A 3D Flower Modeling Method Based on a Single Image

Ju Ming, Zhu Siyuan, Wang Meili, Lin Jiaxian

A 3D Flower Modeling Method Based on a Single Image

Jiaxian Lin[1,a] and Ju Ming[2,a] and Siyuan Zhu[3,a] and Meili Wang[*,a]

[a] College of Information Engineering, Northwest A & F University, Yangling 712100, China
[1]nwsuaf_ljx@qq.com [2]jm55480@nwsuaf.edu.cn
[3]2017050988@nwsuaf.edu.cn [*]wml@nwsuaf.edu.cn

Abstract. Since the structure of the flower is too complex, the modeling of the flower faces huge challenges. This paper collects 3D scenes containing flower models, uses 3dsMax to extract flower models, and constructs flower dataset. This paper proposes an encoder-decoder network structure called MVF3D and adopted the trained MVF3D network to predict the missing perspective information, use a single RGB image to generate a depth map of flowers from different perspectives, and finally use the depth maps to reconstruct the flower models. To evaluate the performance of our proposed method, for simple flowers, the average chamfer distance between the reconstructed 3D model and the real model is 0.27, The experimental results have shown that our proposed method can preserve the true structure of the flower.

Keywords: Flower Models, 3D Reconstruction, Deep Learning.

1 Introduction

With the research and development of visualization technology, virtual reality technology and computer vision technology, the application of 3D modeling technology in virtual reality games is more and more widely used. The flower model increases the diversity of game scene model reconstruction. Despite the modeling tools such as Maya, 3dsMax, and AutoCAD have rapidly development in recent years, using these modeling tools still requires a certain learning cost, and researchers cannot use these modeling tools efficiently to complete the 3D reconstruction of flowers. Therefore, how to generate exquisite flower models under simple input data has become a meaningful work. Traditional flower modeling methods simulate plant growth through mathematical modeling but ignore the individual structure of flowers [1]. The use of interactive hand-painting [2,3] not only requires complicated interactive operations, but also relies too much on the user's subjective understanding of the flower structure, and the reconstructed flowers modeled lack realism.

In daily life, users can easily obtain RGB images of flowers through mobile phones, digital cameras, and other devices. More and more researchers are studying plant modeling methods based on single or multiple RGB images. Using multiple RGB images and simple interactive operations can generate realistic flower models [4]. Similarly, a real natural vegetation scene can be constructed by using a single

RGB image of the tree and the contour of the tree crown [5]. With the development of deep learning, it is no longer difficult to infer a 3D structure from a 2D image [6,7,8,9]. However, there are few studies on flower 3D modeling combined with deep learning, and there is a lack of flower dataset that can be used for deep learning training. In view of the above problems, this paper aims to improve modeling speed of 3D flower model, and presents an algorithm for generating RGBD images of different viewing angles from a single flower image and a method of reconstructing the flower grid model based on RGBD images from multiple different viewing angles. This method avoids manual interaction, improves model reuse, and has certain application value for the production of 3D game scenes.

The structure of this article is as follows. Sect. 2 introduces the composition of the flower dataset. Sect. 3 introduces a deep learning network model MVF3D that uses a single flower image to generate multi-view RGBD images. Sect. 4 introduces how to use RGBD images from different perspectives to perform 3D reconstruction of the flower model. Sect. 5 presents conclusions and future works.

2 Flower Dataset

Deep learning needs to summarize the discovery rules from the dataset, and how to construct the flower data that meets the input of the deep neural network is the primary goal of this paper. The 520 flower models used in this paper come from websites which provide 3D scene models. By manually separating them, 4106 individual flower models are obtained, which are then rendered using Panda3D to obtain a dataset suitable for deep neural networks.

2.1 Flower model preprocessing

This paper collected 520 3D scene models containing flowers from the Internet. An example is shown in Figure 1. In order to facilitate the 3D reconstruction of the flower model, it is necessary to segment the flower clusters in the scene model to ensure that the segmented model contains only one flower. At present, the semantic automatic segmentation algorithm based on the skeleton or feature points of the 3D model has the limitation that with the model complexity increases, the segmentation effect becomes worse. So, this paper used a 3D modeling tool 3dsMax to process the scene model containing flowers.

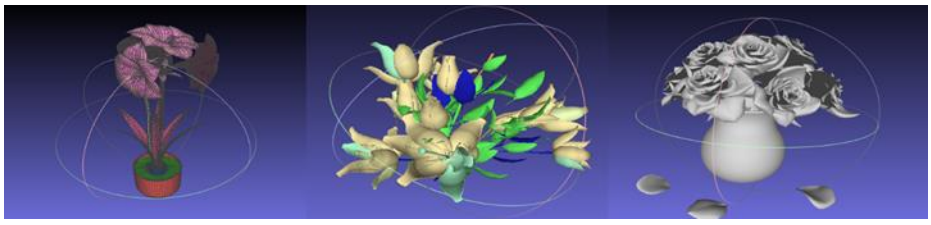

Fig. 1. Scene models containing flowers collected from the Internet

Preprocessing each flower model. The preprocessing operation ensures that the size of the extracted flower model is roughly the same, and the model center is located at the origin of the world coordinates. The flower model after preprocessing is shown in Figure 2.

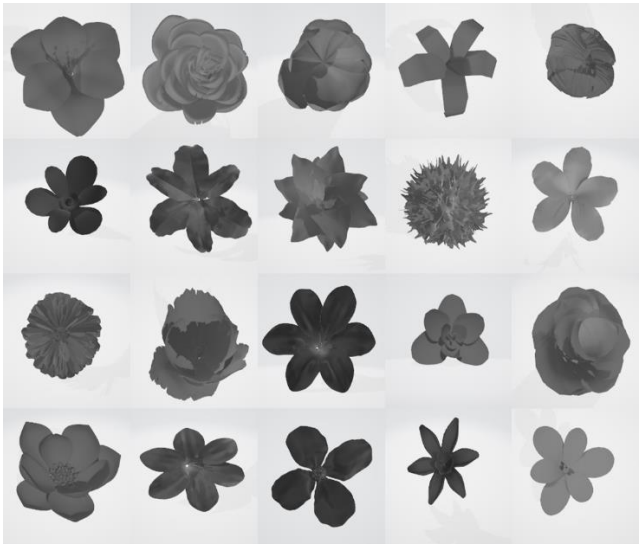

Fig.2. Single flower models segmented by 3dsMax

2.2 Panda3D rendering flower model

When constructing training samples using panda3d, in order to increase the robustness of the network, the given camera elevation angle is a random angle between -10 degrees and 30 degrees. By adding 15 degrees at a time, each model can generate 24 pictures. Regarding the generation of depth maps, Panda3D uses UVN cameras to render objects, and the depth value is the distance from the camera position in UVN coordinates. As shown in Figure 3, the left figure shows the depth map rendered by Panda3D, and the right figure shows the RGB image rendered by Panda3D. It can be seen that the generated RGBD image reflects the shape of the flower under different perspectives.

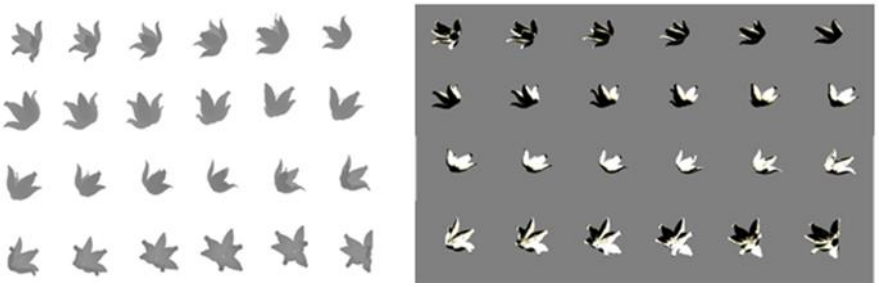

Fig.3. Panda3D rendering effect of flower model

2.3 Distinction of flower dataset

Considering that the quality of RGB images and depth maps generated by deep learning depends on the accuracy of coding for the representation of 3D information, in order to reduce the training difficulty of deep neural networks, this study divides the flower dataset into simple patterns and complex patterns according to their morphology. The pattern is divided into different dataset, and the two flower pattern datasets are trained separately. The simple flower pattern is defined as having a clear petal structure. The human eye can distinguish the number of petals. The complex flower pattern cannot be judged by the human eye on the number and approximate structure of petals. The self-occlusion between the petals is serious, and there is no clear distribution rule. Figure 4 shows examples of simple patterns and complex patterns, with simple patterns on the left and complex patterns on the right. Of the 4106 individual flower models, 1310 are classified as complex models and 2796 are classified as simple models.

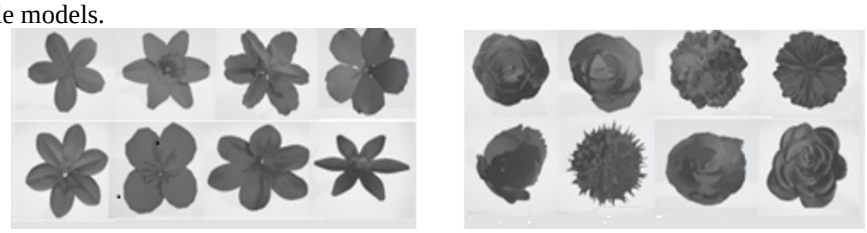

Fig.4. Schematic diagram of pattern differentiation

After rendering each flower model by Panda3D, we can get 24 RGBD images at different viewing angles. Randomly extract two different RGBD images in 24 images, construct 276 sample pairs, select one sample RGB images as input y_i in each sample pair, and another sample RGB image $\widehat{y_i}$ and depth map $\widehat{d_i}$ as deep learning Real label, such a model can build 552 samples. Leave-one-out method is used to divide the training set and the test set. The division of the flower training set is shown in Table 1. So far, we have constructed a 3D flower dataset that can be used for deep neural network training.

Table 1. Preprocessed Flower Datasets.

Types of dataset	Number of training samples	Number of test samples
Simple flower dataset	1 389 052	154340
Complex flower dataset	650 808	72 312

3 Generate Flower RGBD Images Based on MVF3D

3.1 MVF3D Architecture

MV3D (Multi-view 3D Models from Single Images) is a simple and elegant encoder-decoder structure network [7]. Given any view vector θ, the MV3D network can gen-

erate RGB images and depth maps of objects. In order to adapt the MV3D network to the flower dataset constructed in this paper, the network structure of MV3D is modified in this paper. The modified network structure is shown in Figure 5.

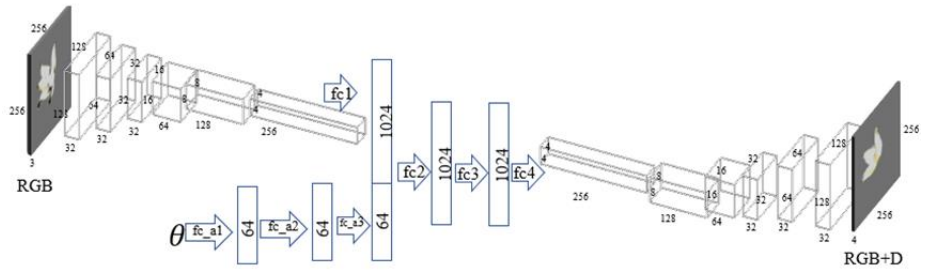

Fig.5. MVF3D algorithm Architecture

This paper named the modified network structure MVF3D (Multi-view 3D Flower Models from Single Images). In order to get a more detailed flower model, the MVF3D network fixes the input image resolution to 256×256, and adds a convolution operation that does not change the size of the feature map after each layer of convolution operations, which increases the depth of the network, making MVF3D can extract more abstract information from the RGB image of flowers [10]. After six convolution operations, a 4×4×256 feature map is obtained. The obtained 4096-dimensional vector is used to extract the 1024-dimensional flower model feature representation through the fully connected layer fc1 (4096×1024). In the encoding process, the target angle of view θ is also required. The target angle of view θ is composed of 5 variables, which are the distance *rad* of the UVN camera and the object, The sine value $\sin el$ and cosine value $\cos az$ of the camera elevation angle *el*, the sine value $\sin az$ and cosine value $\cos az$ of the camera rotation angle *az*. The target view vector $\theta(rad, \sin el, \cos el, \sin az, \cos az)$ is processed by 3 fully connected layers, and the obtained 64-dimensional vector is combined with the vector obtained from the flower RGB image. At this point, the task of the encoder is completed.

After the 1084-dimensional encoding vector is obtained, the decoder performs 3 fully connected layer operations on it, deconvolutes the vector according to the structure of the encoder, and finally obtains the predicted RGB images and depth maps of the target angle of view. MVF3D uses the same loss function (1) as MV3D.

$$L = \sum_i \|y_i - \hat{y}_i\|_2^2 + \lambda \left\| d_i - \hat{d}_i \right\|_1 \tag{1}$$

Where y_i and d_i are the RGB image and depth map output by the MVF3D network, respectively $\hat{y}_i$ and $\hat{d}_i$ are their corresponding labels, and λ is the hyperparameter. Considering the training cost, this study limits the super parameter λ range from 0.1 to 1.0 and increased by 0.1 each time. Experiments show that when λ = 0.8, good results can be achieved. Interested Researchers can spend more time looking for a better λ value.

3.2 Experimental results

In the experiment, the RGB images with different views obtained by training the simple flower model and the complex flower model are shown in Figure 6 and Figure 7, respectively.

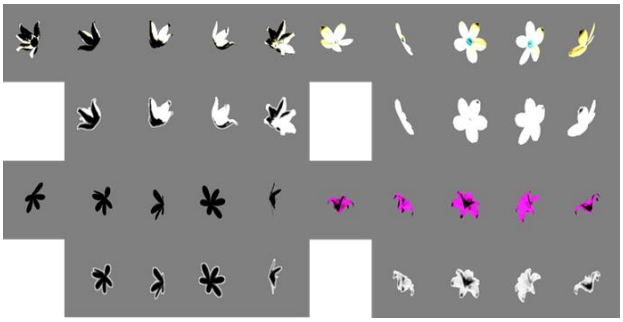

Fig.6. MVF3D generates multi-view results of simple flower model (The upper left is the input RGB image, the first line is the is the real RGB image, and the second line is the result generated by MVF3D)

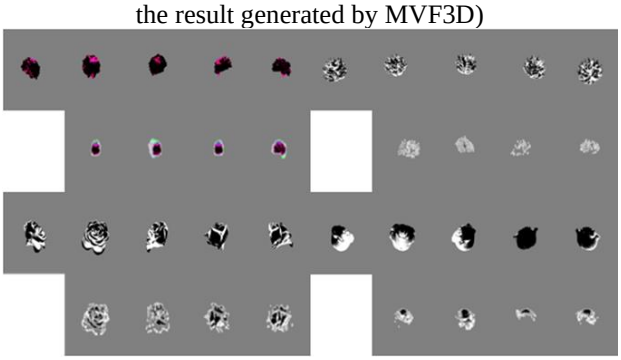

Fig.7. MVF3D generates multi-view results of complex flower model (The upper left is the input RGB image, the first line is the is the real RGB image, and the second line is the result generated by MVF3D)

It can be found that for the simple flower model, the predicted RGB images is blurry compared to the real RGB images, and a part of the detailed information is lost. There is a deviation in the color of the generated RGB images, but the original structure of the flower is roughly restored. The blurring of the generated picture is the result of the Euclidean loss function. The Euclidean loss function guides the network to average the latent images, which leads to the generation of blurred images. The difference in color is because compared with the difference in color, the set loss function (1) punishes the shape of the object with Euclidean loss and L1 loss, and the punishment for color difference is only Euclidean loss. Therefore, MVF3D pays more attention to the geometric structure of flowers than colors. For 3D reconstruction, determining the geometric structure of objects is more important than color. In this paper, the mutual information is used to measure the loss of the image and the real image. Mutual information is also called normalized mutual information. It is an expression to measure the similarity of two images. The larger the value, the more simi-

lar the two pictures are. Usually used as a criterion or objective function in image registration. It has a good effect for the case where the difference of the gray level of the image is not large.

As shown in Figure 8, for flowers with a simple structure, the mutual information of the target depth map and the predicted depth map is 0.6743 on average, which can prove that MVF3D has learned the abstract expression of the flower model and restored most of the depth information of the simple flower model.

Fig.8. MVF3D generates multi-view depth maps of a simple flower model

Compared with the simple flower model, MVF3D performs poorly in the prediction of complex flower data. The generated RGB images are weird. In addition to the more complex pattern and fewer training samples, that also because of the large number of complex flower petals and the complicated structure. As a result, MVF3D has no way to accurately find the feature space corresponding to complex flowers, so MVF3D does not perform as well as simple flowers in complex flowers. As shown in Figure 9, it can be seen that for flowers with complex shapes, MVF3D does not predict the corresponding 3D structure of flowers well. Through calculation, the average mutual information between the target depth map and the predicted depth map of the complex flower model is 0.3396 also confirmed this.

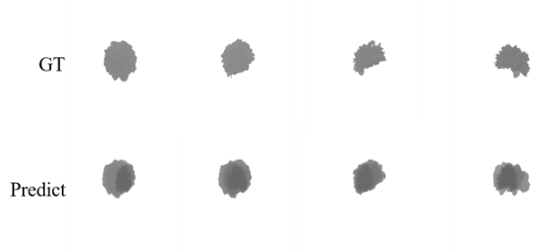

Fig.9. MVF3D generates multi-view depth maps of a complex flower model

4 3D Reconstruction Based on RGBD Images of Flowers

In order to build a complete flower model, the algorithm in this chapter is mainly divided into four steps: In the first step, all matching feature point pair sets $M\{(m_1, m_1'), (m_2, m_2'), \cdots, (m_k, m_k')\}$ are extracted from the RGB images. In the second step, in the depth map D, a point cloud matching algorithm is performed on the set M according to the feature points to obtain the camera poses at different perspec-

tives, and the rotation matrix R and the translation matrix t are obtained. The third step is to combine the point cloud to obtain the complete point cloud P. The fourth step is to use P to generate grid data V. The algorithm framework is shown in Figure 10.

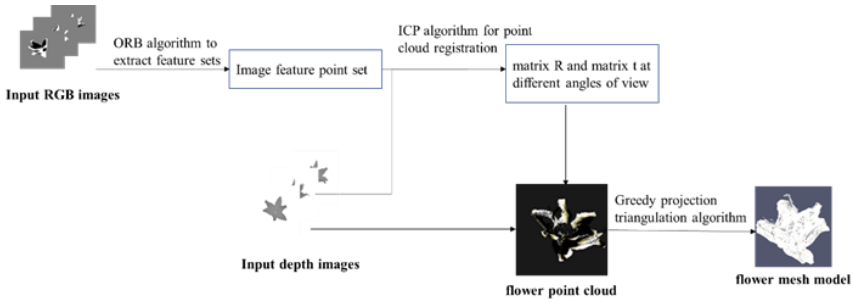

Fig.10. Algorithm framework for 3D reconstruction of flowers

4.1 Flower RGB image feature extraction and matching method

The ORB (Oriented FAST and Rotated BRIEF) algorithm is efficient on extracting and describing feature points [11]. The ORB is fast and has stable rotation invariance. At the same time, the size of the flowers in the RGB images generated by MVF3D is basically the same, and the angle of the adjacent RGB images is 15 degrees different. This paper uses the ORB algorithm to extract features from flower RGB images. As shown in Figure 11, the circle on the picture represents the feature point determined by the ORB algorithm.

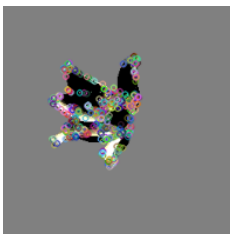

Fig.11. ORB extracted flower RGB image features

After obtaining the ORB features of each flower perspective, this paper uses Hamming distance as shown in equation (2) as the feature point similarity metric for flower images from different perspectives.

$$d(x,y) = \sum_{i=1}^{n} x[i] \oplus y[i] \tag{2}$$

x and y are the n-dimensional ORB feature vectors extracted from the two flower pictures.

Comprehensive efficiency and accuracy, this paper uses FLANN algorithm [12] to match the feature points and descriptors of the flower RGB images. The rotation angles of the two flower pictures to be matched differ by 15 degrees. Figure 12 shows the effect of using the FLANN algorithm to match it.

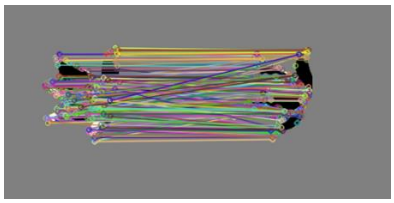

Fig.12. FLANN algorithm matching result

It can be seen that there has a large number of mismatch points. Based on experience, this paper screens the matching results. The screening criterion is to leave the matching points where the Hamming distance is less than twice the minimum distance, and the minimum distance should be greater than 30. After screening, the matching results of the FLANN algorithm obtained in this paper are shown in Figure 13. It can be seen that a large number of mismatched point pairs are filtered, leaving the correct matched point pairs, which also reduces the burden of subsequent camera pose estimation.

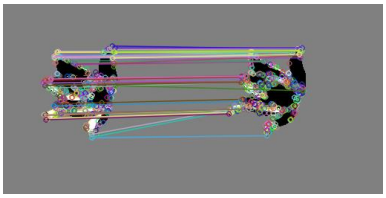

Fig.13. FLANN algorithm matching screening result

4.2 Flower point cloud registration method

In this paper, the ICP algorithm [13] is used to solve the rotation matrix R and translation matrix t between the flowers of different views generated by MVF3D. Since there are 24 flower images with different viewing angles, this paper specifies that the first image rendered by Panda3D is the standard coordinate system, and the remaining view images need to be transformed into the standard coordinate system through rotation and translation operations. In order to facilitate the measurement of the error with the real flower model, after the point cloud registration, the camera pose of the first picture needs to be used to convert the point cloud into the original world coordinate system and normalize it.

The ORB feature has a poor effect of extracting feature point pairs when the image rotation angle is too large. Therefore, when solving the rotation matrix R_{Si} and the translation matrix t_{Si}, we first convert the perspective image i coordinate to the perspective coordinate of the intermediate image i-1 (or i+1) closer to the standard coordinate, and find the rotation matrix $R_{(i-1)i}$ or $R_{(i+1)i}$ and translation matrix $t_{(i-1)i}$ or $t_{(i+1)i}$, and then converted into the coordinates of the target angle standard image S through the intermediate perspective. In this way, the error can be evenly distributed to each transformation.

Since the total number of point clouds obtained after registration is uncertain, this study uses the chamfer distance to measure the gap between the real flower point

cloud and the predicted flower point cloud. The definition of the chamfer distance is shown in equation (3).

$$CD(S_1, S_2) = \frac{1}{|S_1|}\sum_{x\epsilon S_1} \min_{y\epsilon S_2} \|x - y\|_2 + \frac{1}{|S_2|}\sum_{y\epsilon S_2} \min_{x\epsilon S_1} \|y - x\|_2 \quad (3)$$

S_1 is the target point cloud, and S_2 is the predicted point cloud. The chamfer distance calculates the average error between two point clouds. Figure 14 shows the flower point cloud model after point cloud registration using 6 depth maps with the angle of view difference from standard coordinates of 0, 60, 105, 165, -135, and -30. It can be seen that the point cloud is relatively evenly distributed. The average chamfer distance between the real point cloud and the predicted point cloud is 0.2703, which also confirms this view.

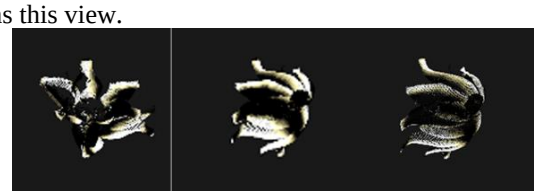

Fig.14. Flower point cloud after registration by ICP algorithm

4.3 Flower meshing method

For the flower model, the petal surface meets the smooth condition, the point cloud generated by multi-view flower image registration will not have holes. Therefore, this paper uses the greedy projection triangulation algorithm [14] to convert the simple flower point cloud into a mesh model, and obtain the final result of the simple flower 3D reconstruction, as shown in Figure 15.

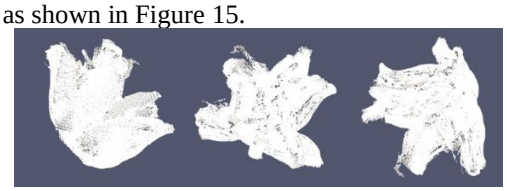

Fig.15. 3D mesh reconstruction effect of a simple flower

It can be seen that the generated mesh model restores the general shape of the flower model. However, due to the blurry RGBD images output by the MVF3D network and the angle of view estimation error, the created 3D model has some noise and lost some details. Compared with directly using a single real RGBD image to convert into a flower mesh model, as shown in Figure 16, the flower model generated in this paper reduces the probability of occurrence of holes and saves the subsequent steps of hole filling.

Fig.16. Simple flower mesh model reconstructed from a single RGBD image

5 Conclusions and Future Works

This paper improved the MV3D network structure, and constructed a flower model dataset for MVF3D network training. The structure can infer the RGB image and depth map of the simple flower model at different viewing angles from the input single flower image. The depth map was used to reconstruct the 3D mesh model of the flower. This work required less manual interaction and improved the modeling efficiency.

In the future work, due to the poor performance of MVF3D for complex flower models, it considers using image segmentation algorithm [15,16] to segment each part of the flower, and then modeling and combining each part to form a complete flower. At the same time, it considers collecting more flower samples to improve the performance of the network facing complex flower models.

For the details of reconstruction, it considers using an adversarial generation network [17] training samples to obtain clearer RGB images in our future work. In the processing of point cloud registration, it may be considered to introduce a deep neural network to improve the registration accuracy of point clouds from different perspectives [18,19].

References

1. Prusinkiewiczy, P.: A look at the visual modeling of plants using L-systems. In: Hofestadt, R., Lengauer, T., Loffler, M., Schomburg, D. (eds.) Bioinformatics. GCB 1996. LNCS, vol. 1278,pp. 11-29. Springer, Heidelberg(1997).
2. Steven, L., Adam, R., Frederic, B., Przemyslaw, P.: TreeSketch: interactive procedural modeling of trees on a tablet. In: Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling, pp. 107–120. Eurographics Association, Goslar (2012)
3. Lintermann, B., Deussen, O.: A Modelling Method and User Interface for Creating Plants. In: Computer Graphics Forum, pp. 73–82. Wiley-Blackwell, Malden (1998).
4. Quan, L., Tan, P., Zeng, G., Yuan, L. Wang, J. Kang, SB.: Image-based plant modeling. ACM Transactions on Graphics 25(3), 599-604(2006).
5. Argudo, O., Chica, A., Andujar, C.: Single-picture reconstruction and rendering of trees for plausible vegetation synthesis. Computers & Graphics 57, 55-67(2016).
6. Chao, W., Zhang, Y., Li, Z., Fu, Y.: Pixel2mesh++: multi-view 3d mesh generation via deformation.ICCV, 1042-1051(2019).

7. Tatarchenko, M., Dosovitskiy, A., Brox, T.: Multi-view 3D Models from Single Images with a Convolutional Network. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds) ECCV 2016. LNCS, vol. 9911, pp. 322–337.Springer, Cham(2016).
8. Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., Jiang, YG.: Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds) ECCV 2018. LNCS, vol. 11215, pp. 55–71. Springer, Cham(2018)
9. Zhou, T., Tulsiani, S., Sun, W., Malik, J., Efros, A.A.: View Synthesis by Appearance Flow. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds) ECCV 2016. LNCS, vol 9908, pp. 286–301. Springer, Cham(2016).
10. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. ICLR, 1-14(2015).
11. Ethan, R., Vincent, R., Kurt, K., Gary, B.: ORB: An efficient alternative to SIFT or SURF. ICCV, 2564–2571(2011).
12. Marius, M., David, G.: Scalable nearest neighbor algorithms for high dimensional data. IEEE transactions on pattern analysis and machine intelligence 36(11), 2227–2240(2014).
13. Besl, P., Mckay, N.: A Method for Registration of 3-D Shapes. The International Society for Optical Engineering 14(3),239-256(1992).
14. Liu, J., Bai, D., Chen, L.: 3-D Point Cloud Registration Algorithm Based on Greedy Projection Triangulation. Applied Sciences 8(10),1–10(2018).
15. Zhang, Z.,Duan, C., Lin, T., Zhou, S., Wang, Y., Gao, X.:GVFOM: a novel external force for active contour based image segmentation. Information Sciences 506,1-18(2020).
16. Wang, W., Wang, Y., Wu, Y., Lin T., Li, S., Chen, B.: Quantification of left ventricle via deep regression learning with contour guidance. IEEE Access 7, 47918-47928 (2019).
17. Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Aaron, C., Yoshua, B. :Generative adversarial networks. Advances in Neural Information Processing Systems 3, 2672-2680(2014).
18. Wang, Y., Xie, Z., Xu, K., Dou, Y., Lei, Y.: An efficient and effective convolutional auto-encoder extreme learning machine network for 3d feature learning. Neurocomputing 174(PB), 988-998(2015).
19. Dou, P., Kakadiaris, I. A.: Multi-View 3D Face Reconstruction with Deep Recurrent Neural Networks. Image and Vision Computing, 80-91(2018).