Jurnal Teknologi Informasi dan Pendidikan
Volume 15, No. 2, September 2022
https://doi.org/10.24036/tip.v15i2

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

3D MODEL RECONSTRUCTION USING GAN AND 2.5D SKETCHES FROM 2D IMAGE

Quach Thi Bich Nhuong[1], Pham Dinh Sac [1], Nguyen Minh Nhut[1], Hien-Thanh Le[1]*✉
[1]Department of Technology, Dong Nai Technology University, Bien Hoa 810000, Viet Nam
**Corresponding Author: lethanhhien2012@gmail.com*

Article Information

Article history:

No. 613
Rec. July 31, 2022
Rev. august 31, 2022
Acc. September 25, 2022
Pub. September 29, 2022
Page. 1 – 10

Keywords:

- *Reconstruction*
- *Deep learning*
- *Convolutional neural network*
- *2.5D sketch*
- *3D shape*

ABSTRACT

In the current 4.0 era, many fields such as medicine, cinema, architecture, etc. often use 3D models to visualize objects. However, there is not always enough information or equipment to build a 3D model. Another approach is to take multiple 2D images and convert to 3D shapes. This method requires information on images taken of objects at different angles. To get around this, we use a 2.5D sketch as an intermediary when going from 2D to 3D. A 2D photo is easier to create a 2.5D sketch than to convert directly to a 3D shape. In this paper, we propose a model consisting of three modules: The first is converting from 2D image to 2.5D sketch. The second is to go from a 2.5D sketch to a 3D shape. At last, refine the recently made 3D shape. Investigates the ShapeNet Core55 dataset show that our model gives improved results than customary models

.

1. INTRODUCTION

Currently, in order to simulate the image of the object in a visual and vivid way, helping the observer to have a more complete detailed view of the object and to interact with the object to achieve high results, 3D modeling It is applied in many fields such as: medical imaging, creating scenes and building characters in cinema, designing in architecture, 3D printing, ... [1] [2]. In some fields, 3D shape reconstruction of objects has been successfully performed using specialized equipment that captures images of objects from different angles for 3D shape reconstruction. [3] [4] [5]. To reconstruct 3D shapes directly from a single image requires full knowledge of the specific 3D geometry of the object. This poses a challenge in the method-based approach because 3D object information is very diverse in real images, previous research directions have only focused entirely on synthetic data [5] [6].][7], so it is often affected by the problem of data domain adaptation by imperfect finishing due to direct conversion from 2D to 3D, the reconstructed image has not reached the best efficiency. The 3D shape reconstruction methods have shown that if

reconstructing the 3D shape of the object through the 2.5D outline of the image will solve problems such as: contrasted and the total 3D shape, the portrayals of 2.5D drafts reestablished from a 2D picture are a lot simpler; 2.5D sketch recuperation models are likewise bound to go from engineered to genuine information. Additionally, to recreate 3D shapes from 2.5D representations, frameworks can advance totally from the composite information, thus can easily generate realistic 2.5D sketches without needing to simulate the changes of objects in real images such as light, texture... this partially solves the problem of domain adaptation [6] [8] [9]. From the above considerations, we propose a method of 3D shape reconstruction through 2.5D images that can solve the limitations posed in recreating 3D shapes of objects directly from single RGB images in order to achieve best regenerative performance. Due to the complexity of overlapping multiple objects in an image, we also limit our study to 2D RGB images consisting of only one object.

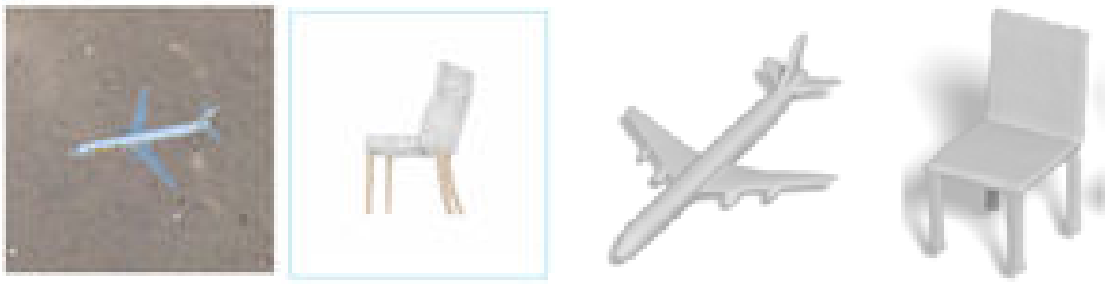

Figure 1. RGB and 3D images of the object

2. RELATED WORK

Right now, to tackle the issue of 3D shape remaking from a solitary RGB picture of an item, it very well may be partitioned into three methodologies: Method using embedded network TL (TL-embedding network), method method using Generative Adversarial Network (GAN) and method using Recurrent Reconstruction Neural Network.

2.1. TL-embedding network method

Derived from the idea of creating a vector representation of an object that meets two criteria [7]: One is that the representation must be generalizable in 3D, that is, it is possible to reconstruct the 3D shape of the objects. Second, it must be predictable from the 2D image, meaning that this 3D shape representation can be easily inferred from the 2D image. The TL inserted network model has two primary parts: a Training part (T-organization) and a Testing part (L-organization). Network L eliminates the encoding part of the autoencoder network (T-organization) and interfaces the result of the picture installing organization to a decoder to get voxel yield, which shows that the TL-implanted organization can be utilized to foresee a 3D voxel map for a given 2D picture.

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

2.2. Method of using adversarial network to generate 3D auto-encoder patterns

Based on the design of the example bad guy network [10][11], it comprises two primary parts, the example generator, and the discriminator. In the 3D-GAN [8] network, the model generator G maps a 200-layered secret vector z, erratically analyzed from a probability hidden away space, to 64 × 64 × 64 blocks, tending to a thing G(z) in 3D voxel space. The D discriminator yields a sureness regarding D(x) whether the commitment of the 3D thing is certifiable or made x. This technique makes 3D articles by examining the secret vector z and planning it into the item space. In the not very much organized network model for test age with 3D Variational Autoencoder Generative Adversarial Network - 3D-VEA-GAN [12], the creators broadened the 3D-GAN model by adding an extra picture encoder E, which accepts the 2D picture x as information and results in the secret portrayal vector z. This is exhibited in the investigation of VAE-GAN, which consolidates VAE and GAN by imparting the VAE decoder to GAN's example generator.

2.3. The method of using neural networks with regression reconstruction

This technique comes from the Long Short-Term Memory Network (LSTM) and the Convolutional Neural Network (CNN), the evaluation has improved and proposed another arrangement. 3D Recurrent Reconstruction Neural Network (3D-R2N2) [6]. The 3D-R2N2 cross-section takes at least one picture of an item according to alternate points of view and gives a recreation of the item as a lattice. The greatest benefit of the R2N2 network in both preparation and testing is that it requires no component class marks or picture explanations (for example no sections, central issues, or class marks). To tackle the hardships and difficulties of 3D shape remaking from a solitary RGB picture, we propose an answer for the issue by incorporating profound learning models with the learned shapes. preceding. This is a start-to-finish teachable model that makes three strides: gauge 2.5D representations, gauge 3D shapes, and refine 3D shapes

3. PROPOSED METHOD

As mentioned in the previous section, the end-to-end trainable proposed model consists of three steps: estimating 2.5D sketches, estimating 3D shapes, and refining 3D shapes.

Jurnal Teknologi Informasi dan Pendidikan
Volume 15, No. 2, September 2022
https://doi.org/10.24036/tip.v15i2

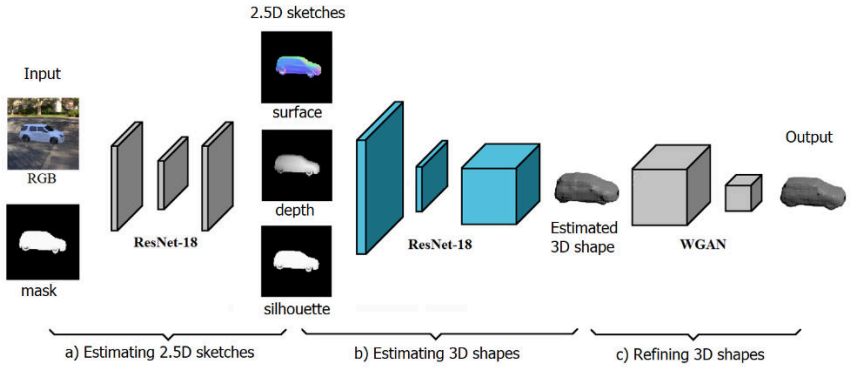

Figure 2. 3D shape reconstruction model through 2.5D sketch

3.1. Estimate 2.5D sketch from 2D image

This is the first component of the model (Figure 2a), which estimates the 2.5D outline of the object from a 2D RGB image. Inspired by the approach of MarrNet [13], this model is based on the architecture of ResNet-18 residual network [14]. Encoder using Resnet-18 with Conv1 layer modified kernel size from 7×7 to 3×3, stride is 2 and padding is 1 for the purpose of noise reduction and image smoothing when performing the integration. Convolution to encode from a 256×256 RGB 2D picture into 512 part guides of size 8×8. The decoder involves four deconvolution layers with a part of size 5×5, bounces of 2, and padding of 2. The result is a 2.5D sketch with surface, profundity, and veil data (Fig. 2b) and has a similar goal of 256×256.

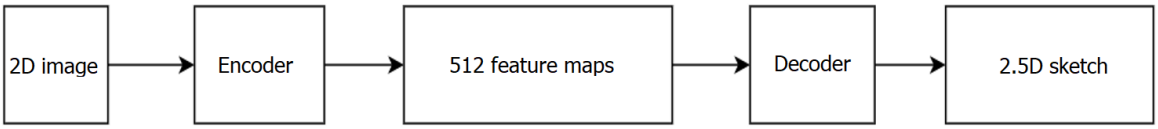

Figure 3. Flowchart of 2.5D sketch estimation process

3.2. Estimating 3D shapes from 2.5D. sketches

The second part of the model (Figure 2b) is an estimate of the 3D object shape from the 2.5D sketch estimated in the previous step (Figure 2a). Since it takes only 3 information of surface, depth, and shadow as input, the model can be trained from synthetic datasets without the problem of non-adaptation because it is easy to render 2.5 sketches D rather than 2D realistic images. Inspired by the embedded TL network and the 3D-VEA-GAN network presented in Section 2, the 3D shape estimation model (Figure 4) is an encoder-decoder network used to predict 3D shapes from 2.5D sketch. The encoder is also adapted from ResNet-18, performing convolution with a 3×3 kernel, stride of 2, padding of 1 to encode the 2.5D sketch into a 200-dimensional hidden vector. This vector then goes through a decoder consisting of 5 layers of 3D deconvolution with step and padding changed across

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

the layers, combining batch-norm and a ReLU activation function to generate 128×128×128 three-dimensional voxel shape.

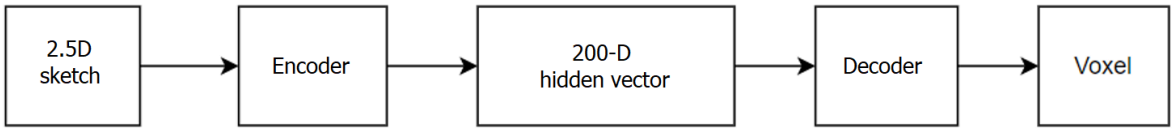

Figure 4. 3Dshape estimation process flow chart

3.3. Fine-tune the accuracy of predicted 3D shapes

Due to the 3D shape obtained from the 2.5D sketch following the above step, the results will not be high. Therefore, we refine this shape using the 3D-GAN model [11]. In this way, the model will increase the accuracy of the final 3D shape. The idea is to build a discriminator that will check the 3D shape created from step 2.

The contrast between the proposed technique and the strategy for MarrNet: Both of the above approaches are inspired by the TL embedded network and the 3D-VEA-GAN network to rely on the 2.5D sketch as an intermediary. However, MarrNet only uses the neural network and the loss functions corresponding to the 3 information of the 2.5D sketch to edit the estimated 3D shape. Meanwhile, we use the GAN model to enhance the quality of the 3D shape. Compared with the conventional neural network, the GAN model will be better at tuning tasks. Therefore, by creating a discriminator in the GAN model, the proposed model can go from the 2D RGB image to the refined 3D shape estimated from this image.

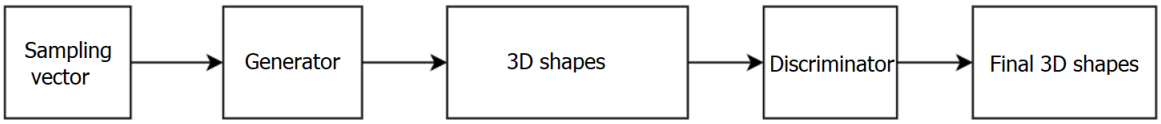

Figure 5. Flowchart for refining the accuracy of the predicted 3D shape

First, we utilize a pre-prepared 3D-GAN [11] organization to decide whether the 3D shape created in sync 2 of the model is sensible. Its example generator combines a 3D shape from a haphazardly drawn vector, and its discriminator separates created shapes from genuine shapes schematically in Figure 5. Subsequently, the discriminator. is equipped for displaying genuine shape dispersions and can be utilized as the misfortune capability of the model. The model generator does not participate in the final training of the model's shape. The pattern generator generates a 3D shape with input as a random vector going through 5 layers of 3D convolution with stride and variable padding across the layers, batch-norm matching. and ReLU along with the final sigmoid layer to create a 128×128×128 voxel shape. For the discriminator, use 5 layers of 3D convolution and leaky ReLU to distinguish the 3D shape generated from the model generator and the actual shape

3.4. Training

The 2.5D sketch estimation network was trained with the loss function $L_{2.5D}$ which is the sum of the $L2$ errors of the 3 surface, depth and mask information, utilizing the Stochastic Gradient Descent - SGD calculation with a learning rate is 1e-3, number of epoch iterations is 300, using optimization according to Adam's algorithm.

For the 3D shape estimation network, we utilize the cross-entropy loss function L_{3D} to train the network at this stage and still using the SGD algorithm [15], the learning rate is 1e-3 with momentum is 0.9, the number of epoch iterations is 80

Finally, the 3D-GAN model to refine the 3D shape, because of the multidimensionality of the 3D shape (128×128×128), the preparation of the GAN becomes unsteady. To take care of this issue, we utilize the blunder GAN Wasserstein [16][17].

$$L_{WGAN} = \underset{\tilde{x}\sim P_g}{\mathrm{E}}[D(\tilde{x})] - \underset{x\sim P_r}{E}[D(x)] + \lambda \underset{\hat{x}\sim P_{\hat{x}}}{E}[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2] \quad (1)$$

where D is the discriminator, P_g and P_r are the estimated 3D shape and the actual 3D shape, respectively.

Algorithm 1. Complete 3D shape network training

```
Input: epoch
Output: 3D shape complete network model
Begin
        netG ← GanGenerator()          # Initialization
        netD ← GanDiscriminator()      # Initialization
        optimizerG ← torch.Adam(netG.parameters(), learningrate)
        optimizerD ← torch.Adam(netG.parameters(), learningrate)
        # Train D
        # Real
        errorDReal ← netD(real).mean()
        errorDReal.backward()          # pytorch library
        # Fake
        errorDFake ← netD(fake).mean()
        errorDFake.backward()
        # gradient penalty
        grad ← torch.autograd.grad()
        gradientPenalty ← (((grads + 1e-16).norm(2, dim←1) - wnorm) **
2).mean()*wlambda
        gradientPenalty.backward()
        optimizerD.step()
        #Train G
        gen ← netG()
```


P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

errorG ← netD(gen).mean()
errorG.backward()
optimizerG.step()

End

According to the author Gulrajan, the second part of formula (1) is the gradient penalty, we choose $\lambda = 10$as suggested by the author. During training, follow the formula $\mathop{\mathbb{E}}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})] - \mathop{\mathbb{E}}_{x\sim\mathbb{P}_r}[D(x)]$, the discriminator tries to minimize the loss function L_{WGAN} during the pattern generator try to maximize the loss function. From formula (1), we further define the precision error as là $L_{exacting} = -\mathop{\mathbb{E}}_{\tilde{x}\sim P_c}[D(\tilde{x})]$ enhances the performance of the model, where P_c is the complete reconstruction from this network. The 3D shape exactness tweaking network is prepared on the example generator G and the discriminator D, which utilizes Adam [18] enhancement with a learning pace of 1e-4 and a clump size of 4 for 80 ages. Network D is prepared on genuine example, faker example, and slope punishment as equation (1)

The complete model is trained against the 3D shape estimation network and the D discriminator of the 3D shape refinement network. This model uses a loss function that is the sum of the loss function error of the 3D shape estimation model and the accuracy error as mentioned in the previous section: $L = L_{3D} + \alpha L_{exacting}$. In the experiment, we choose $\alpha = 10^{-11}$ for the best results, this model also uses the SGD algorithm and runs in 80 epochs. The subtleties of the preparation are displayed in Algorithm 1.

4. EXPERIMENTS

The framework is conveyed on a PC with CPU processor setup: Intel® Core™ i7-7700HQ 2.8GHz, super lift up to 3.8Ghz, 6M Cache, RAM 16GB DDR4 transport 2400Mhz, 256GB SSD hard drive, outfitted with discrete designs card NVIDIA GeForce GTX 1050i 4GB DDR5 128 bit. PC with Ubuntu 16.04 LTS working framework introduced with Python 3.6, CUDA 9.0, Blender 2.76 programming introduced, running on Anaconda climate with help bundles: pytorch=0.4.1, numpy=1.15.4, tensorflow=1.5.1, torchvision= 0.2.1,.

4.1. Evaluation Metric

$$IoU = \frac{|A \cap B|}{|A \cup B|} \quad (3)$$

where A, B is the resulting 3D shape of the model and the actual shape of the object

4.2. ShapeNet Core55 dataset

ShapeNet Core55 dataset [19] provided by MIT institute, the total number of subjects in both training and testing dataset is 5,652 objects where seats are 1,816, car is 1,906,

airplane is 1,930 with each object consists of an RGB image in 20 random, unrestricted views, and with a description of the object's general shape across the views. The dataset is 152.7GB in size. The data ratio for training and testing is 70/30 or 80/20, In this paper, we choose the ratio 80/20 to increase the training data. Due to imperfect test conditions, we test 3 times and average results as Table 1.

When compared with other methods, our method for IoU is quite good for rectangular objects like chairs, cars. However, the method gives bad results for the aircraft object.

Table 1 Average accuracy of 3 trials

Experiments	IoU			
	Chair	**Car**	**Planes**	**Average**
1st	0.501	0.786	0.564	0.617
2nd	0.485	0.735	0.512	0.577
3rd	0.483	0.767	0.547	0.599
Average	0.490	0.763	0.541	0.598

Table 2 Compare the proposed method with other methods

Experiments	IoU			
	Car	**Chair**	**Planes**	**Average**
3D-EPN [20]	0.274	0.147	0.155	0.181
Proposed method	**0.763**	**0.490**	0.541	0.598
ShapeHD [21]	0.698	0.488	0.452	0.529
DRC 3D [9]	0.760	0.470	**0.570**	**0.600**

From Table 2 we can see that the model gives good results with rectangular shapes such as chairs and cars. However, for non-rectangular shapes, the model still gives bad results. Other methods train on synthetic data and focus on isolated objects. When training data is synthetic and test is performed on real data, there is also a significant discrepancy of

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

test performance. On the other hand, these methods only possible with shallow architectures.

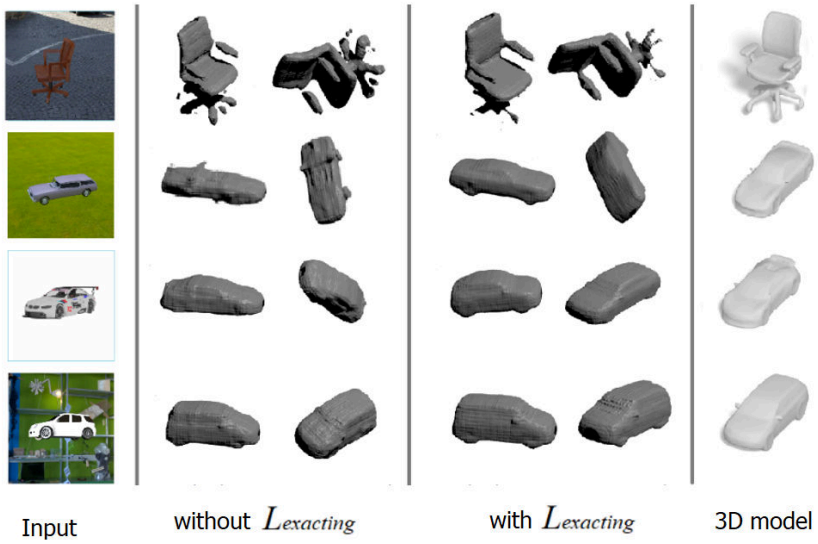

Figure 6 Results of the complete reconstruction of the 3D shape of the object on some models of airplanes, cars and seats

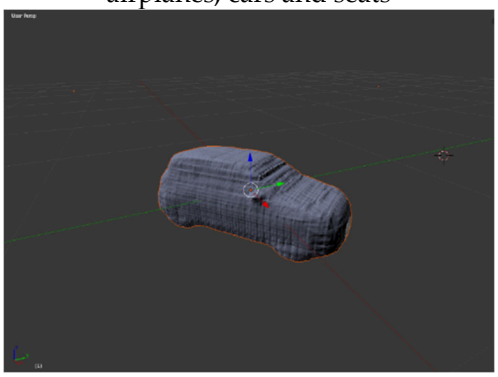

Figure 7 Rendering a 3D image of an object using Blender

Figure 6 shows that when $L_{exacting}$ is not used, the resulting 3D model is not as smooth as when $L_{exacting}$ is used. To visualize the created 3D model, we use Blender software to describe the generated 3D image as shown in Figure

5. CONCLUSION AND DEVELOPMENT ORIENTATION

In this paper, we built a model to reproduce the 3D shape of the object through 2.5D sketch. With the addition of a transition from a 2D still image to a 2.5D sketch, the model is divided into several stages with a specific purpose. On account of that, the 3D state of the article is implicit in the most ideal way. Tests show that our technique works on the nature of the produced 3D shapes when contrasted and past strategies.

In the future, we will continue to research and develop the model to increase performance and expand the study on images of other objects as well as many overlapping objects in the same image. In addition, based on the resulting 3D shape of the object, we build algorithms to calculate the volume, or mass of the object (such as a person) from a photograph or image recorded directly. continue through camera.

REFERENCES

[1] D. Freitag. "The Role of 3D Displays in Medical Imaging Applications." Internet: https://www.meddeviceonline.com, May. 18, 2015.
[2] L. Landini et al. "3D Medical Image Processing," in Image Processing in Radiology. Berlin: Heidelberg, 2008, pp. 67-85.
[3] A. Patel and K. Mehta. "3D Modeling and Rendering of 2D Medical Image," in International Conference on Communication Systems and Network Technologies, pp. 149-152, 2012
[4] B. Landoni. "3D Scanning with Microsoft Kinect." Internet: https://www.open-electronics.org, May. 6, 2015
[5] T. Shubham et al. "Multi-view Supervision for Single-view," in arXiv:1704.06254, pp. 1-9, 2017
[6] C. B. Choy et al. "3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction," in arXiv:1604.00449, pp. 1-17, 2016
[7] R. Girdhar et al. "Learning a Predictable and Generative Vector Representation for Objects," in arXiv:1603.08637v2, pp. 1-16, 2016
[8] J. Wu et al. "Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling," in NIPS'16, pp. 82–90, 2016
[9] S. Tulsiani et al. "Multi-view Supervision for Single-view Reconstruction via Differentiable Ray Consistency," in arXiv: 1704.06254, pp. 1-9, 2017
[10] A. Krizhevsky et al. "Imagenet Classification with Deep Convolutional Neural Networks," in NIPS, pp. 1097-1105, 2012
[11] L. Do. "Generative Adversarial Networks (GANs)." Internet: https://ai.hblab.vn, Sep. 2017
[12] A. Larsen et al. "Autoencoding Beyond Pixels using a Learned Similarity Metric," in arXiv:1512.09300v2, pp. 2-4, 2016
[13] Wu, Jiajun, et al. "Marrnet: 3d shape reconstruction via 2.5 d sketches." arXiv preprint arXiv:1711.03129 (2017)
[14] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on Computer vision and pattern recognition. 2016
[15] Ketkar, Nikhil. "Stochastic gradient descent." Deep learning with Python. Apress, Berkeley, CA, 2017. 113-132.
[16] L. Weng. "From GAN to WGAN?." Internet: https://lilianweng.github.io, Aug. 2017
[17] I. Gulrajan et al. "Improved Training of Wasserstein GANs," in arXiv:1704.00028v3, pp. 1-5, 2017
[18] D. P. Kingma and J. L. Ba. "Adam: A Method for Stochastic Optimization," in arXiv:1412.6980, pp. 1-9, 2015
[19] Di. Zhou et al. "IoU Loss for 2D/3D Object Detection," in arXiv: 1908.03851, pp. 3-4, 2019
[20] A. Chang et al. "Shapenet: An Information-rich 3D Model Repository," in arXiv:1512.03012, pp. 2-6, 2015
[21] T. Shubham et al. "Multi-view Supervision for Single-view," in arXiv:1704.06254, pp. 1-9, 2017
J. Wu et al. "Learning Shape Priors for Single-View 3D Completion and Reconstruction," in arXiv:1809.05068, pp. 1-14, 2018

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id