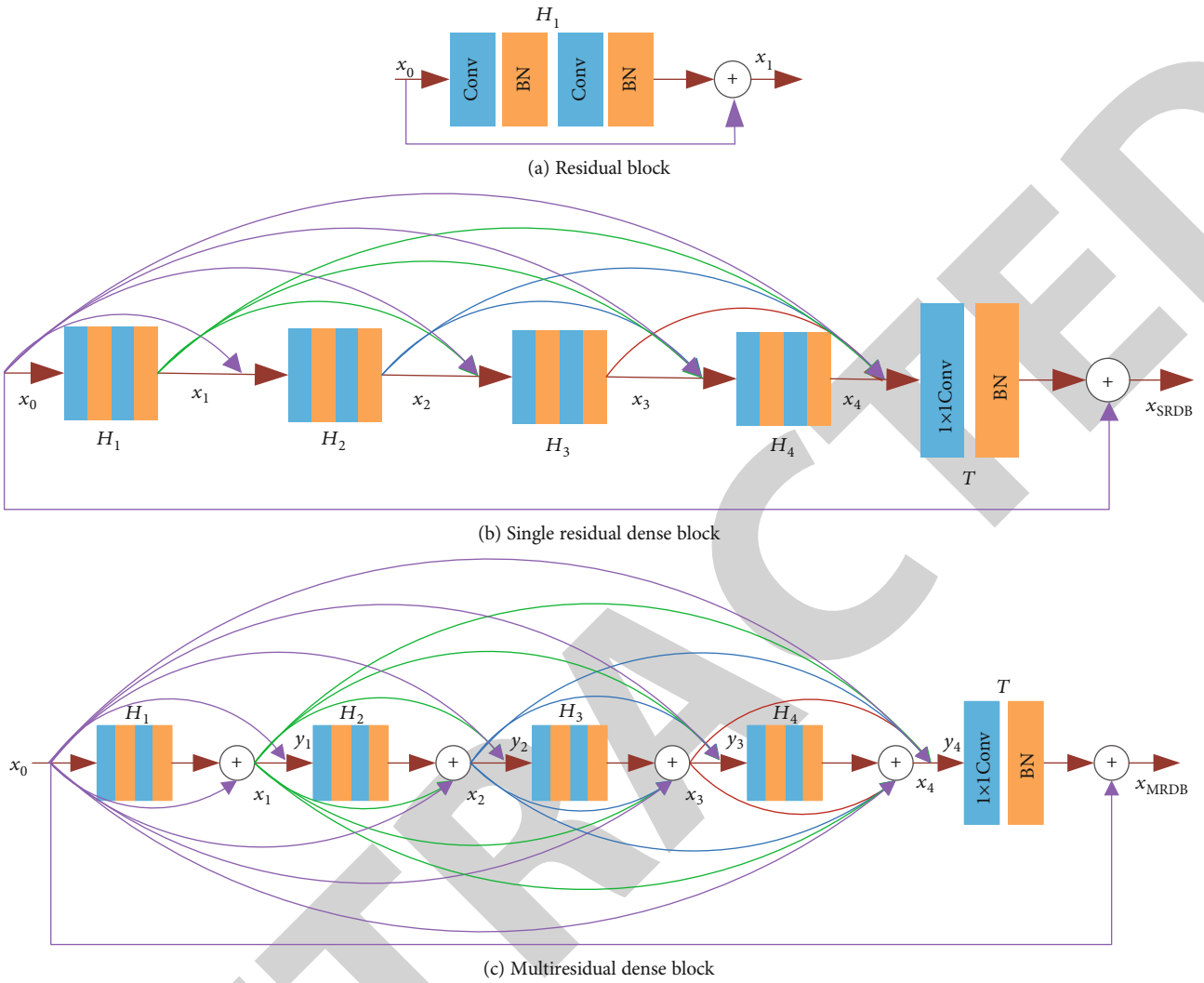


Figure 2: (a) Residual block in MVD [39]. (b) Single residual dense block in RDN [50]. (c) Our multiresidual dense block.

consider the multiresidual dense block of Figure 2(c). First, we denote the x_0 and y_0 as the residual input and dense input of a single MRDB, and $x_0 = y_0$. For x_1, it can be expressed as

$$x_1 = H_1(y_0) + x_0. \tag{4}$$

Then, y_1 is expressed as the fusion of residual output x_1 and x_0:

$$y_1 = [x_0, x_1]. \tag{5}$$

Combining Equations (4) and (5), it can be seen that the input of DenseNet in MRDB includes the output of RenseNet.

Further, we denote that x_l and y_l are the output of the residual network and the densely connected network in the lth layer, respectively. The lth layer accepts all of the preceding input feature maps $x_0, x_1, \cdots, x_{l-1}$ and the y_{l-1} of the $(l-1)$th layer as the residual output x_l:

$$x_l = H_l([y_{l-1}]) + \left(\sum_{t=0}^{l-1} x_t\right). \tag{6}$$

Similarly, we can get the output y_l of the lth layer:

$$y_l = [x_0, x_1, \cdots, x_l]. \tag{7}$$

Thus, transform Equation (7) into Equation (6), and Equation (6) can be further written as

$$x_l = H_l([x_0, x_1, \cdots, x_{l-1}]) + \left(\sum_{t=0}^{l-1} x_t\right). \tag{8}$$

Comparing Equation (8) with Equation (2), the first term on the right side of Equation (8) is formally equal to Equation (2). However, $x_1, \cdots, x_{l-1}$ in Equation (8) is essentially the residual input of Equation (1). In addition, Equation (8) adds a summation operation for all feature maps $x_0, x_1, \cdots, x_{l-1}$ of the preceding lth layer. From the above analysis, Equation (8) combines the features of the residual network and the dense network and expands them.

Finally, the output of a single MRDB can be expressed as

$$x_{MRDB} = T(y_l) + x_0. \tag{9}$$

We assume that the growth rate of the model is G [45]. Each $H_l(\cdot)$ produces G feature maps, and the result is $G_0 + G \times (l-1)$, where G_0 is the number of feature map channels of the input layer.

3.1.4. Implementation Details. We use the structure shown in Figures 2(b) and 2(c) in single residual dense networks and multiresidual dense networks, respectively. In the experiment, the kernel filter stride length of all convolutional layers is 1. The kernel depth G is 128 and 64, respectively. Since the multiresidual dense network has deeper and denser connections, it will inevitably lead to an increase in the parameters of the model. Performing 1×1 convolution after feature input is a common means of reducing model parameters [45, 50]. Our $H_l(\cdot)$ form is Conv(1×1)-BN-Conv(3×3)-BN. In addition, the final concatenation operation of each multiresidual dense block produces a large number of feature maps. We use 1×1 convolution to reduce its number and follow a batch normalization operation to feed the next multiresidual dense block. We let the number of single residual dense blocks and multiresidual dense blocks be L, which is set to 8 or 4 in the experiment.

3.2. Low-Resolution Network. The bottom of Figure 1 shows the overall low-resolution 3D reconstruction network. First, a 2D encoder with multifeature fusion is used to encode the input image into a fixed-length hidden layer vector. Then, traditional 3D deconvolution and 3D ESPCN are used to decode the latent vector to generate a low-resolution 3D volume. In the next part, we will introduce the 2D encoder with multifeature fusion and 3D ESPCN, respectively.

3.2.1. 2D Encoder with Multifeature Fusion. For coarse-to-fine 3D object reconstruction methods, high-quality low-resolution 3D object reconstruction is a basis for its higher resolution 3D reconstruction. In order to further improve the feature extraction capability of the 2D encoder to enhance the 3D reconstruction performance of the model, we use different layers of feature maps for fusion. An improved network comparison is shown in Figure 3.

Both encoder networks consist of a standard convolutional layer, a batch normalization layer, and a leaky rectified linear unit (LReLU). The encoder encodes the input data into a low-dimensional hidden vector, and the decoder decodes the compressed vector to reconstruct a 3D object. The advantage of this approach is that it can compress the input high-dimensional data into a low-dimensional representation and then reconstruct its 3D object through the representation.

By observing the traditional encoder of Figure 3(a), we find that the encoder of this mode has less utilization of features. In the image superresolution experiment of RDN [50], the global feature fusion (GFF) method proved to be able to improve the performance of the model. This is a method of extracting the output of all residual dense blocks in the network for fusion. Inspired by this, we extract the output from each nonlinear transformation $H_l(\cdot)$ in the encoder to fuse, as shown in Figure 3(b). To match the number of $H_l(\cdot)$ output feature map channels of different lth layers, we use a 1×1 convolution. The definition of $H_l(\cdot)$ is consistent with Section 3.1. Since the number of convolution channels after feature fusion is too large, their direct compression to a 1024-dimensional feature vector will result in huge model parameters. Therefore, we use a 1×1 convolution to reduce the dimensions of the fused features. The multifeature fusion encoder output is expressed as

$$x_{\mathrm{MFF}} = T([x_1, x_2, \cdots, x_l]). \tag{10}$$

Finally, the output of the encoder is compressed to a 1024-dimensional feature vector through a flat layer and a fully connected layer. We find that multilayer feature fusion can encourage models to learn new features.

3.2.2. 3D Subpixel Convolution Layer. In the image superresolution experiment, combining multiple low-resolution images (feature maps in low-resolution space) to generate a higher resolution image is a more efficient processing method [40]. Inspired by this, in the voxel-based 3D convolutional neural network, multiple low-resolution 3D shapes can be combined into a higher resolution 3D shape. This operation can be named 3D SPCL, as shown in Figure 4.

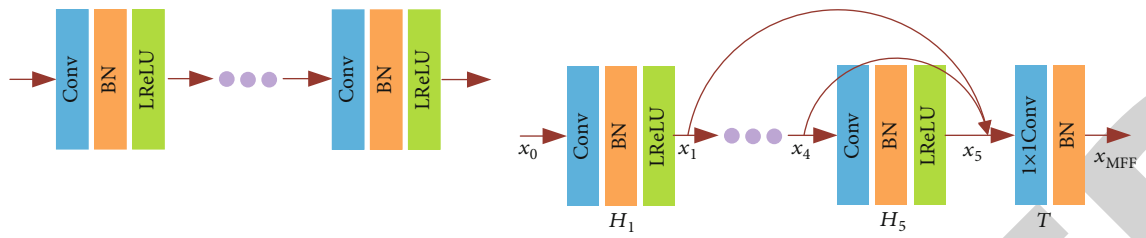
Generally, the size of a single low-resolution 3D volume and a single high-resolution 3D volume can be expressed as $H \times W \times D$ and $nh \times nW \times nD$, respectively. We will refer to n as the upscaling ratio. First, a traditional voxel-based decoder is used to generate n^3 low-resolution 3D shapes from the latent space, the size of which is $H \times W \times D \cdot n^3$. Then, 3D SPCL is used to rearrange the generated n^3 low-resolution 3D shapes into one high-resolution 3D shape. 3D SPCL is a periodic operation that rearranges the elements of the $H \times W \times D \times n^3$ tensor to a tensor of shape $nH \times W \times D \cdot n^2$. Then, the W channel and the D channel are arranged in sequence. Finally, a tensor of shape $nH \times nW \times nD$ is the output. The entire 3D SPCL does not involve convolution operations. Compared with the traditional 3D decoding method based on voxels, this method reduces the 3D deconvolution operation at higher resolution. Therefore, using 3D SPCL when generating 3D shapes can make the model have a faster decoding speed.

4. Experiment

In this part, we show the experimental results of the improved multiview decomposition (IMVD) network for 3D object superresolution and 3D object reconstruction of a single RGB image. In addition, we analyze the importance of each component in the network. The qualitative and quantitative results show that the proposed method can improve the expression ability of the model.

4.1. Dataset and Metric

4.1.1. 3D Object Superresolution Dataset. The 3D object superresolution dataset consists of a $32 \times 32 \times 32$ low-resolution voxel model and a corresponding $256 \times 256 \times 256$ high-resolution voxel model. Following the MVD approach, we also use the ShapeNetCore [52] dataset to transform CAD models into 3D shapes represented by voxels. Two classes are selected from the ShapeNetCore dataset: chair and plane. Their



(a) Traditional encoder network (b) Encoder network with multifeature fusion

Figure 3: Improved network structure for comparison. (a) Traditional encoder network in MVD [39]. (b) Our encoder network.

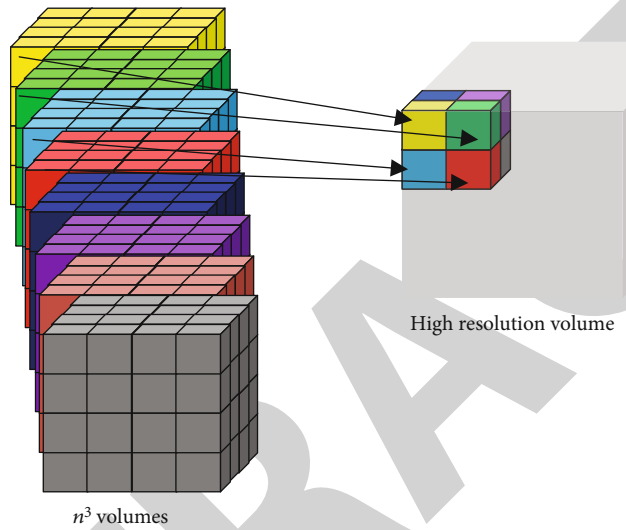


Figure 4: The 3D SPCL operation rearranges multiple low-resolution 3D volumes into a high-resolution 3D volume.

numbers are approximately 7000 and 4000, respectively. We preprocess the 3D object superresolution dataset and extract 6 orthographic depth maps (ODMs) for each object in the dataset corresponding to low resolution and high resolution. The final dataset is divided into a training set, a validation set, and a test set. We used 70% of the dataset as the training set, 10% as the validation set, and 20% as the test set. The dataset we created is named 3D superresolution dataset ($Data_{SR}$).

4.1.2. Low-Resolution 3D Reconstruction Dataset. The 3D object reconstruction experimental dataset of a single RGB image is based on $Data_{SR}$. Similarly, we refer to the relevant dataset production methods in MVD. Based on the completed $Data_{SR}$, we render each CAD model as a 128×128 RGB image to obtain a random viewpoint and possible azimuthal rotation of the object between $(-20°, 30°)$. Similarly, the completed dataset is divided into a training set, a validation set, and a test set according to the 3D superresolution experimental dataset, with a ratio of 70 : 10 : 20, respectively. Finally, the dataset we follow is named $Data_{HSP}$.

4.1.3. Evaluation Metric. In all 3D reconstruction experiments, the evaluation metric uses the intersection over union (IoU). Applying IoU to evaluate the corresponding model on the $Data_{SR}$ and $Data_{HSP}$ enables quantitative analysis of model performance.

4.2. Training Details. We train the entire model in two stages. The 3D superresolution model and the low-resolution 3D reconstruction model are separately trained. Finally, the two training models of the two stages are combined to form the final high-resolution 3D object reconstruction model of a single RGB image, which is the improved multiview decomposition (IMVD) network.

In the 3D object superresolution experiment, the silhouette estimation network and the depth estimation network are, respectively, trained. Following the MVD, the 3D object superresolution experiment was reconstructed from $32 \times 32 \times 32$ resolution to $256 \times 256 \times 256$ resolution. The dataset used for model training comes from the 3D superresolution dataset described in Section 4.1. During the training process, both use the Adam [53] default parameter training, the learning rate is 10^{-4}, the training minimum batch size is 32, the training epoch is 300, and the error function uses the mean square error (MSE) loss function. The training set is used for network training, and the validation set is used to evaluate model performance at the end of each epoch. The current model is retained only if the IoU score of the reconstruction

result evaluation is greater than the largest IoU score of the previous reconstruction result.

In a low-resolution 3D object reconstruction experiment, the encoder with multifeature fusion and the 3D ESPCN decoder are trained. Using the Adam optimizer, the learning rate is 10^{-3}, the training minimum batch is 128, the training epoch is 300, and the mean square error term is used as the loss function. The update of the model is the same as the operation in the 3D object superresolution experiment.

After the silhouette estimation network, the depth estimation network, and the low-resolution 3D object reconstruction network have all been trained, the 3D model carving combines three networks to accomplish the high-resolution reconstruction. For model carving, it includes silhouette carving and depth map carving. Firstly, the rough 3D shape after upsampling is carved using estimated silhouette maps to ensure the correctness of its structure. Then, the estimated depth maps will be used for detail carving. The voxels that have not reached the corresponding depth in the 3D shape after silhouette carving will be deleted. We implemented the model with the TensorFlow Architecture and trained on a single NVIDIA GTX 1080 GPU.

4.3. 3D Object Superresolution Experiment

4.3.1. Model Parameters, Size, and IoU Comparison. Table 1 shows the experimental comparison of SRDN and MRDN on the Data_{SR} chair for different block numbers L (8 or 4) and different size feature maps G (128 or 64). The number in italic in Table 1 indicates the highest IoU score for the corresponding category of 3D reconstruction. We use SRDN and MRDN to improve MVD in superresolution experiments of the chair and can achieve higher IoU scores than MVD. We roughly calculate the number of MVD superresolution network layers with 16 residual blocks as shown in Figure 2(a), and the total number of layers is 32. Similarly, the number of IMVD network layers improved by MRDB is 72.

As can be seen from Table 1, when the number of network layers is increased by about 1 time, the MRDN model parameters are reduced by about 25%. At the expense of the IoU reconstruction score, the model parameters are reduced by 81% when the feature map G is reduced by half. We observe that in the MRDB experiment, keeping the feature map G constant and reducing L by half make the model IoU fall. This suggests that designing deeper networks can enhance the expressive ability of the model. In Table 1, MRDN-4 ($G = 128$) and MRDN-8 ($G = 64$) are scaled-down on L and G, respectively. Although the IoU scores are almost the same, the latter model parameters are reduced by approximately 56%. In addition, the MRDN model parameters can be reduced by 45% when SRDN and MRDN are close to the obtained IoU score.

4.3.2. Qualitative Results. We show qualitative results in Figure 5. We rendered from 32^3 resolution to 256^3 on the test set. The low-resolution 3D shapes of real chairs and planes are used as input for this experiment (line 1 of Figure 5). The output results of MVD [39] are shown in line 2 of Figure 5. The IMVD results are shown in line 3 of Figure 5.

Table 1: Comparison of parameters and IoU (%) on the Data_{SR} chair model. "-" means that the model is out of our running memory without IoU results. "*" indicates the result of our implementation.

Method	Parameters	Size	IoU
RB [39]	5.28M	21.1M	68.4*
MRDN-4 ($G = 128$)	2.25M	9.0M	69.3
SRDN-8 ($G = 64$)	1.83M	7.3M	69.1
MRDN-8 ($G = 64$)	1.00M	4.0M	69.2
SRDN-8 ($G = 128$)	7.27M	-	-
MRDN-8 ($G = 128$)	3.97M	15.9M	69.8

As can be seen from the comparison of Figure 5, the MVD method tends to break in a thin object portion. However, our IMVD results are more complete in this situation. The experimental results show that extracting more feature information through the multiresidual dense network is beneficial to enhance the expressive ability of the model.

4.3.3. Quantitative Results. We trained each class in Data_{HSP} separately in a 3D object superresolution experiment. The results are compared with various methods employed in MVD and presented in Table 2. The benchmark method directly increases the resolution of the 3D volume from 32^3 to 256^3 through the nearest neighbor upsampling. The MVD method combines depth estimation and silhouette estimation. It can be seen from Table 2 that our method performs better than the MVD method in the experiment. We all achieved higher scores in different categories.

4.4. Single-Image 3D Reconstruction Experiment

4.5. Model Parameters and Iteration Time. We show the parameter sizes and required iteration time of different low-resolution 3D reconstruction models, as shown in Table 3. It can be seen from Table 3 that IMVD has increased in the number of parameters and decreased in iteration time. Generally, 3D reconstruction experiments of a single image often use 13 categories in the ShapeNetCore dataset. The total number of models in 13 categories is approximately 39,832. According to the method of generating the dataset in this article, the number of models in the training set of each category is approximately 2,144. According to the iteration time in Table 3 and the training method in this paper, the training time of IMVD in 13 categories will be reduced by approximately 4 hours compared with MVD. For higher resolution 3D reconstruction experiments, this method has more advantages in training time.

4.5.1. Convergence Curve Analysis. In Figure 6, we show the convergence curve on the validation set. In Figures 6(a) and 6(b), the red curves represent the convergence of the MVD method on the chair and aircraft validation set, respectively. Similarly, the green curve corresponds to our IMVD method. We train the model to use the same parameters, just changing the structure of the model. The training epoch was 300, and the reconstructed IoU score was evaluated on the validation set at the end of each epoch. The original MVD oscillated

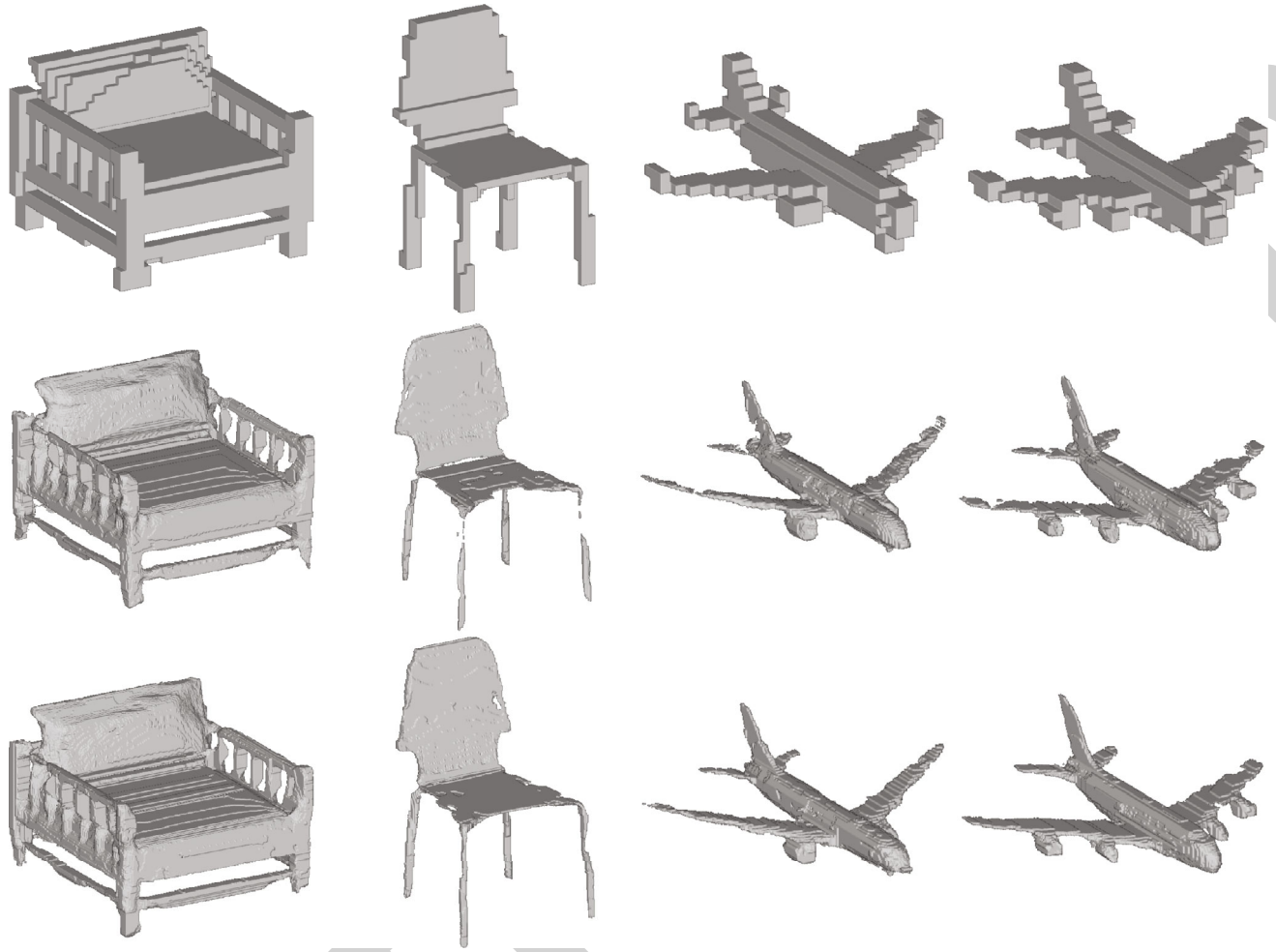


Figure 5: 3D object superresolution results on $Data_{SR}$.

Table 2: 3D object superresolution reconstruction IoU score at 256^3.

Class	Benchmark [39]	Depth [39]	Silhouette [39]	MVD [39]	IMVD (ours)
Chair	54.9	58.5	67.3	68.5	69.8
Plane	39.9	50.5	70.2	71.1	72.9

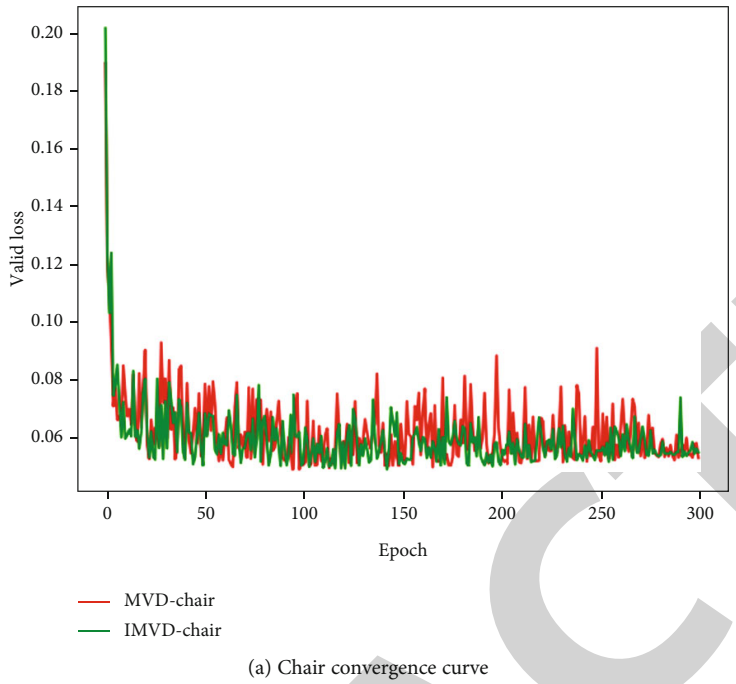
Table 3: Model parameters and iteration time at 32^3 resolution. The batch size is 2.

Method	Parameters (M)	Iteration time (ms)
MVD [39]	27.02	50.8
MVD+MFF	27.15	49.9
MVD+3D ESPCN	27.01	47.7
IMVD	27.14	47.1

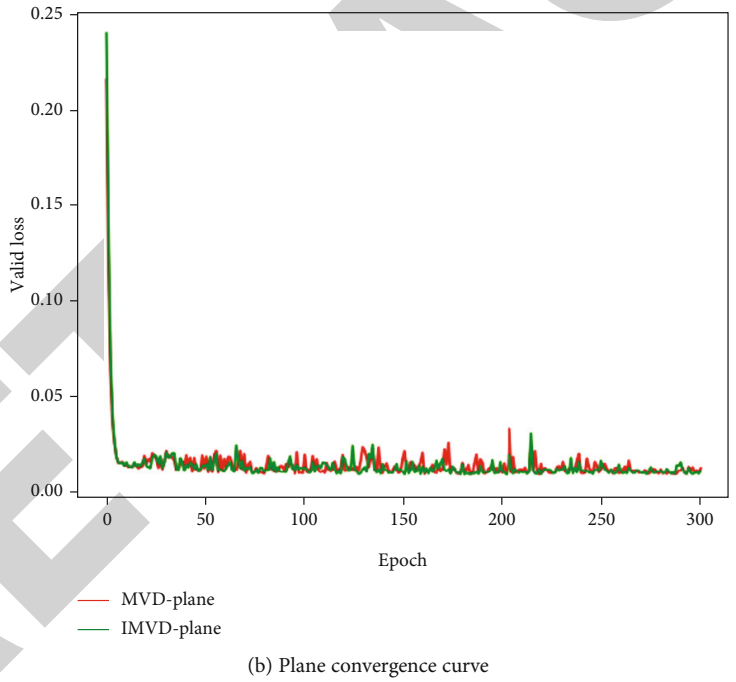
over the entire training cycle of the training chair. Our IMVD uses a multifeature fusion approach to reduce the degree of model oscillation, which helps to improve the model expression ability. In Figure 6(b), the model of the aircraft itself has no complicated and thin parts like a chair. Therefore, it seems that there is not much difference between the improved convergence curves of the IMVD network and the original MVD network on the validation set. In summary, we can see from the comparative analysis in Figure 6 that the improved network can improve the stability of model training.

4.5.2. Quantitative Results. We show quantitative results in Table 4. We compared several methods, HSP [22], AE [39], and MVD [39], which all use $Data_{HSP}$ to reconstruct 3D objects from a single RGB image at 256^3 resolution. As can be seen from Table 4, the proposed IMVD method can achieve a higher IoU score on a single-image reconstruction 256^3 resolution 3D object.

4.6. Ablation Studies. Table 5 quantitatively demonstrates the effects of MFF, 3D ESPCN, and MRDB. The IoU scores of the reconstruction results are in the second column, and the third column corresponds to the plane and the chair, respectively. The last column represents the average IoU score for the plane and chair reconstruction results. The first column in Table 5 represents the combination of the different components we proposed. Among them, the benchmark is the method of MVD. We add MFF and MRDB (from line 3 to line 4 of Table 5) to the benchmark method. Since the



(a) Chair convergence curve



(b) Plane convergence curve

Figure 6: Convergence curve analysis on the validation set. The curve represents the evaluation of the loss value over 300 epochs of the corresponding validation set on the Data_{HSP}.

Table 4: Single-image reconstruction IoU score at 256^3 resolution.

Class	AE [39]	HSP [22]	MVD [39]	IMVD (ours)
Chair	36.4	37.8	40.1	41.9
Plane	28.6	56.1	56.4	58.8

Table 5: The IoU score evaluates the contribution of each component.

Component	Chair	Plane	Average
Benchmark [39]	40.1	56.4	48.25
3D ESPCN	40.2	56.4	48.30
MFF	41.2	57.9	49.55
MRDB	41.3	57.0	49.15
MFF+MRDB	41.9	58.6	50.25

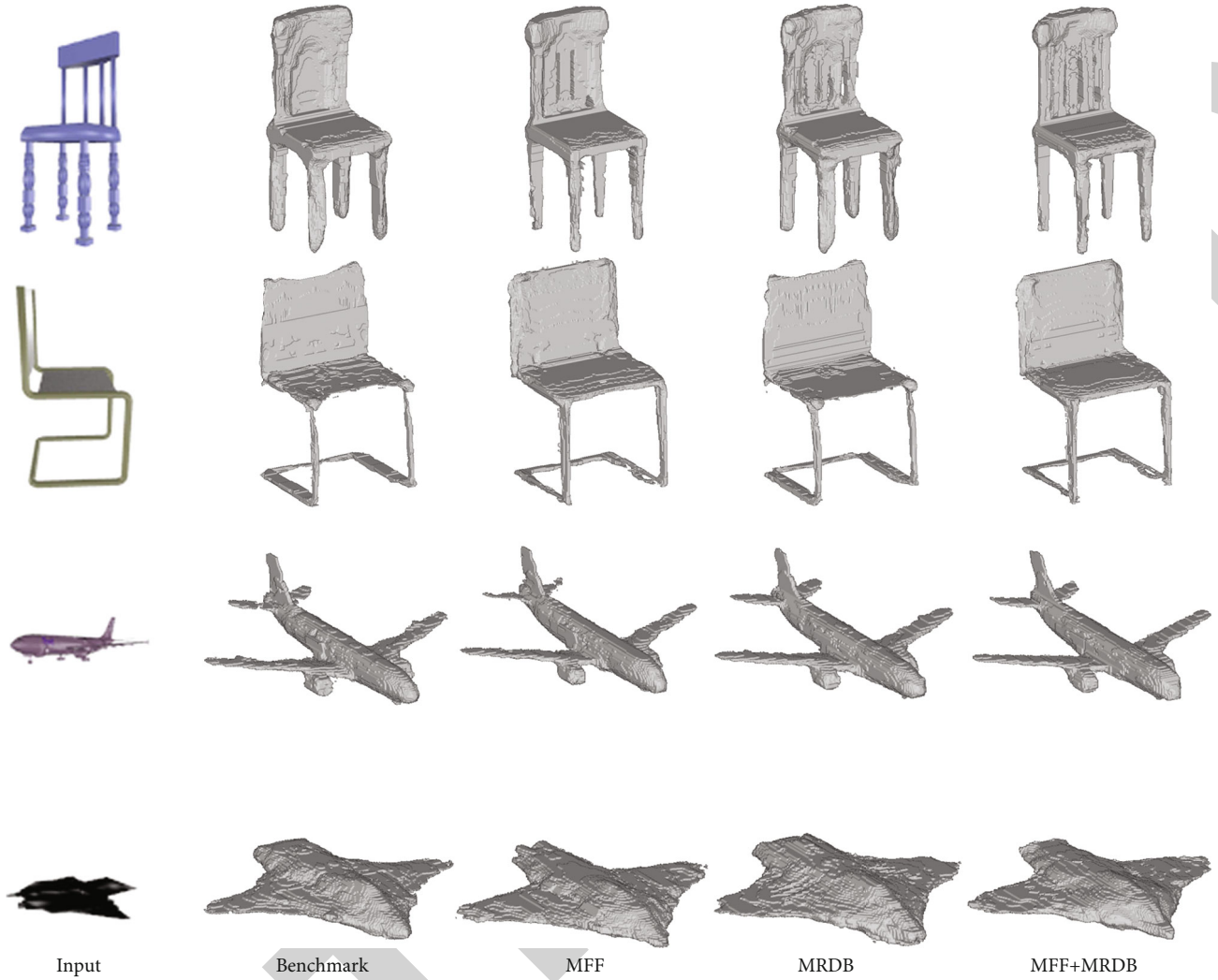


Figure 7: Qualitative results of the ablation study. This figure reflects the contribution of each improved component to the model.

addition of 3D ESPCN basically did not improve the performance of the model, it can be seen that adding another component can improve the performance of the model. We add modules for the combination of MFF and MRDB on the benchmark (in the last row of Table 5). After adding two components, the performance of the model has been further improved.

Figure 7 qualitatively shows the contribution of MFF and MRDB in the model. The first column of Figure 7 represents the input RGB image. The second column is a method of MVD, and the reconstruction result is broken at the edge portion (columns 3 to 5 of Figure 7). However, partial fractures have been improved after the addition of MFF or MRDB. In addition, it can be seen in the reconstruction of the first row of the chair in Figure 7 that the input RGB image of the chair back is a series of unconnected pillars. However, the 3D reconstruction result of MVD does not reflect this feature. After adding MFF or MRDB alone, the reconstruction results show this part of the details. This detail can be further enhanced after combining MFF and MRDB. It can be seen from the comparison of the third column to the fifth column of Figure 7 that the final reconstruction result of IMVD is mainly refined based on MFF. This also reflects the impact of the resolution of low-resolution 3D object reconstruction on high-resolution 3D object representation. At present, the rendering of CAD models in the dataset is performed in random colors, and the background of all rendered images is clean. In the future, images with textures and backgrounds can be used for rendering to enrich the dataset, which will make the model more robust to 3D object reconstruction from 2D images in real scenes. In addition, there are other methods, such as exploring new algorithms to extract more effective image features, using different training architectures, and supervising methods to optimize [54].

5. Conclusion

We improve the depth map superresolution network and low-resolution 3D reconstruction network of the single image in MVD, respectively. The improved model shows better performance compared with MVD in the corresponding experiment. We propose an architecture that includes multiple MRDB blocks, which can make the network structure design deeper and make full use of the multilayer

structure information to enhance the model expression ability. Even though the network design is deeper, the model parameters are even smaller. In addition, we use multifeature fusion and 3D ESPCN to improve the 2D encoder and 3D decoder, respectively. Both of these can reduce the training time of the model. At present, there are few studies on 3D reconstruction technology and edge computing based on deep learning, but their combination has broad application prospects. In intelligent manufacturing, edge computing is conducive to extend various computing resources to the edge of the Internet of Things and realizes manufacturing and production [55]. However, the problem of 3D data heterogeneity between different devices may need to be resolved. The use of 3D reconstruction methods based on deep learning may be one of the means to solve this problem in the future.

Data Availability

The 3D model dataset used to support the findings of this study can be downloaded from the public website: https://www.shapenet.org/.

Conflicts of Interest

No potential conflict of interest was reported by the authors.

Authors' Contributions

Jiansheng Peng, Kui Fu, and Qingjin Wei contributed equally to this work.

Acknowledgments

The authors are highly thankful to the National Natural Science Foundation of China (NO. 62063006), to the Development Research Center of Guangxi Relatively Sparse-populated Minorities (ID: GXRKJSZ201901), and to the Natural Science Foundation of Guangxi Province (NO. 2018GXNSFAA281164). This research was financially supported by the project of outstanding thousand young teachers' training in higher education institutions of Guangxi, Guangxi Colleges and Universities Key Laboratory Breeding Base of System Control and Information Processing.

References

[1] J. L. Schönberger and J. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4104–4113, Las Vegas, NY, USA, 2016.

[2] K. Haming and G. Peters, "The structure-from-motion reconstruction pipeline-a survey with focus on short image sequences," *Kybernetika*, vol. 46, no. 5, pp. 926–937, 2010.

[3] C. Cadena, L. Carlone, H. Carrillo et al., "Past, present, and future of simultaneous localization and mapping: toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[4] L. Galteri, C. Ferrari, G. Lisanti, S. Berretti, and A. Del Bimbo, "Deep 3D morphable model refinement via progressive growing of conditional generative adversarial networks," *Computer Vision and Image Understanding*, vol. 185, pp. 31–42, 2019.

[5] A. Kar, S. Tulsiani, J. Carreira, and J. Malik, "Category-specific object reconstruction from a single image," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1966–1974, Boston, MA, USA, 2015.

[6] Z. Yao, D. He, Y. Chen et al., "Inspection of exterior substance on high-speed train bottom based on improved deep learning method," *Measurement*, vol. 163, article 108013, 2020.

[7] L. Li, T. T. Goh, and D. Jin, "How textual quality of online reviews affect classification performance: a case of deep learning sentiment analysis," *Neural Computing and Applications*, vol. 32, no. 9, pp. 4387–4415, 2020.

[8] S. Wan, X. Xu, T. Wang, and Z. Gu, "An intelligent video analysis method for abnormal event detection in intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–9, 2020.

[9] S. Wan, Y. Xia, L. Qi, Y.-H. Yang, and M. Atiquzzaman, "Automated colorization of a grayscale image with seed points propagation," *IEEE Transactions on Multimedia*, vol. 22, no. 7, pp. 1756–1768, 2020.

[10] Y. Xi, Y. Zhang, S. Ding, and S. Wan, "Visual question answering model based on visual relationship detection," *Signal Processing: Image Communication*, vol. 80, article 115648, 2020.

[11] S. Ding, S. Qu, Y. Xi, and S. Wan, "Stimulus-driven and concept-driven analysis for image caption generation," *Neurocomputing*, vol. 398, pp. 520–530, 2020.

[12] C. Zhang, X. Guo, X. Guo et al., "Machine learning model comparison for automatic segmentation of intracoronary optical coherence tomography and plaque cap thickness quantification," *Computer Modeling in Engineering & Sciences*, vol. 123, no. 2, pp. 631–646, 2020.

[13] S. Wan, Z. Gu, and Q. Ni, "Cognitive computing and wireless communications on the edge for healthcare service robots," *Computer Communications*, vol. 149, pp. 99–106, 2020.

[14] S. Wan and S. Goudos, "Faster R-CNN for multi-class fruit detection using a robotic vision system," *Computer Networks*, vol. 168, article 107036, 2020.

[15] J. Chen, K. Li, Q. Deng, K. Li, and P. S. Yu, "Distributed deep learning model for intelligent video surveillance systems with edge computing," *IEEE Transactions on Industrial Informatics*, 2019.

[16] C. Liu, Y. Cao, Y. Luo et al., "A new deep learning-based food recognition system for dietary assessment on an edge computing service infrastructure," *IEEE Transactions on Services Computing*, vol. 11, pp. 249–261, 2018.

[17] A. Ndikumana, N. H. Tran, D. H. Kim, K. T. Kim, and C. S. Hong, "Deep learning based caching for self-driving cars in multi-access edge computing," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–16, 2020.

[18] Z. Wu, S. Song, A. Khosla et al., "3D ShapeNets: a deep representation for volumetric shapes," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1912–1920, Boston, MA, USA, 2015.

[19] R. Girdhar, D. F. Fouhe, M. Rodriguez, and A. Gupta, "Learning a predictable and generative vector representation for objects," in *Computer Vision – ECCV 2016. ECCV 2016*, pp. 484–499, Springer, 2016.

[20] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese, "3D-R2N2: a unified approach for single and multi-view 3D object

reconstruction," in *Computer Vision – ECCV 2016. ECCV 2016*, pp. 628–644, Springer, 2016.

[21] M. Tatarchenko, A. Dosovitskiy, and T. Brox, "Octree generating networks: efficient convolutional architectures for high-resolution 3D outputs," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2088–2096, Venice, Italy, 2017.

[22] C. Häne, S. Tulsiani, and J. Malik, "Hierarchical surface prediction for 3D object reconstruction," in *2017 International Conference on 3D Vision (3DV)*, pp. 76–84, Qingdao, China, 2017.

[23] G. Riegler, A. O. Ulusoy, H. Bischof, and A. Geiger, "OctNetFusion: learning depth fusion from data," in *2017 International Conference on 3D Vision (3DV)*, pp. 57–66, Qingdao, China, 2017.

[24] S. Yu, X. Chen, S. Wang, L. Pu, and D. Wu, "An edge computing-based photo crowdsourcing framework for real-time 3D reconstruction," *IEEE Transactions on Mobile Computing*, 2020.

[25] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum, "Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 82–90, Barcelona, Spain, 2016.

[26] X. Xu, X. Zhang, X. Liu, J. Jiang, L. Qi, and M. Z. A. Bhuiyan, "Adaptive computation offloading with edge for 5G-envisioned internet of connected vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[27] X. Xu, X. Liu, X. Yin, S. Wang, Q. Qi, and L. Qi, "Privacy-aware offloading for training tasks of generative adversarial network in edge computing," *Information Sciences*, vol. 532, pp. 1–15, 2020.

[28] H. Fan, H. Su, and L. Guibas, "A point set generation network for 3D object reconstruction from a single image," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 605–613, Honolulu, HI, USA, 2017.

[29] K. L. Navaneet, P. Mandikal, M. Agarwal, and R. V. Babu, "CAPNet: continuous approximation projection for 3D point cloud reconstruction using 2D supervision," in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 8819–8826, Hilton Midtown, NY, USA, 2019.

[30] P. Mandikal and V. B. Radhakrishnan, "Dense 3D point cloud reconstruction using a deep pyramid network," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1052–1060, Waikoloa Village, HI, USA, 2019.

[31] H. Kato and T. Harada, "Learning view priors for single-view 3D reconstruction," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9778–9787, Long Beach, CA, USA, 2019.

[32] C. Wen, Y. Zhang, Z. Li, and Y. Fu, "Pixel2Mesh++: multi-view 3D mesh generation via deformation," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1042–1051, Munich, Germany, 2019.

[33] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry, "A papier-mâché approach to learning 3D surface generation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 216–224, Salt Lake City, UT, USA, 2018.

[34] C. Zou, E. Yumer, J. Yang, D. Ceylan, and D. Hoiem, "3D-PRNN: generating shape primitives with recurrent neural networks," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 900–909, Venice, Italy, 2017.

[35] P. S. Wang, Y. Liu, Y. X. Guo, C. Sun, and X. Tong, "O-CNN: octree-based convolutional neural networks for 3D shape analysis," *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 72–81, 2016.

[36] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "DeepSDF: learning continuous signed distance functions for shape representation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 165–174, Long Beach, CA, 2019.

[37] Q. Xu, W. Wang, D. Ceylan, R. Mech, and U. Neumann, "DISN: deep implicit surface network for high-quality single-view 3D reconstruction," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 495–502, Vancouver, Canada, 2019.

[38] S. R. Richter and S. Roth, "Matryoshka networks: predicting 3D geometry via nested shape layers," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1936–1944, Salt Lake City, USA, 2018.

[39] E. Smith, S. Fujimoto, and D. Meger, "Multi-view silhouette and depth decomposition for high resolution 3D object representation," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 6479–6489, Montréal, Canada, 2018.

[40] W. Shi, J. Caballero, F. Huszar et al., "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1874–1883, Las Vegas, NY, USA, 2016.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, 2016.

[42] J. Wu, Y. Wang, T. Xue, and X. Sun, "MarrNet: 3D shape reconstruction via 2.5D sketches," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 8–15, Long Beach, CA, USA, 2017.

[43] A. A. Soltani, H. Huang, J. Wu, T. D. Kulkarni, and J. B. Tenenbaum, "Synthesizing 3D shapes via modeling multi-view depth maps and silhouettes with deep generative networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1511–1519, HI, USA, 2017.

[44] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proceedings of the European Conference on Computer Vision*, pp. 184–199, Cham, 2014.

[45] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700–4708, Honolulu, USA, 2017.

[46] Y. Chen, J. Li, H. Xiao, X. Jin, S. Yan, and J. Feng, "Dual path networks," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 4467–4475, Long Beach, CA, USA, 2017.

[47] K. Fu, J. Peng, H. Zhang, X. Wang, and F. Jiang, "Image super-resolution based on generative adversarial networks: a brief review," *CMC-Computers, Materials & Continua*, vol. 64, no. 3, pp. 1977–1997, 2020.

[48] Y. Tai, J. Yang, and X. Liu, "Image super-resolution via deep recursive residual network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3147–3155, Honolulu, HI, USA, 2017.

[49] C. Ledig, L. Theis, F. Huszár et al., "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4681–4690, Honolulu, HI, USA, 2017.

[50] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2472–2481, Salt Lake City, USA, 2018.

[51] X. Wang, K. Yu, S. Wu et al., "ESRGAN: enhanced super-resolution generative adversarial networks," in *Proceedings of the European Conference on Computer Vision*, pp. 63–79, Munich, Germany, 2018.

[52] A. X. Chang, T. Funkhouser, L. Guibas et al., "ShapeNet: an information-rich 3D model repository," 2015, http://arxiv.org/abs/1512.03012.

[53] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," 2014, http://arxiv.org/abs/1412.6980.

[54] K. Fu, J. Peng, Q. He, and H. Zhang, "Single image 3D object reconstruction based on deep learning: a review," *Multimedia Tools and Applications*, pp. 1–36, 2020.

[55] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: a comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 869–904, 2020.