http://www.ftsm.ukm.my/apjitm-2021-1002-07
Asia-Pacific Journal of Information Technology and Multimedia
Jurnal Teknologi Maklumat dan Multimedia Asia-Pasifik
Vol. 10 No. 2, December 2021: 95 - 109
e-ISSN: 2289-2192

Received: 31 March 2021
Accepted: 23 August 2021
Published: 1 December 2021

3DINFOGAN: 3D MODELS' RECONSTRUCTION IN INFOGANS

DU CHUNQI
SHINOBU HASEGAWA

ABSTRACT

In computer vision and computer graphics, 3D reconstruction is the process of capturing real objects' shapes and appearances. 3D models always can be constructed by active methods which use high-quality scanner equipment, or passive methods that learn from the dataset. However, both of these two methods only aimed to construct the 3D models, without showing what element affects the generation of 3D models. Therefore, the goal of this research is to apply deep learning to automatically generating 3D models, and finding the latent variables which affect the reconstructing process. The existing research GANs can be trained in little data with two networks called Generator and Discriminator, respectively. Generator can produce synthetic data, and Discriminator can discriminate between the generator's output and real data. The existing research shows that InFoGAN can maximize the mutual information between latent variables and observation. In our approach, we will generate the 3D models based on InFoGAN and design two constraints, shape-constraint and parameters-constraint, respectively. Shape-constraint utilizes the data augmentation method to limit the synthetic data generated in the models' profiles. At the same time, we also try to employ parameters-constraint to find the 3D models' relationship corresponding to the latent variables. Furthermore, our approach will be a challenge in the architecture of generating 3D models built on InFoGAN. Finally, in the process of generation, we might discover the contribution of the latent variables influencing the 3D models to the whole network.

Keywords**:** 3D Reconstruction, Deep Learning, Latent variables

INTRODUCTION

Image-based 3D reconstruction technology involves many popular fields, such as computer image processing, computer graphics(Li, R., et al. 2011), computer vision(Alahari, K., et al. 2013), and computer-aided design. At present, image-based 3D reconstruction technology has become a hot field with great potential and applied in many aspects(Sra, M., et al. 2016), such as e-commerce, space flight, remote sensing surveying and mapping, virtual museums(Aoki, H., et al. 2008), and other high-tech fields. Before the rise of deep neural networks, our more mature methods for 3D reconstruction of objects or scenes include single image reconstruction using the principle of geometric projection, 3D reconstruction using binocular stereo vision, and 3D reconstruction based on depth images. Although these methods have achieved good reconstruction results in certain aspects (Aditya, T. S. 2010), they still have disadvantages such as a high amount of calculation, poor reconstruction effect, high price, and low degree of automation(Sinha, A., et al. 2016).

Nowadays, the high development of deep learning enables many researchers to carry out reconstruction experiments from 2D images or 3D models(Sinha, A., et al. 2016)(Gao, L., et al. 2019). Image-based 3D reconstruction technology has the advantages of fast, simple, and realistic and can better realize the virtualization of objects. At present, most researchers dedicated to improving the accuracy and details of

3D model reconstruction(Chen, Z., et al. 2019). The main problem of this technology is how to recover three-dimensional information based on disturbed or incomplete two-dimensional information, which is also the main problem of computer vision. As we know, accuracy and details are significant for the reconstruction of 3D models. However, in the process of 3D model reconstruction, whether the latent variables corresponding to the characteristic semantics can be found from the latent space to explain the generation of 3D objects may increase the convenience of 3D model reconstruction. In-depth research on image-based reconstruction, technology can promote the understanding and research of 3D reconstruction and promote the development of related disciplines(Chen, Z., et al. 2019).

The goal of this research is to apply deep learning to reconstruct the 3-dimensional models from the probabilistic latent space with 3D chairs multi-view images datasets and find out the corresponding controllable features that can represent the 3-dimensional model by seeking the potential representations in the 3-dimensional space. To achieve this goal, we started by solving there following two questions:

1. How to reconstruct the 3D chair models from multi-view images?
2. How to manipulate the models to find effective latent variables?

Our main idea is to transform such multi-view 3D chair images with different rotation and width to 3-dimensional models from the probabilistic latent space as shown in Figure 1.

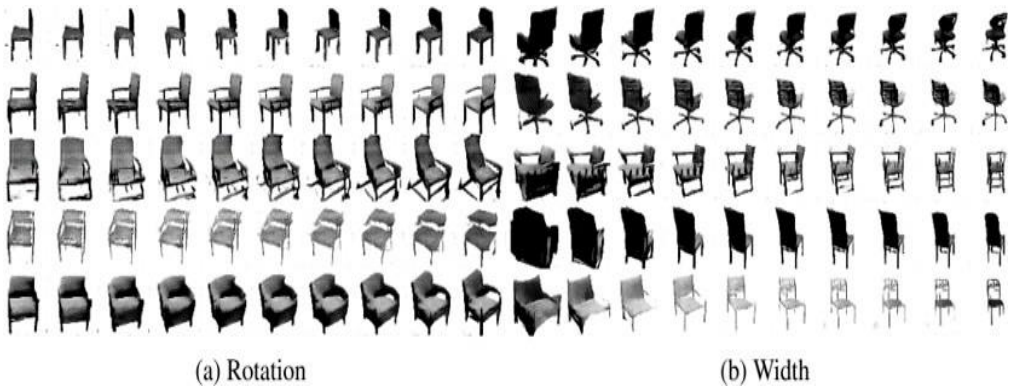

FIGURE 1. Multi-View 3D Chair Images

RELATED WORKS

3D RECONSTRUCTION

Three-dimensional reconstruction is a reverse process of three-dimensional object or scene image description, which can restore a three-dimensional object or scene from two- dimensional image(Malik, A., et al. 2020). Image-based 3D reconstruction is a method to extract the 3D depth information from several scenes and object images and reconstruct the 3D model of the object or scenes with a strong sense of reality according to the obtained 3D depth information. With the latest development of artificial intelligence and computer hardware, deep learning method increasingly used in the 3D model reconstruction. In 3D object reconstruction, we can use voxel grids (Girdhar, R., et al. 2016)(Liao, Y., et al. 2018), octrees (Tatarchenko, M., et al. 2017), point cloud (Gadelha, M., et al. 2018)(Wang, L., et al. 2020), and other forms (Henderson, P., et al. 2020)to

represent the 3D shape representation of deep learning shapes. Girdhar et al.(Girdhar, R., et al. 2016)propose an embedding space that can predict voxel from 2D images and retrieve 3D models. Wu et al. (Wu, J., et al. 2016) extended GANs from images to voxels and generate three-dimensional objects represented by voxels in a three-dimensional probabilistic space.

Although there are currently many different deep learning models and 3D model representations are used in 3D model reconstruction tasks. But most of them focused on the accuracy and smoothness of the generated model shape, without an in-depth discussion on the impact of latent code on the generative deep model during the 3D model generation process.

GANs

Goodfellow et al. proposed Generative Adversarial Networks (GANs), a framework for estimating deep generative models via minimax games. It learns a generative model G (Generator) that captures the data distribution and trains the G by playing against a discriminative model D (Discriminator) that estimates the probability that a sample came from the training data(Goodfellow ,et al. 2016).

Generative Adversarial Networks (GANs) (Goodfellow ,et al. 2016) is one of the most promising deep models for unsupervised learning in complex distribution by mapping from a latent distribution to the real data via adversarial learning between a generative model and a discriminative model(Radford, A., et al. 2015). After learning such a non-linear mapping, GAN can produce photo-realistic images from randomly sampled latent variables(Brock, A., et al. 2018).

However, to control the image generation, it is necessary to understand the randomly sampled latent variables corresponding to which features. For example, images of handwritten characters are defined by many properties such as character type, orientation, width, curvature, and so on. Because the GANs uses a simple continuous input noise vector, and there are no restrictions on how the Generator can use this noise. Therefore, the noise may be used by the Generator in a highly entangled way, resulting in the single dimension of the noise vector does not correspond to the semantic features of the data (Brock, A., et al. 2018).

INFOGANS

Information Maximizing Generative Adversarial Networks (InfoGANs) have recently shown to learn the understanding of latent variables (Brock, A., et al. 2018). Chen et al. (Chen, X., et al. 2016) proposed a generative adversarial network called InFoGANs which can maximize the mutual information between a small subset of the latent variables and the observation. InFoGANs is an information-theoretic extension to the generative adversarial network that can learn disentangled representations in a completely unsupervised manner. InFoGANs fixes a part of the latent vectors and divides them into categorical codes and continuous codes. Categorical codes (each categorical code contains N discrete codes) can model discrete variation in data, and continuous codes can capture continuous variations.

METHODOLOGY

PROPOSED METHOD

Figure 2 shows an overview of our research design and methodology. In this chapter, we introduce our necessary equipment and model for 3D object generation. The first is the mapping from 3D chair multi-view images to the 3-dimensional space. Then, we discuss how to construct the framework of the 3D model generative adversarial network by using volume convolution network and the recent advance of generative countermeasure network.

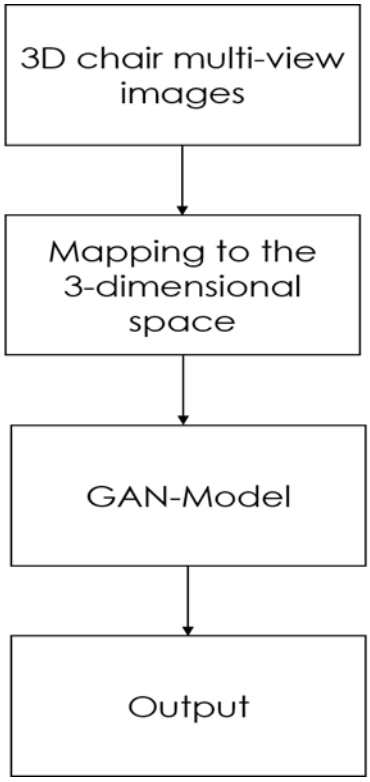

FIGURE 2. Overview Of Method

REFERENCE ARCHITECTURE

INPUT

InFoGANs modelled the latent codes with three categorical codes (each categorical code contains ten discrete codes), which can model discrete variation in data, and two continuous codes that can capture continuous variations. Compare with the InFoGANs, the same datasets and same parameters setting were used in this research, as shown in Figure3.

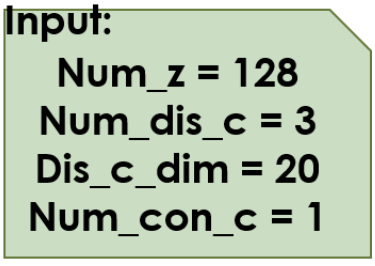

FIGURE 3. Original Parameters

1. Num_z means Noise variables;
2. Num_dis_c means Number of discrete latent codes;
3. Dis_c_dim means Dimension of each discrete codes;
4. Num_con_c means Number of continuous latent codes.

For the comparison with InFoGANs, the same datasets and same parameters setting were used in this research.

GENERATOR

Generator G, mapping a randomly sampled 200-dimensional vector noise z with latent c vector from the probability latent space to a 64*64*64 cube that represents the object G (z, c) in 3D voxel space. The Generator consists of the five 3-dimensional volumetric convolutional layers with batch normalization and four middle ReLU layers added along with a final Sigmoid layer.

DISCRIMINATOR

Following Chen et al. (Chen, X., et al. 2016) the whole Discriminator D is composed of 'discriminator' and D function, i.e., Discriminator = ' discriminator ' + D_Function. The part of 'discriminator' just has feature extraction and sharing function, the last layer of discrimination is in the D_function, so the combination of the two is the complete Discriminator.

The extracted features of the 'discriminator' input the D function, and the output of the D function shows whether the 3D objects input the Discriminator D is real or not. We experimented and found that the running time of four volumetric convolutional layers is faster than the running time of the original five convolutional layers in the 'discriminator' part. Although it lost a little bit of detailed feature extraction, it did not affect the final results. The 'discriminator' constitutes almost the same as the Generator, except for the number of layers and Leaky ReLU layers instead of ReLU layers, as shown in Figure 4.

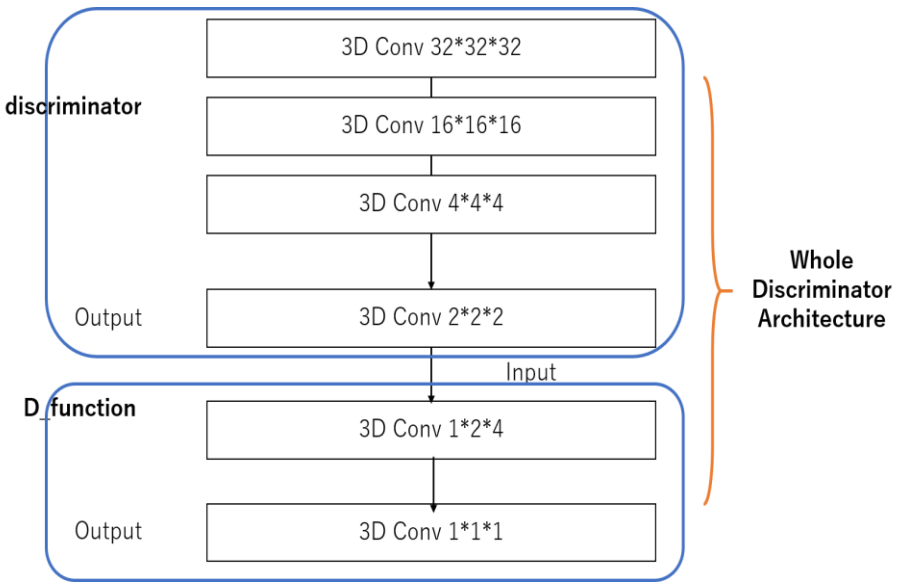

FIGURE 4. Architecture Of Discriminator

Q_FUNCTION

Q_function shares all volumetric convolutional layers in the discriminator with the D function. In the Q function, we use the natural choice of softmax nonlinearity to represent the distribution of categorical latent codes in Q and use Gaussian distribution to represent the distribution of continuous in Q. We add the loss of Q function to the Generative nets, aim to emphasize the distribution of discrete and continuous codes in the generating which usually be neglected in the traditional GANs(Chen, X., et al. 2016).

LOSS FUNCTION

The responsibility of the Discriminator is to distinguish between true or fake. Therefore, the loss function set of the Discriminator consists of the Loss_real function and the Loss_fake function. The specific functions of these two loss functions are as below:

Loss_real function: The discriminator identifies the probability that the real data matches the real labels.

Loss_fake function: The discriminator identifies the probability that the fake data matches the fake labels.

On account of the mutual information between the **G (z, c)** and latent codes **c**. The loss functions of the Generator are more than the original function set. There is not only gen_loss function but also categorical codes and continuous codes corresponding to dis _loss function and con_loss function. The specific functions of these two loss functions are shown below:

Gen_real function: The discriminator identifies the probability that the fake data matches the real labels.

Dis_real function: BEHIND THE ORIGINAL NOISE PLUS whole discrete code which is present in **ONE-HOT CODE**.

Con_real function: Combine the Crossentropy Loss and BCE Loss.

SHAPE-LIMIT LOSS

Below shows a formula for the Shape-limit loss function(y_i: i-th target distribution,x_i: i-th feature distribution). The purpose of the loss function is to control the discrete points in the synthetic fake 3D models generated in a similar shape to real data. We begin with the distribution of fake models and real data, adjusting the similarity of the distribution. And we add this function to the whole Generator Loss function set by reducing the total value of the function set to achieve the goal we want.

$$SHAPE - LIMIT\ LOSS = \lambda_1 * (y_i * log^{y_i - x_i}) + \lambda_2 * (y_i - x_i)^2.$$

Figure 5 shows the diagram of the formula's left part, when the difference of target and feature from 0 to 1, the result is a negative value, and the closer the difference is to 0, the closer the result is nearly negative infinity. Even if a positive function is added to this part, such as the right part of the formula, it will not affect the interpolation between the target and feature (between 0 and 1). However, to prevent the gradient explosion caused by the total value of the loss function falling too fast, we set two hyper-parameters $\lambda_1$and $\lambda_2$to adjust the balance of generation.

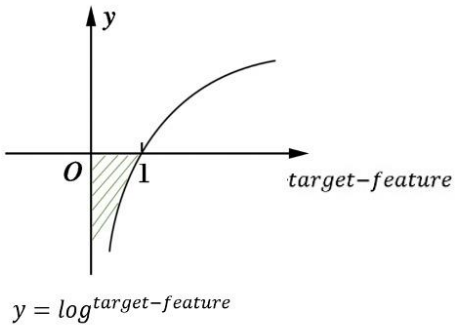

FIGURE 5. The Diagram of Formula's Left Part

Figure 6 shows the downtrend of Shape-limit loss, which means the loss assists the decrease of the Generator's loss function during the generation process, and at the same time according to the results of synthetic models with Shape-limit loss or without Shape-limit loss (shown in chapter 4), provided our Shape-limit loss worked well.

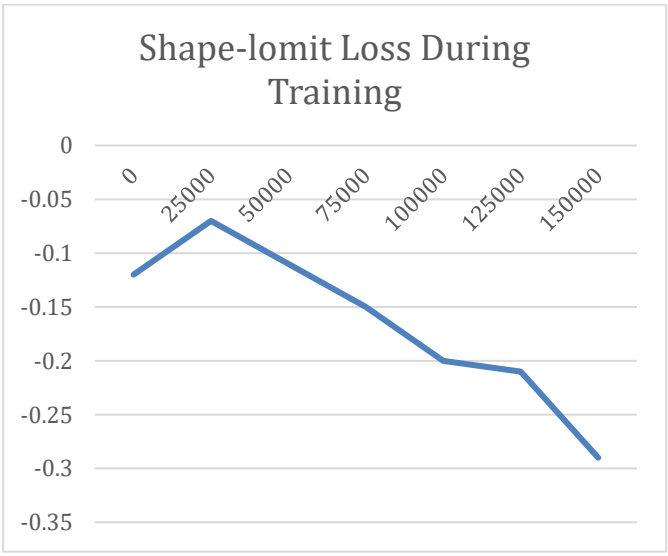

FIGURE 6. Shape-lomit Loss

SOFT LABEL

Labels are very commonly used in machine learning. They divided into two types: a soft label and a hard label. In this subsection, we will introduce the identity of the label. Also, we will explain the difference between the hard label and soft label.

For example (Zhou, Z. H. 2016), if you receive a melon, how to judge that it is a good melon? For humans, based on previous experience, we will first extract some useful information from the specific thing of melon, such as the colour of a melon, the shape of the melon, the sound of percussion, and so on. Then use certain rules from this information to judge. We think melons with a greenish-green colour, curled-up roots, and turbid sounds are good melons under normal circumstances.

In the above example, "good melon" and "bad melon", these two judgments are labels (Zhou, Z. H. 2016).

Hard label: The value of the hard label always show 0 or 1.

Soft label: Obviously, the value of soft coding is more flexible, such as 0.4, 0.4, 1.2, and so forth.

For this model, Generator and discriminator can advance in a smooth process.

EXPERIMENT AND EVALUATION

SOFT LABEL

We manually set the real_label and fake_label in the model of Discriminator and the setting of real_label is (0.7, 1.2), fake_lable is (0, 0.3). Figure 7 shows the loss expression in the case of no soft label, but with the Shape-limit loss function, there are very large ups and downs in the early stage during the training, which means that the training process is not smooth greatly reduces the efficiency of training. Figure 8 shows the loss expression in the case of both Shape-limit loss and soft label. Under the effect of soft labels, their training proceeded smoothly without major fluctuations.

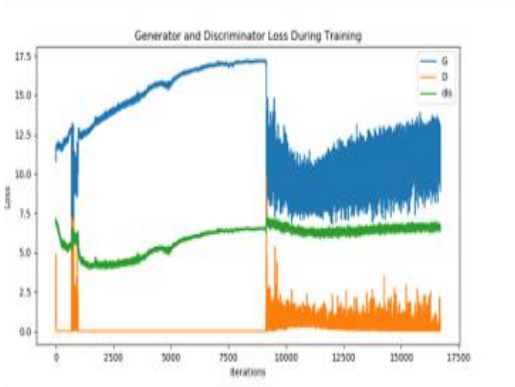

FIGURE 7. No Soft Label

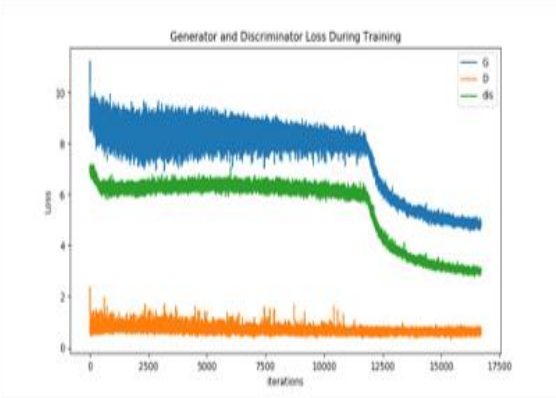

FIGURE 8. With Soft Label

SHAPE-LIMIT LOSS

We compared different results of three cases, with soft label but no Shape-limit loss, with both Shape-limit loss and soft label, no soft label but with Shape-limit loss. The results of Figure 11 are contrasted with the results of Figure 9 and Figure 10, which reflect the success of Shape-limit loss. Without Shape-limit loss, even if soft labels are added and the training is stable, the 3D synthetic model cannot be obtained

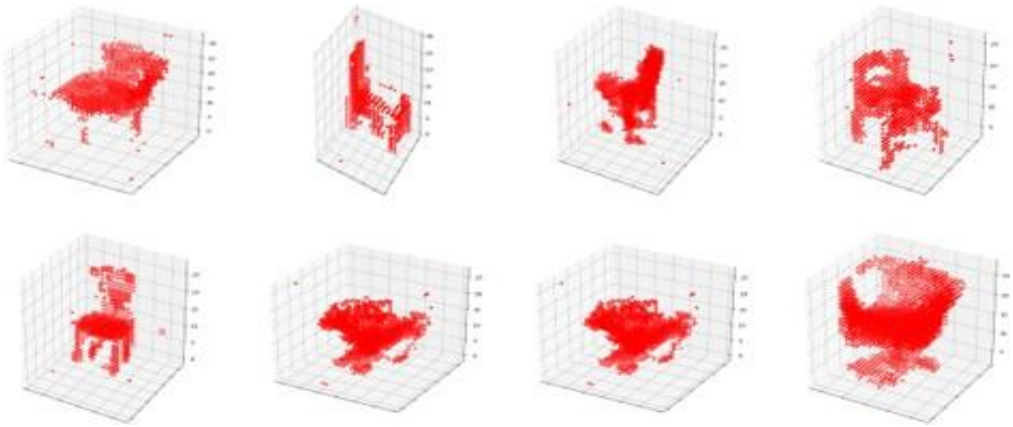

FIGURE 9. No Soft Label But With Shape-limit Loss

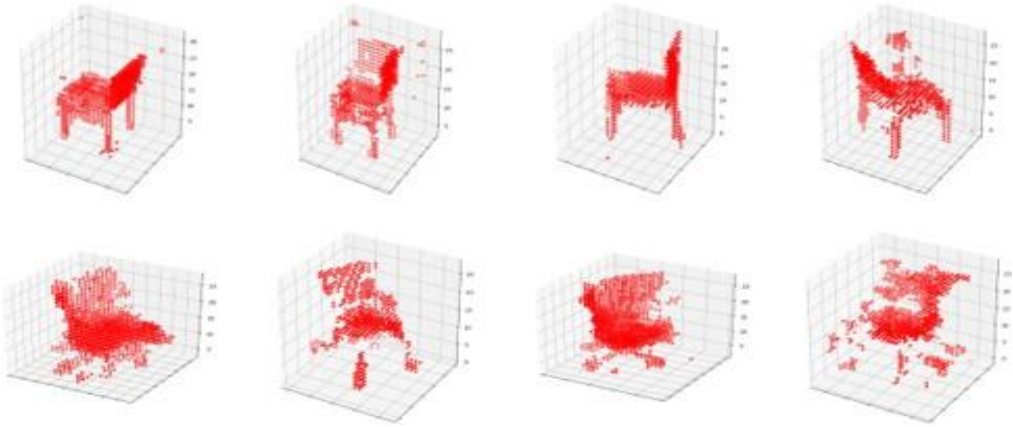

FIGURE 10. With Soft Label And Shape-limit Loss

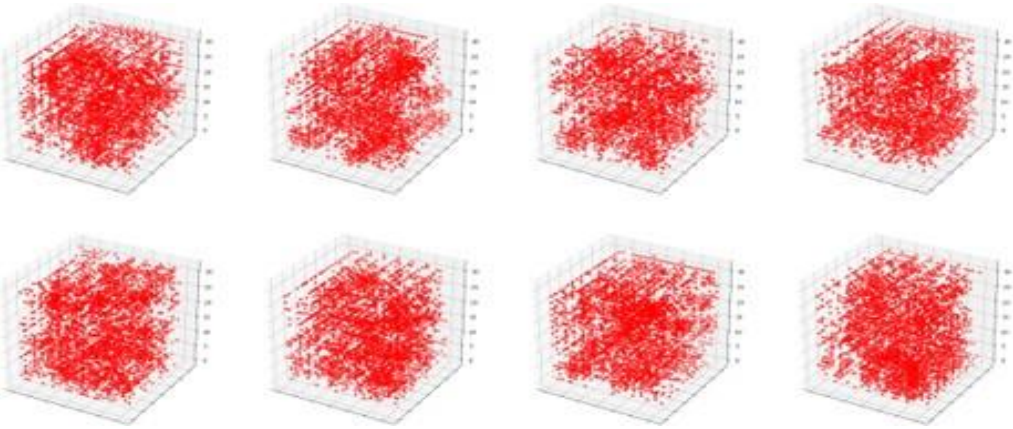

FIGURE 11. With Soft Label But Without Shape-limit Loss

EVALUATION

In our evaluation, we evaluate not only the model visually but also quantitatively. Most methods for evaluating the shape of the reconstructed model based on measuring the distance between the target point and the generated point. In this experiment, we used two of these methods the average distance between generated voxels and target voxels,

and the MSE (mean squared error) methods to assess the reconstruction errors of our deep model. To have the value of reference and comparison, we compared the values of CNN-GANs and IM-GANs with the same methods, which shown in Table 1. We also utilized the AP (average precision) method to evaluate the voxel prediction in our model and compared it with 3D-GAN, which shown in Table 2. All of them are model reconstruction experiments based on highly comparable voxel units.

TABLE 1. 3D Reconstruction Errors

	A1
3DInFoGANs-average	34.3
CNN64-average	**7.34**
IM64-GANs-average	8.96
3DInFoGANs-MSE	29.9
CNN64-MSE	**7.76**
IM64-GANs-MSE	11.43

TABLE 2. Average Precision For Voxel Prediction

	A1
3DInFoGANs-AP	20.1
3DGAN-AP	**47.2**

The models shown in the above tables all use 64 sampling resolutions, the obtained MSE and average results are multiplied by 1000, and the better performance results have marked in bold. It is clear from Table 1 and Table 2 that our reconstruction error and voxel prediction inaccuracy are high. It can be known from experiments and tests that in addition to the model construction still needs improvement, the convergence of the discrete latent codes still needs to be further increased. Although we have completed the purpose of this research, through evaluation, it still needs to be deepened.

In addition to evaluating reconstruction errors, we also evaluate the accuracy of model generation. It is obvious from the visual angle that the accuracy of our generated model is not as good as CNN-GAN and IM-GAN. The result of voxel prediction proves from the data that the accuracy of our generated model is very low. Therefore, the next goal of this research is to reduce reconstruction errors and improve the accuracy of model generation.

EXPERIMENT AND TEST FOR LATENT CODES

The purpose of this experiment is to obtain a 3D synthetic model successfully. This model is expected to achieve our experimental goal, apply deep learning to reconstruct the 3-dimensional models from the probability latent space, and discover the latent codes in the 3-dimensional space corresponding controllable features.

MODEL OPTIMIZATION

The hidden layer in deep learning is equivalent to the linear combination of input features, and the weight between the hidden and input layers is equivalent to the weight of input features in the linear combination. Moreover, the learning ability of the deep model will increase exponentially with the increase of depth.

We experimented with model optimization without the influence of the latent codes and set initial input parameters. There are five layers in the original InFoGANs discriminator net because of the unique architecture of InFoGANs' Discriminator. The features extracted from the five layers need to share in the D_function and Q_function. The deeper the model, the finer the extracted features. Due to it took too much time to run, we still tried to reduce one layer to run the different layers experiment and compare the run time results.

After the experiments, we get a similar result of 3D synthetic models in the case of 5 share layers and 4 share layers. Both can successfully generate 3D objects in chair shapes. However, in the case of 4 share layers, running 100 epochs took 1 minute at most. The opposite of, in the case of 5 share layers, running 100 epochs took 1 minute at least because of the time cost compared to when both cases can generate 3D synthetic models. We naturally choose the four share layers.

PARAMETERS

In this experiment, we want to find which parameters influence the generating of 3D synthetic models. From the results with the initial parameters, we have known it can successfully synthetic 3D models without fixed latent codes. We just design three style experiments for continuous codes and discrete codes.

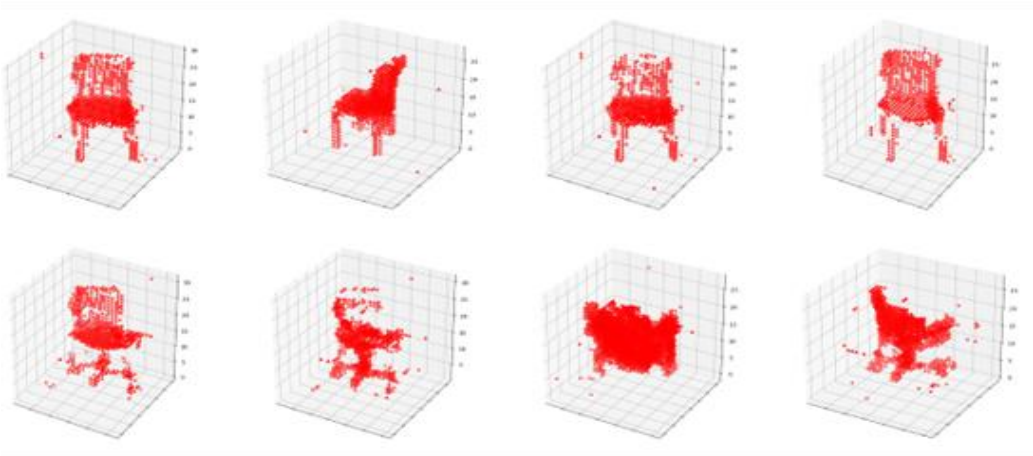

FIGURE 12. Only With Continuous Codes

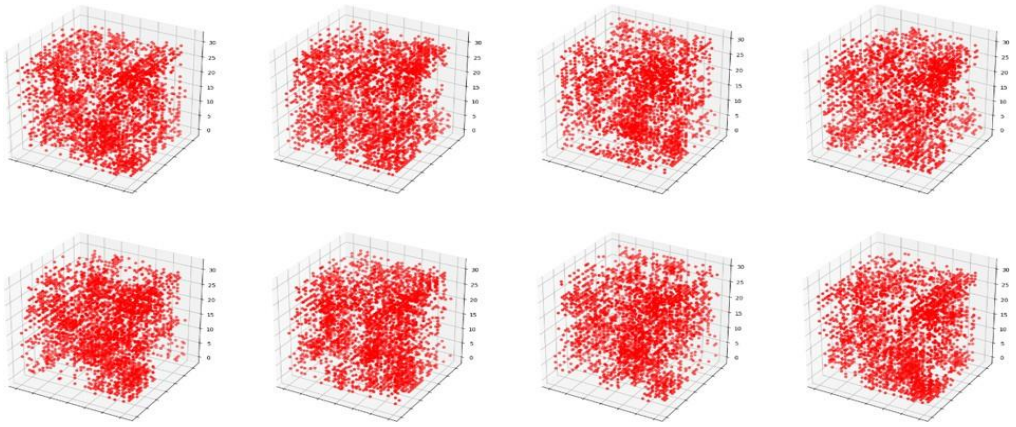

FIGURE 13. Only With Discrete Codes

From Figure 11 and Figure 12, we can find there is no way to converge the discrete points in the generation of the 3d composite model, resulting in discrete points scattered outside the shape. Of course, although the results of the discrete points are very different from the original results, this does not mean that the discrete codes have a great influence on the generation of the 3D synthetic model, which still requires us to conduct a final test.

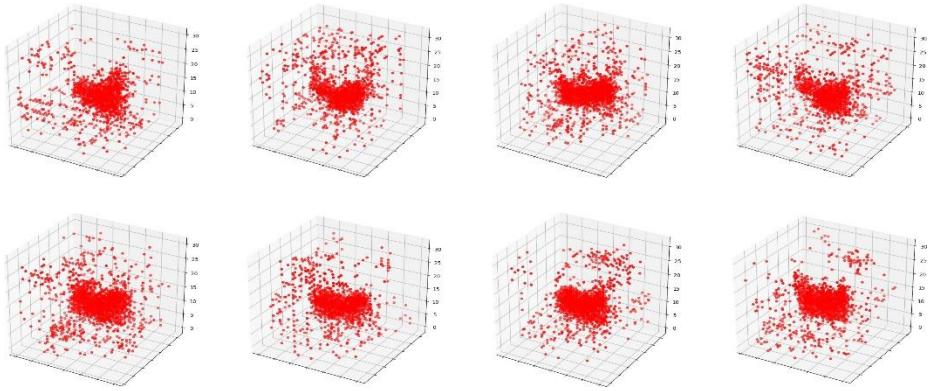

FIGURE 13. With 10 Discrete Codes and 2 Continuous Codes.

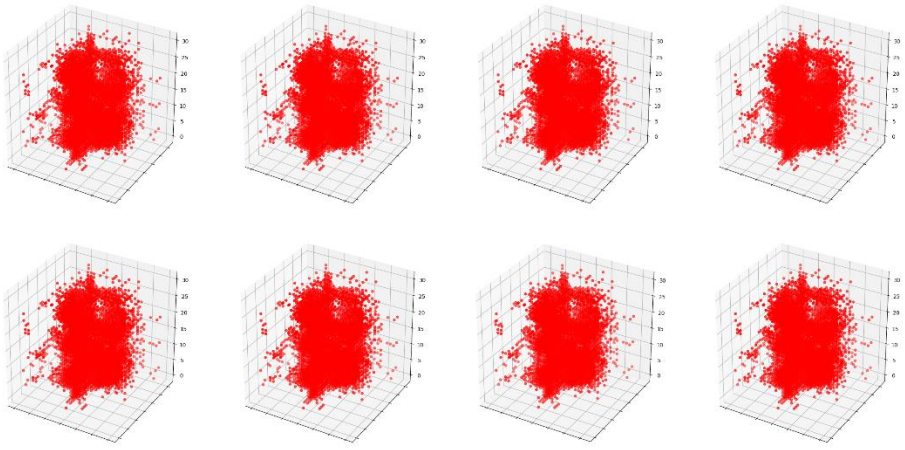

FIGURE 14. With 30 Discrete Codes and 2 Continuous Codes.

Only from the experimental results of fixing discrete codes and continuous codes individually, when only the continuous codes are fixed, the expected results can be obtained. On the contrary, when only the discrete codes are fixed, the current model

cannot converge to the discrete points very well, and the shape of the 3D synthetic model cannot be clearly seen. From the experimental results of fixing the discrete and continuous codes at the same time(Figure 13 and Figure 14), although discrete points still cannot converge well, and the shape of the 3D synthetic model cannot be clearly seen. With the increase of discrete codes, the result is as bad as the experimental result of fixed discrete codes alone. However, from the experimental results, we can roughly judge, the continuous codes correspond to the shape semantic features of the 3D synthetic model, and the discrete codes correspond to the detailed semantic features of the 3D synthetic model. Specific judgments still need to be tested before we can get them.

TEST

The purpose of our research is not only to reconstruct the 3D model but also to find the corresponding semantic features of latent codes in the latent space. To achieve this goal, we selected a certain type of 3D synthetic model with clear multi-tenant results to find the corresponding semantic features. Below (Figure 15, Figure 16 and Figure 17)we select the test of a set of results to illustrate:

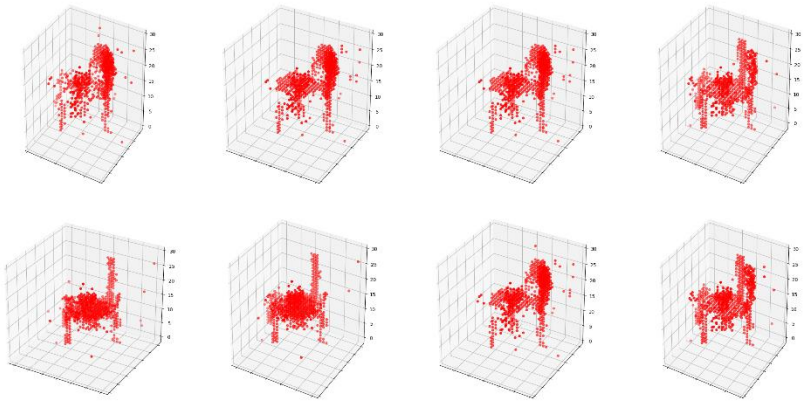

FIGURE 15. Original

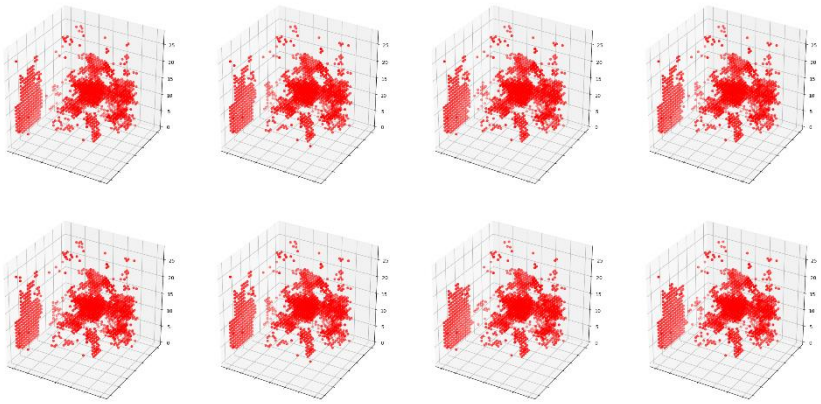

FIGURE 16. Change The Continuous Codes

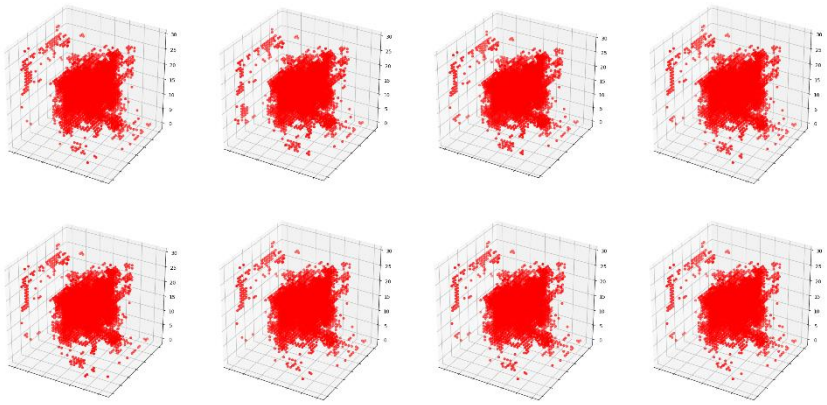

FIGURE 17. Change The Discrete Codes

Figure 15 is the original results set. Figure 17 shows the results set by changing the discrete codes. Figure 16 shows the results set by changing the continuous codes. From the test, we found the change of the continuous codes has an affect on the shape generating process. Through the change of the discrete codes, we found that, as discussed before, the discrete codes have a certain influence on the details of the 3D synthesized model. However, due to the limitations of the current model, we can only roughly test the impact of latent codes, and not find out the corresponding semantic features in 3-dimensional space.

CONCLUSION

In this article, we proposed a 3D model reconstruction method based on the deep model and converted the original 2-dimensional neural network to a 3-dimensional neural network for generating 3D objects. we added a shape-restricted loss function to the basic generation network and achieved the effect of converging discrete points by controlling the distribution of the discrete points of the synthesized 3D object to be close to the distribution of the target actual object. Based on this method by modifying the value of the fixed latent code in the latent space to obtain the change of the generated 3D model, which proves that the latent codes influence the generation process. This research laid the foundation for the next step of the research on the inverse mapping in the 3-dimensional space of the GAN model. We hope that further research on GAN inverse mapping can improve not only the generation efficiency of the 3D model but also the controllability of the generation.

REFERENCES

Aoki, H., Oman, C. M., Buckland, D. A., & Natapoff, A. (2008). Desktop-VR system for preflight 3D navigation training. *Acta astronautica*, *63*(7-10), 841-847.

Aditya, T. S. (2010, December). Survey on passive methods of image tampering detection. In *2010 International Conference on Communication and Computational Intelligence (INCOCCI)* (pp. 431-436). IEEE.

Alahari, K., Seguin, G., Sivic, J., & Laptev, I. 2013. Pose estimation and segmentation of people in 3D movies. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 2112-2119).

Brock, A., Donahue, J., & Simonyan, K. (2018). Large scale GAN training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., & Abbeel, P. (2016). Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *arXiv preprint arXiv:1606.03657*.

Chen, Z., & Zhang, H. (2019). Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 5939-5948).

Girdhar, R., Fouhey, D. F., Rodriguez, M., & Gupta, A. (2016, October). Learning a predictable and generative vector representation for objects. In *European Conference on Computer Vision* (pp. 484-499). Springer, Cham.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Machine learning basics. *Deep learning*, *1*, 98-164.

Gadelha, M., Wang, R., & Maji, S. (2018). Multiresolution tree networks for 3d point cloud processing. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 103-118)

Gao, L., Yang, J., Wu, T., Yuan, Y. J., Fu, H., Lai, Y. K., & Zhang, H. (2019). SDM-NET: Deep generative network for structured deformable mesh. *ACM Transactions on Graphics (TOG)*, *38*(6), 1-15.

Henderson, P., Tsiminaki, V., & Lampert, C. H. (2020). Leveraging 2d data to learn textured 3d mesh generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 7498-7507).

Li, R., Zou, K., Xu, X., Li, Y., & Li, Z. 2011, October. Research of interactive 3D virtual fitting room on web environment. In 2011 Fourth International Symposium on Computational Intelligence and Design (Vol. 1, pp. 32-35). IEEE.

Liao, Y., Donne, S., & Geiger, A. (2018). Deep marching cubes: Learning explicit surface representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2916-2925).

Malik, A., Lhachemi, H., & Shorten, R. (2020). I-nteract 2.0: A Cyber-Physical System to Design 3D Models using Mixed Reality Technologies and Deep Learning for Additive Manufacturing. *arXiv preprint arXiv:2010.11025*.

Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

Sinha, A., Bai, J., & Ramani, K. (2016, October). Deep learning 3D shape surfaces using geometry images. In *European conference on computer vision* (pp. 223-240). Springer, Cham.

Sra, M., Garrido-Jurado, S., Schmandt, C., & Maes, P. (2016, November). Procedurally generated virtual reality from 3D reconstructed physical space. In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology* (pp. 191-200).

Tatarchenko, M., Dosovitskiy, A., & Brox, T. (2017). Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 2088-2096).

Wu, J., Zhang, C., Xue, T., Freeman, W. T., & Tenenbaum, J. B. (2016). Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. *arXiv preprint arXiv:1610.07584*.

Wang, L., Hao, Y., Li, X., & Fang, Y. (2020). 3D Meta-Registration: Learning to Learn Registration of 3D Point Clouds. *arXiv preprint arXiv:2010.11504*.

Zhou, Z. H. (2016). Learnware: on the future of machine learning. *Frontiers Comput. Sci.*, *10*(4), 589-590.

Du Chunqi
Shinobu Hasegawa
Research Center for Entertainment Science,
Japan Advanced Institute of Science and Technology, Japan
s1810126@gmail.com, hasegawa@jaist.ac.jp