

Received August 3, 2020, accepted September 3, 2020, date of publication September 15, 2020, date of current version September 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3024288

# 3D Model Inpainting Based on 3D Deep Convolutional Generative Adversarial Network

XINYING WANG<sup>1</sup>, DIKAI XU<sup>1</sup>, AND FANGMING GU<sup>2</sup>

<sup>1</sup>College of Computer Science and Engineering, Changchun University of Technology, Changchun 130012, China

<sup>2</sup>College of Computer Science and Technology, Jilin University, Changchun 130012, China

Corresponding author: Xinying Wang (wang\_xinying1979@163.com)

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61806024 and in part by the Jilin Provincial Department of Education Science and Technology Project under Grant JJKH20200678KJ.

**ABSTRACT** In recent years, the problem of hole repairing in the 3D model has been widely concerned in related fields. As the Generative Adversarial Network (GAN) has achieved great success in generating realistic images, a 3D mesh model repair method based on the 3D Deep Convolutional Generative Adversarial Network (3D-DCGAN) is proposed in this paper. The algorithm contains two GANs: a local GAN and a global GAN. Four steps have been used to implement this concept. First, the 3D model is voxelized, and a mask is used to identify the repairing area; Second, the repairing area is generated by training local GAN; Third, the repaired region is combined with the 3D model to be repaired, thereafter, the global GAN is trained with the combined model. Finally, a decent repaired model is obtained with the perfect transition. The experimental results show that this algorithm can effectively generate the repairing area while retaining the details of the area and blend it with the model to be repaired.

**INDEX TERMS** 3D model inpainting, GAN, local GAN, global GAN.

## I. INTRODUCTION

As a way to record object information, 3D model images possess a lot of features, significantly better than 2D images. Nowadays, they are widely used in the fields such as 3D printing, building modelling, spacecraft design, cultural relic restoration and interior design. Besides, they also exist in a variety of forms, like point clouds, curved surfaces and voxels. Due to the limitation of current technology, 3D models will inevitably lead to model defects when scanning, and the corrupted parts may have irreversible effects on research. Thus, it's particularly important to repair the corrupted 3D models.

By far, image or 3D model repair, which is also named as image or 3D model inpainting, has entered the age of GAN methods from that of traditional geometric methods and neural network methods. As an unsupervised learning method, the GAN has the great ability for style transfer, super resolution, image inpainting, denoising and so on. Motivated by its feature learning ability, we proposed the 3D-DCGAN to repair incomplete 3D mesh models in this paper, which

mainly has three advantages compared to the methods proposed yet:

- 1) The 3D-DCGAN uses the Convolution Neural Network [30] to enhance the performance of the original GAN, so as to have a better feature extraction ability.
- 2) The local GAN generates the repairing region with sufficient details.
- 3) The global GAN generates a natural and smooth transition, and these two networks cooperate to obtain a repaired 3D model with higher quality.

The rest of the sections in this paper are organized as follows. Section II reviews the related work; section III introduces the pretreatment of training sets and the architecture of the 3D-DCGAN; section IV presents the results of our method and makes some comparison with other method; section V summarizes the pros and cons of the 3D-DCGAN.

## II. REVIEW OF RELATED WORK

### A. GEOMETRIC METHODS

A wide variety of methods for repairing 3D models have been proposed years ago. The polynomial function [1], the radial basis function [2], Laplacian filtering [3] and other interpolation algorithms carry out a certain degree of repairing,

The associate editor coordinating the review of this manuscript and approving it for publication was K. C. Santosh.

leaving abrupt transitions yet. In [4]–[6], people construct a base surface using triangulation, on which surface details are then added through continuous iteration. However, it merely repairs the grid but not the detailed features. Besides, dynamic programming was utilized and the augmented Lagrangian method was introduced to cope with the variation model to repair the hole [7], but it is of difficulty to recover the details of the hole region when too much information is missing. After the voxel model is obtained, [8] a symbolic distance function is adopted to construct a directed distance field that is extended to achieve the repair of the hole area [9], [10]. While the missing holes will be repaired by the construction of the octree reconstruction contour surface, detailed features of the repair part will be lost owing to the resampling surface.

### B. NEURAL NETWORK METHODS

The rise of neural networks has inspired the idea of using neural networks to solve real-world problems. In [11]–[14], everyone discusses the self-training and learning of the image to be repaired through the introduction of the neural network and extracts the features for completion, each of whom has obtained a relatively ideal effect. A novel network based on sparse coding and the noise reduction autoencoder deep network is put forward in [15], which has achieved the removal of the image overlay text and the repair of deleted areas. Still, it is not suitable for the removal of large-scale overlay areas. Notably, the fully convolutional neural network [16] of Contextual Attention [17] has achieved pretty good results on CelebA and CelebA-HQ face datasets.

### C. GAN METHODS

In 2014, the GAN proposed by Ian Goodfellow [18] learned the characteristics through the game. Specifically speaking, the generator and the discriminator within the network are against each other to learn features of the training data, after which people can barely differentiate whether the generated picture is real or fake are then generated. The proposal of the GAN provides a brand-new idea for image generation and also gives rise to a large scale of research on 2D and 3D image generation methods based on it.

Context Encoder [19], proposed in 2016, combines the architecture of the Encoder-Decoder and the GAN to predict the missing region, which opens the way of inpainting with the GAN. The Context Encoder works well for filling in missing regions, but not so well for local consistency. In [20], [21], high-resolution inpainting and on-demand learning have been improved in line with the Context Encoder to achieve much better results than previous methods. In [22], the local discriminator and the global discriminator are set to discriminate the completion part and the whole separately, which maintains the global consistency and optimizes the details. The finding of the coding in the training set closest to the corrupted image and the placement of the coding into a pre-trained GAN for semantic inpainting have ensured that the method's extracted information is no longer limited to a single image [23]. WGAN-GP [24] jointly incorporates the direct

and indirect measurements, which overcomes the limitation of incorporating only information from direct measurements to enhance the semantic inpainting performance. A hybrid architecture composed of the 3D-ED-GAN and the LRCN is proposed in [25], which accomplishes the transformation from low-resolution inpainting to high-resolution one. The 3D-RecGAN [26] obtains the mapping relationship between the voxelized partial 2.5D views and corresponding full 3D shapes by putting them into a GAN, and leverages the accuracy of the reconstructed model by introducing a generator, which loosely follows the idea of an autoencoder with U-net architecture and WGAN-GP as a discriminator. Although the algorithm reconstructs the structure of simple models well, it is hard to handle intricate models. The way the Point Encoder GAN [27] directly deals with point clouds using max-pooling and T-Nets [28] also gets a decent inpainting performance. In [29], the precision of depth inpainting is increased by using four kriging models on the basis of semi-variance models and color-similarity functions.

## III. MATERIALS AND METHODS

### A. 3D DATASET AND PRETREATMENT

The training data used in this paper comes from the ModelNet40 provided by Princeton University, which is a 3D model database consisting of 40 classes with a total amount of 12431 3D models. In the experiment, we combine the training set and the test set in the original class into one dataset to augment the number of training data volume as well as extracted features. (Our experiments are all tested in Ubuntu 18.04.4 with Python 3.7.3, Tensorflow 2.0 and one Nvidia Tesla P100 GPU.) Then, off format files are transferred into .stl format files, after which the attitude is adjusted so that they are placed at a uniform angle. After that, we voxelized these 3D models with resolution  $80 \times 80 \times 80$ . Finally, the matrices of those voxelized 3D models are saved as .mat format files.

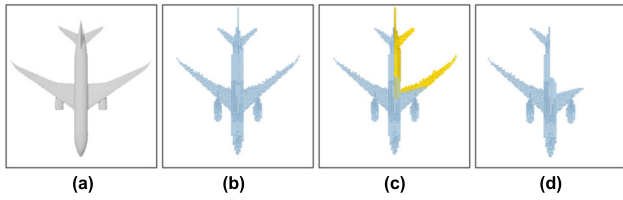
In order to acquire the incomplete 3D model to be repaired, we crop a certain part of the 3D model randomly chosen from the training dataset.

Fig. 1 shows the voxelization of an airplane and the procedure of its transformation into a repairing model. As shown in Fig. 1 c, the cropped area in our work is the upper right quarter of its own (the yellow part). When this section is removed, the remainder of this model is what we called the repairing model.

Apart from an uncomplete 3D model called the repairing model, we also get a mask used to restrict the region generated by the GAN after cropping. In terms of the mask, it can also be used to re-align the repair region with the repairing model when the generator produces a model of the corresponding region, and the aligned model is exactly what the algorithm outputs in the end.

### B. ARCHITECTURE OF 3D-DCGAN

The 3D-DCGAN proposed in this paper is comprised of two GANs, a local GAN and a global GAN [22]. While the



**FIGURE 1.** Model voxelization and crop. (a) original airplane. (b) voxelization. (c) cropped region. (d) repairing model.

primary purpose of the local GAN is to generate the repaired region, this region can invariably occur a very unnatural transition or even fault after merging the repaired region and the repairing model. To cope with this issue, a global GAN working on the top of the local GAN that ensures a global judgement of the combined model made by it is introduced. In this regard, it optimizes the generator to generate a better repaired region with great details and natural transitions. Besides, both the local GAN and the global GAN are deep convolutional networks, and so the excellent extraction feature of the DCGAN [31] having been discussed in [32]–[34] is preserved in our networks.

Fig. 2 illustrates the process of the 3D-DCGAN for 3D model repair. First, the generator and the local discriminator consist of the local GAN, which is trained to generate high-quality repaired models. Then, combine the repaired model with the initial repairing 3D model. Later, put the combined 3D model into the global GAN composed of a global discriminator and the generator in the local GAN.

Since both the local GAN and the GAN share one generator in the training process, it is plausible to learn both local and global features so as to enhance the final output of the network.

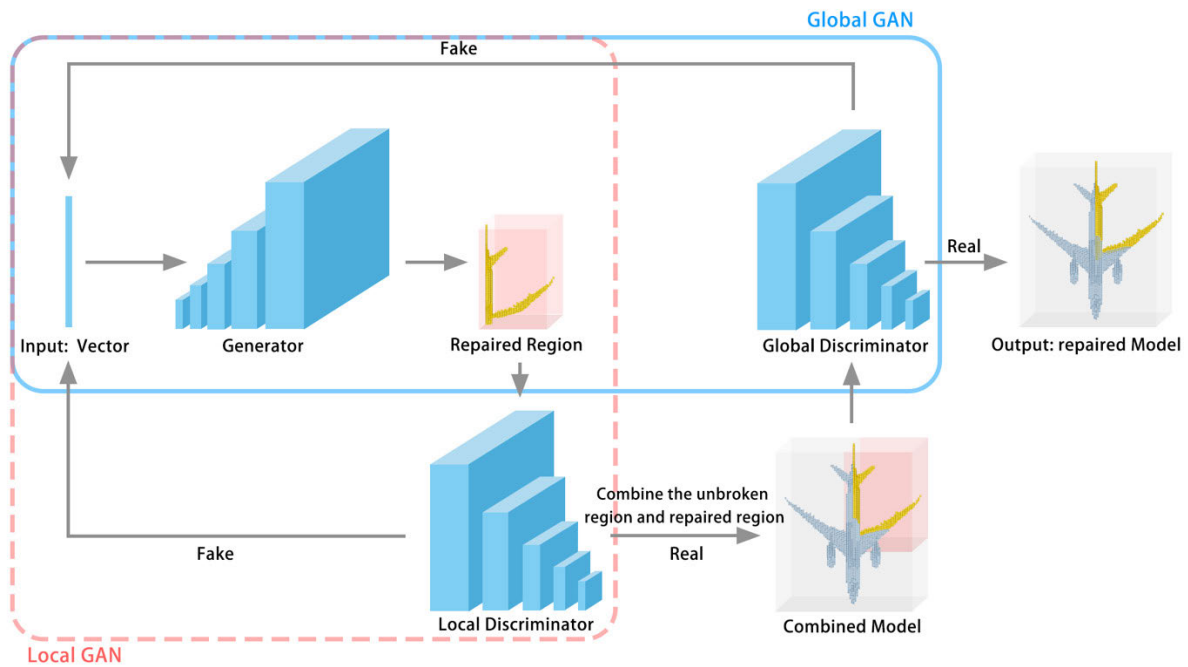
As a repairing module, the local GAN itself is a DCGAN that processes the 3D model data and generates “Fake” 3D images, and thus the GAN’s calculating equation still covers this (1).

$$\begin{aligned} & \min_G \max_{D_{Local}} V(D_{Local}, G) \\ & = \mathbb{E}_{x \sim p_{data(x)}} [\log D_{Local}(x)] \\ & \quad + \mathbb{E}_{z \sim p_z(z)} [\log (1 - D_{Local}(G(z)))] \end{aligned} \quad (1)$$

As in (1),  $x$  denotes the real 3D model;  $z$  represents the noise of the generator’s input;  $G(z)$  means the 3D model generated by the generator.  $D_{Local}(x)$  denotes the probability of the local discriminator to distinguish whether the 3D model is real or not.

Generators and discriminators adopted in the 3D-DCGAN and the DCGAN have the identical structure. Nevertheless, this paper adopts 3D model data all the way through the entire process, and so parameters within the network should be adjusted from the 2D convolutional kernel to the 3D one. The 3D convolution equation is shown in (2).

$$x = picture(i, j, k) = \sum_{a=0}^{layers-1} \sum_{b=0}^{rows-1} \sum_{c=0}^{columns-1} f(a, b, c) \times h(i - a, j - b, k - c) \quad (2)$$



**FIGURE 2.** Structure of 3D-DCGAN.

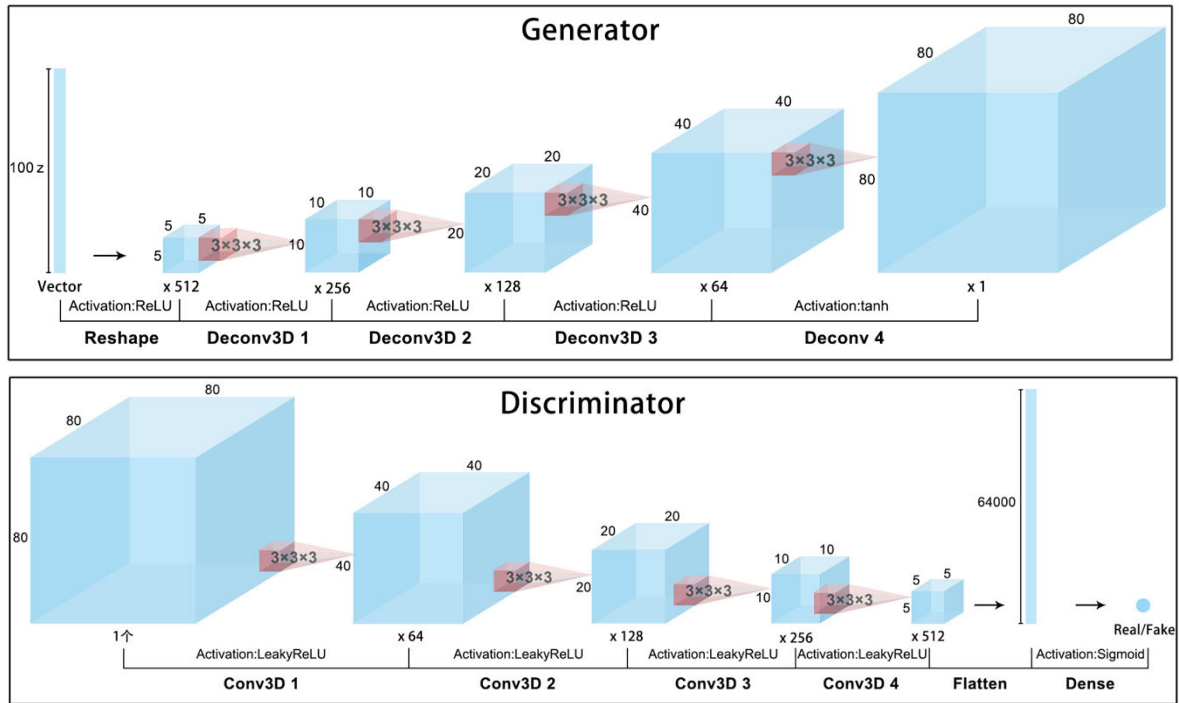


FIGURE 3. Architecture of Generator and Discriminator of 3D Deep Convolutional GAN.

After adapting, (1) becomes what is shown in (3).

$$\begin{aligned}
 & \min_G \max_{D_{Local}} V(D_{Local}, G) \\
 &= \mathbb{E}_{x \sim p_{data(x)}} \left[ \log D_{Local} \left( \sum_{a=0}^{layers-1} \sum_{b=0}^{rows-1} \sum_{c=0}^{columns-1} f(a, b, c) \right. \right. \\
 & \quad \left. \left. \times h(i-a, j-b, k-c) \right) \right] \\
 &+ \mathbb{E}_{z \sim p_{z(z)}} [\log(1 - D_{Local}(G(z)))] \tag{3}
 \end{aligned}$$

In the global GAN, due to the introduction of the original repairing model, Eq. (3) is no longer applicable, and the new one is shown in Eq. (4).

$$\begin{aligned}
 & \min_G \max_{D_{Global}} V(D_{Global}, G) \\
 &= \mathbb{E}_{x \sim p_{data(x)}} \left[ \log D_{Global} \left( \sum_{a=0}^{layers-1} \sum_{b=0}^{rows-1} \sum_{c=0}^{columns-1} f(a, b, c) \right. \right. \\
 & \quad \left. \left. \times h(i-a, j-b, k-c) \right) \right] \\
 &+ \mathbb{E}_{z \sim p_{z(z)}} [\log(1 - D_{Global}(G(z) + x_n \odot (1 - Mask_n)))] \tag{4}
 \end{aligned}$$

As in (3) and (4), the distinction between these two lies in the introduction of  $x_n \odot (1 - Mask_n)$  in  $D_{Global}(G(z) + x_n \odot (1 - Mask_n))$ , which represents the repairing model during merging.  $x_n$  denotes any voxelized

model in the dataset, and  $Mask_n$  is the mask of the repairing region in the model  $n$ . The reverse mask is taken through  $1 - Mask_n$  and it's easy to get the unbroken region by multiplying the reverse mask with the original voxelized model.

Considering the importance of having a generator that knows the repairing region well, the generator in the local GAN and the global GAN is the same one, which helps get lots of local and global features to merge well.

### C. THE GENERATOR AND DISCRIMINATOR OF 3D-DCGAN

The generators and discriminators in this paper adopt 3D convolutional kernels, enabling both of them to handle 3D models.

In the generator (as shown in Fig.3 Generator), we use 100-dimensional random noise as the initial input, and then transfer the data into a  $[5 \times 5 \times 5 \times 512]$  tensor through one fully connected layer, which is also known as reshaping. The tensor is then calculated by four 3D deconvolutional layers and finally output as a  $[80 \times 80 \times 80 \times 1]$  tensor (which is also the final repaired model). In the first layer, the layer is made up of 256 convolution kernels; the stride is  $2 \times 2 \times 2$ ; the activation function is ReLU. The second layer has 128 convolution kernels; the stride and the activation function are the same as those in the previous layer. The third layer highly resembles layer 2, but the convolution kernel number is set to 64. The final layer has varied a lot. That is to say, only one convolution kernel is set up and the activation function becomes tanh. The output of the generator is a  $[80 \times 80 \times 80 \times 1]$  sparse matrix.

The discriminator structure used in this paper has an inverse operational relationship with the generator structure (as shown in Fig. 3 Discriminator). The input of the discriminator is a  $[80 \times 80 \times 80 \times 1]$  sparse matrix (the voxelized image), going through 4 convolutional layers before outputting true or false as the final estimation. The layer 1 to layer 4 in the discriminator all have LeakyReLU as the activation function, with the strides of  $2 \times 2 \times 2$ . Moreover, the differences among them are the number and the size of convolution kernels. Finally, the Sigmoid cross-entropy function is set to determine the truth or falsity of the input image.

The parameters of the generator and the discriminator in our work are listed in Table 1.

#### D. ALGORITHM DESCRIPTION

The training procedure of the 3D-DCGAN is listed in Algorithm I, which presents the entire 3D-DCGAN training process. We take the voxelized 3D model sets and a voxelized 3D model with the missing region, which are the same class as training and inpainting inputs. Every training epoch incorporates local GAN and global GAN training. While the former takes inputs mentioned before as input, the latter takes output of the local GAN as its input. Therefore, the detailed features learned by the local GAN are conveyed to the global GAN. The combination of the repaired region and the unbroken region helps the global GAN obtain the relationship between local and global regions. From the final result, which is also the output of the global GAN, the relationship signifies a better transition.

### IV. RESULTS AND DISCUSSION

#### A. EXPERIMENTAL RESULTS

Fig. 4 shows the results of the repaired models using disparate GANs and our method proposed in this paper. In Fig. 4 c, due to the lack of feature learning approaches for the original GAN [18], it is impossible to effectively get traits of the missing region, which brings about very unsatisfactory results. Fig. 4 d shows the result of the BiGAN [35], in which every layer is fully connected and requires a lot of calculation. As a result, we lower the resolution, and the results turn out to be the worst among all the algorithms. In Fig. 4 e, the DCGAN [31] does a good job of repairing the missing regions, but there is still much room for improvement in the fusion of the repaired region and the model to be repaired. More precisely, there are significant faults at the articulation and a small amount of noise. In Fig. 4 f, the method proposed in this paper has significantly improved in junctions, transitions and noise control compared with the GAN, the BiGAN and the DCGAN.

In Fig. 5, all of the repaired bed models have been zoomed in. As the original GAN is dramatically affected by the missing of feature extraction, really bad results are generated. Furthermore, the BiGAN gets even worse results than the GAN considering the large amount of calculation that leads to the reduction of resolution. Compared to the former two,

---

#### Algorithm 1 The Procedure of 3D-DCGAN

---

**Input:** A voxelized 3D model training set and a voxelized 3D model with missing region

**Output:** A voxelized 3D model with missing region inpainted

**Algorithm:** Training procedure of 3D-DCGAN

---

**for** number of training epochs **do**:

**for** number of training iterations **do**:

**Train the Local GAN:**

            •Train *Local Discriminator* with batch-size voxelized 3D models.

            •Train *Local Discriminator* with generated 3D models:

                Generator produces batch-size 3D models by noise.

                Train the *Local Discriminator* with the generated 3D models for one iteration.

**end for**

    •Combine *Local Discriminator* and *Generator* as *Local GAN*, then train one epoch.

**for** number of training iterations **do**:

**Train the Global GAN:**

            •Generate *repaired model*:

                The generator produces one batch-size repaired region.

**Combine the repaired region with unbroken region as the repaired model using mask  $Mask_n$ .**

                •Train *Global Discriminator* with the batch-size voxelized 3D models.

                •Train *Global Discriminator* with the batch-size repaired model.

**end for**

        •Combine *Global Discriminator* and *Generator* as *Global GAN*, then train one epoch.

**end for**

---

the DCGAN with the convolutional layers generates much better results. The results have been largely restored to the overall character of the missing region, but there are still articulation jumps and detail clutter affecting the results, which leaves so much room for improvement. In our work, the algorithm makes a near-perfect restoration of the *bed*, and even the pillow on the bed has been perfectly reproduced. Besides, the jumps in the articulation have been addressed properly.

#### B. RESULTS EVALUATION

To assess the quality of the 3D complementary model, we use two evaluation metrics most commonly used in 2D images: the mean square error (MSE) function and the peak signal-to-noise ratio (PSNR) function.

The MSE function reflects the difference between original images and processed images, and the value of the mean



TABLE 1. Parameters list of the generator (top) and the discriminator (bottom).

| Layer        | Inputs       | Kernel        | Stride | Activation | Outputs      |
|--------------|--------------|---------------|--------|------------|--------------|
| Vectors      | —            | —             | —      | —          | 100          |
| Reshape      | 100          | —             | —      | ReLU       | 5×5×5×512    |
| Deconv3D 1st | 5×5×5×512    | 256@3×3×3×512 | 2×2×2  | ReLU       | 10×10×10×256 |
| Deconv3D 2nd | 10×10×10×256 | 128@3×3×3×256 | 2×2×2  | ReLU       | 20×20×20×128 |
| Deconv3D 3rd | 20×20×20×128 | 64@3×3×3×128  | 2×2×2  | ReLU       | 40×40×40×64  |
| Deconv3D 4th | 40×40×40×64  | 1@3×3×3×64    | 2×2×2  | tanh       | 80×80×80×1   |
| Conv3D 1st   | 80×80×80×1   | 64@3×3×3×1    | 2×2×2  | LeakyReLU  | 40×40×40×64  |
| Conv3D 2nd   | 40×40×40×64  | 128@3×3×3×64  | 2×2×2  | LeakyReLU  | 20×20×20×128 |
| Conv3D 3rd   | 20×20×20×128 | 256@3×3×3×128 | 2×2×2  | LeakyReLU  | 10×10×10×256 |
| Conv3D 4th   | 10×10×10×256 | 512@3×3×3×256 | 2×2×2  | LeakyReLU  | 5×5×5×512    |
| Flatten      | 5×5×5×512    | —             | —      | —          | 64000        |
| Dense        | 64000        | —             | —      | Sigmoid    | 1            |

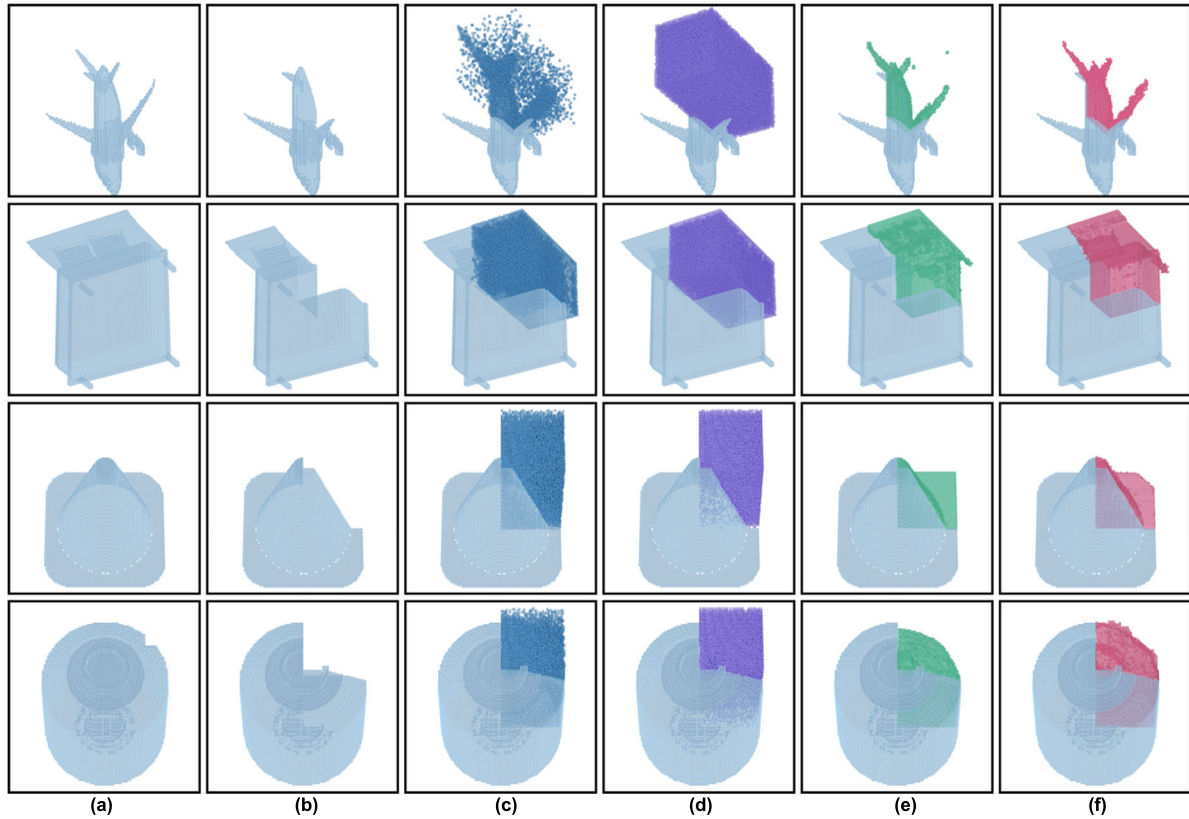


FIGURE 4. Comparison with state-of-the-art methods to repairing the repairing models. (a) Original Model. (b) Repairing Model. (c) GAN [18]. (d) BiGAN [35]. (e) DCGAN [31]. (f) Ours.

square error is used to determine the level of distortion. Because the data in this paper is 3D models, the equation needs to be modified as the image converted from 2D to 3D, and the modified equation is shown in (5).

$$MSE = \frac{1}{N_{Rows} \times N_{Columns} \times N_{Layers}}$$

$$\times \sum_{x=0}^{N_{Rows}-1} \sum_{y=0}^{N_{Columns}-1} \sum_{z=0}^{N_{Layers}-1} (f(x, y, z) - \hat{f}(x, y, z))^2 \quad (5)$$

As in (5),  $N_{Rows}$ ,  $N_{Columns}$  and  $N_{Layers}$  represent the number of pixels of all three dimensions, respectively. Besides,  $f(x, y, z)$  and  $\hat{f}(x, y, z)$  denote the original 3D model and the repaired 3D model, respectively.

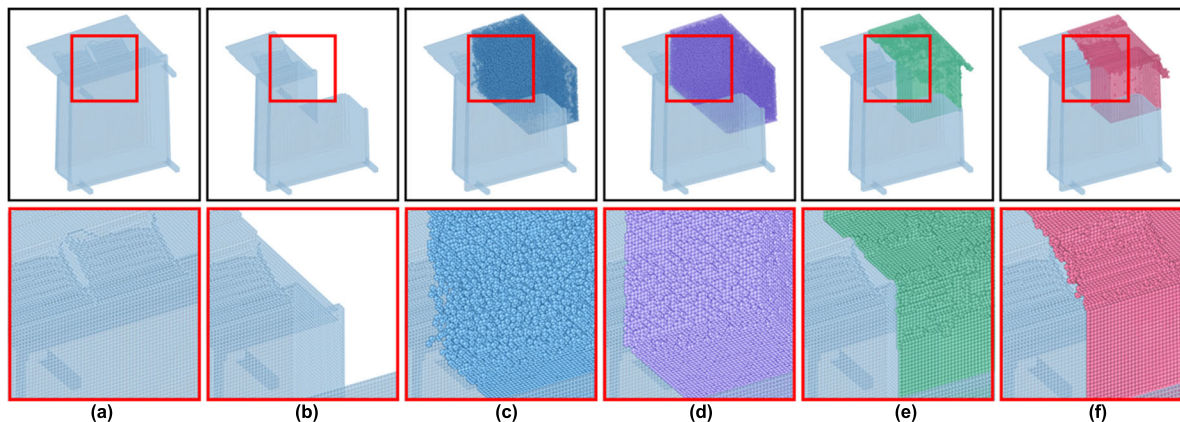


FIGURE 5. Zoom-in on the images of repaired bed models. (a) Original Model. (b) Repairing Model. (c) GAN. (d) BiGAN. (e) DCGAN. (f) Ours.

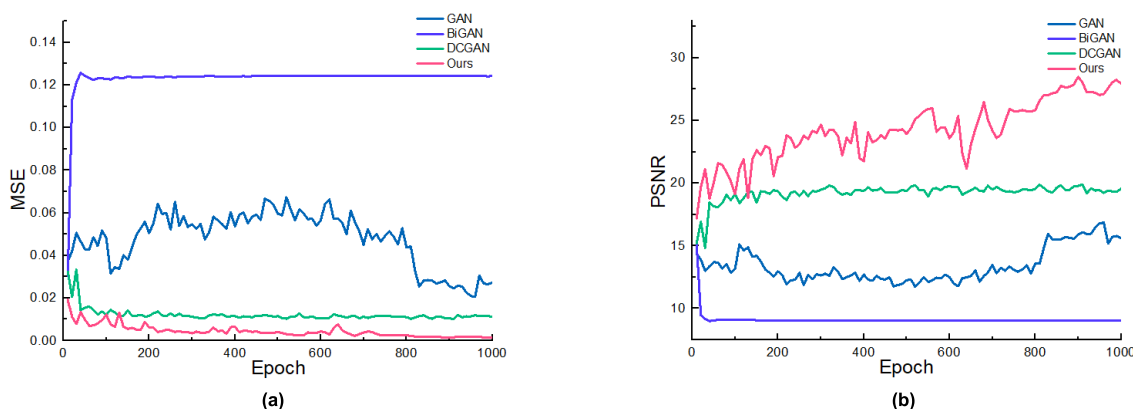


FIGURE 6. Repaired airplane model quality evaluation. (a) MSE. (b) PSNR.

The peak signal-to-noise ratio function is used as a measure to evaluate the quality of reconstruction in image processing with (6).

$$PSNR = 10 \times \log_{10} \left( \frac{(I_{max})^2}{MSE} \right) = 20 \times \log_{10} \left( \frac{I_{max}}{\sqrt{MSE}} \right) \quad (6)$$

Since the 3D model in this paper is a (0,1) sparse matrix, the maximum possible pixel value of the image represented by  $I_{max}$  is taken as 1. As an indicator to reflect the degree of distortion, the smaller the value of the MSE is, the smaller the distortion is. On the contrary, the larger the value of the PSNR is, the better the image is.

Fig. 6 shows the MSE and PSNR value curves of the repaired *airplane* models of dissimilar algorithms, which conveys the same information as in Fig. 5. In all algorithms, the repairing capability of the GAN and the BiGAN is at the bottom, and that of the DCGAN is in the middle. The results of our proposed method get the best grades in both evaluation criteria.

Table 2 shows the comparison of the repairing degree among the GAN, the BiGAN, the DCGAN and our method, which is determined by the Hausdorff Distance between the

TABLE 2. Hausdorff Distance of GANs.

| Model    | GAN [18] | BiGAN [35] | DCGAN [31] | Ours   |
|----------|----------|------------|------------|--------|
| Airplane | 48.3839  | 48.7545    | 23.0000    | 5.8310 |
| Bed      | 33.3766  | 33.3766    | 19.0000    | 3.0000 |
| Cone     | 46.0217  | 46.3357    | 13.6014    | 1.7321 |
| Bottle   | 22.1133  | 23.6008    | 8.2462     | 5.0000 |

repaired model and the original model. It can be seen from the table that the BiGAN has the highest Hausdorff Distance value among all these methods; the DCGAN is somewhere between GAN and our method; our method gets the lowest value. From the GAN to the method proposed by us, we have got decent progress in both value and effect, signifying that the results obtained from our work produce both fewer jumps and much better transitions.

## V. CONCLUSION

The 3D-DCGAN proposed in this paper, which is a repairing algorithm, allows the repaired region to maintain the

detailed characteristics of its region through the local GAN and the global GAN. Besides, the repaired region can also be integrated into the entire model, improving the quality of the repaired model efficaciously. Compared with the existing algorithm, it can not only be applied to small holes but also large holes or even multiple holes.

With the increase of demand for higher image resolution, there will be an exponential increase in the computation of the algorithm proposed in this paper. In the following work, we will mainly focus on diminishing the redundant computation among layers in the network so that the algorithm can be applied to higher resolution. Also, the proposed algorithm is effective for the repairing models with low complexity structure. However, as the complexity of the model structure increases, the effectiveness of the repairing model lowers, which makes a deeper study of such cases inevitable.

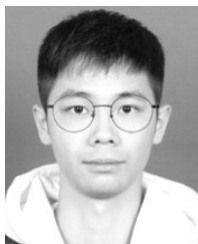
## REFERENCES

- [1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.
- [2] J. Branch, F. Prieto, and P. Boulanger, "Automatic hole-filling of triangular meshes using local radial basis function," in *Proc. 3rd Int. Symp. 3D Data Process., Visualizat., Transmiss. (3DPVT)*, Jun. 2006, pp. 727–734.
- [3] M. Wei, J. Wu, and M. Pang, "An integrated approach to filling holes in meshes," in *Proc. Int. Conf. Artif. Intell. Comput. Intell.*, Oct. 2010, pp. 306–310.
- [4] P. Liepa, "Filling holes in meshes," in *Proc. Eurograph/ACM SIGGRAPH Symp. Geometry Process.*, Jun. 2003, pp. 200–205.
- [5] Z. LIU, Y. Wang, and L. Bai, "Detail-preserving hole-filling for complex 3D models," *J. Comput. Aided Design Comput. Graph.*, vol. 28, no. 12, pp. 2052–2059, Dec. 2016.
- [6] P. Wen, Y. Lei, and M. Sun, "Defective hole identification and hole-filling for 3d reconstruction mesh models," *Appl. Res. Comput.*, vol. 37, no. 04, pp. 1234–1238, Jan. 2020.
- [7] Y. Zhai and X. Wu, "Mesh inpainting using extended total variation regularization," *J. Comput. Aided Design Comput. Graph.*, vol. 31, no. 11, pp. 1889–1897, Nov. 2019.
- [8] J. Davis, S. R. Marschner, M. Garr, and M. Levoy, "Filling holes in complex surfaces using volumetric diffusion," in *Proc. 1st Int. Symp. 3D Data Process. Visualizat. Transmiss.*, 2002, pp. 428–441.
- [9] T. Ju, "Robust repair of polygonal models," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 888–895, Aug. 2004.
- [10] S. Bischoff and L. Kobbelt, "Structure preserving CAD model repair," in *Comput. Graphics Forum*, vol. 24, no. 3, Amsterdam, The Netherlands: North Holland, 1982, pp. 524–536.
- [11] N. Cai, Z. Su, Z. Lin, H. Wang, Z. Yang, and B. W.-K. Ling, "Blind inpainting using the fully convolutional neural network," *Vis. Comput.*, vol. 33, no. 2, pp. 249–261, Feb. 2017.
- [12] A. Fawzi, H. Samulowitz, D. Turaga, and P. Frossard, "Image inpainting through neural networks hallucinations," in *Proc. IEEE 12th Image, Video, Multidimensional Signal Process. Workshop (IVMSP)*, Jul. 2016, pp. 1–5.
- [13] R. Köhler, C. Schuler, B. Schölkopf, and S. Harmeling, "Mask-specific inpainting with deep neural networks," in *Proc. German Conf. Pattern Recognit.* Springer, Oct. 2014, pp. 523–534.
- [14] Y. Song, C. Yang, Z. Lin, X. Liu, Q. Huang, H. Li, and C. C. J. Kuo, "Contextual-based image inpainting: Infer, match, and translate," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 3–19.
- [15] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 341–349.
- [16] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.
- [17] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5505–5514.
- [18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
- [19] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2536–2544.
- [20] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li, "High-resolution image inpainting using multi-scale neural patch synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6721–6729.
- [21] R. Gao and K. Grauman, "On-demand learning for deep image restoration," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1086–1095.
- [22] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," *ACM Trans. Graph.*, vol. 36, no. 4, p. 107, 2017.
- [23] R. A. Yeh, C. Chen, T. Y. Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do, "Semantic image inpainting with deep generative models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5485–5493.
- [24] Q. Zheng, L. Zeng, and G. E. Karniadakis, "Physics-informed semantic inpainting: Application to geostatistical modeling," *J. Comput. Phys.*, vol. 419, Oct. 2020, Art. no. 109676.
- [25] W. Wang, Q. Huang, S. You, C. Yang, and U. Neumann, "Shape inpainting using 3D generative adversarial network and recurrent convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2298–2306.
- [26] B. Yang, H. Wen, S. Wang, R. Clark, A. Markham, and N. Trigoni, "3D object reconstruction from a single depth view with adversarial learning," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 679–688.
- [27] Y. Yu, Z. Huang, F. Li, H. Zhang, and X. Le, "Point encoder GAN: A deep learning model for 3D point cloud inpainting," *Neurocomputing*, vol. 384, pp. 192–199, Apr. 2020.
- [28] J. Kossaifi, A. Bulat, G. Tzimiropoulos, and M. Pantic, "T-Net: Parametrizing fully convolutional nets with a single high-order tensor," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7822–7831.
- [29] Y. Keamane, A. Heednacram, and P. Youngkong, "Implementation of four kriging models for depth inpainting," *ICT Express*, vol. 6, no. 3, pp. 209–213, Sep. 2020.
- [30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.
- [31] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*. [Online]. Available: <http://arxiv.org/abs/1511.06434>
- [32] S. Liu, M. Yu, M. Li, and Q. Xu, "The research of virtual face based on deep convolutional generative adversarial networks using TensorFlow," *Phys. A, Stat. Mech. Appl.*, vol. 521, pp. 667–680, May 2019.
- [33] B. Kang and J. Choe, "Uncertainty quantification of channel reservoirs assisted by cluster analysis and deep convolutional generative adversarial networks," *J. Petroleum Sci. Eng.*, vol. 187, Apr. 2020, Art. no. 106742.
- [34] M. Cheng, F. Fang, C. C. Pain, and I. M. Navon, "Data-driven modelling of nonlinear spatio-temporal fluid flows using a deep convolutional generative adversarial network," *Comput. Methods Appl. Mech. Eng.*, vol. 365, Jun. 2020, Art. no. 113000.
- [35] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," 2016, *arXiv:1605.09782*. [Online]. Available: <http://arxiv.org/abs/1605.09782>



**XINYING WANG** received the B.S., M.S., and Ph.D. degrees in computer science from Jilin University, in 2001, 2005, and 2009, respectively. She is currently an Associate Professor with the College of Computer Science and Engineering, Changchun University of Technology. Her current research interests include computer vision, deep learning, and data mining.





**DIKAI XU** received the B.S. degree from the College of Computer Science and Technology, Ningbo University of Technology, in 2018. He is currently pursuing the M.E. degree in College of Computer Science and Engineering, Changchun University of Technology. His current research interests include deep learning and 3D image processing.



**FANGMING GU** received the Ph.D. degree from Jilin University, China, in 2010. He is currently a Lecturer with the College of Computer Science and Technology, Jilin University. His current research interests include artificial intelligence, data mining, deep learning, and image processing.

...