第 27 回日本バーチャルリアリティ学会大会論文集（2022 年 9 月）

3D Model Generation of a Shooting Scene from a Single Snapshot

スナップショット画像からの３次元撮影シーンモデルの生成方法

Luoxi Zhang[1)], Hidehiko Shishido[2)], Itaru Kitahara[2)]

1) Doctoral Program in Empowerment Informatics, University of Tsukuba
(〒305-8573 Tennoudai 1-1-1, Tsukuba, Ibaraki, zhang.luoxi@image.iit.tsukuba.ac.jp)
2) Center for Computational Sciences, University of Tsukuba
(〒305-8577 Tennoudai 1-1-1, Tsukuba, Ibaraki, {shishido | kitahara}@ccs.tsukuba.ac.jp)

Abstract: This paper proposes a method to acquire the 3D model of a scene from a single snap-shot image. We applied Mask R-CNN to divide the image into foreground and background for modeling. PifuHD is applied to process the human body (foreground) region for generating the 3D model. MiDas single-view depth estimation and open3D library are employed to obtain and process the point cloud of the background. Finally, both foreground and background 3D models are integrated by using the ICP algorithm. Our experiment shows that our proposed method successfully reconstructs a 3D model of the captured scene from a single shot.
Keywords: **3D Reconstruction, Monocular RGB-D Image, Point Cloud**

1. Introduction

Numbers of research focus on the task of 3D reconstruction and modeling from captured images. Usually, the 3D information often requires multiple 2D images from different perspectives as the input, which strongly restricts the shooting conditions. However, people usually capture images with one camera only once; thus a snapshot is the most common shooting method. In order to solve the mismatching between realistic input and desired output, we proposed a 3D modeling method that uses only one snapshot.

3D reconstruction of natural scenes based on monocular vision is a hybrid visual task. It can be combined with many current popular technologies, such as 3D printing technology and virtual reality visual games.

The research in this paper can be divided into three sub-tasks: “character modeling”, “background modeling”, and “position matching” between character model and background model. However, due to the single-purpose depth blur and the occlusion of the part of the figure in the image, the data set that can be adapted to our research is limited.

In contrast to the method of depth estimation based on the time series of multiple single-view images, our research has only one monocular image and no time series. Therefore, our depth acquisition needs to obtain the model according to neural network training. Thus, our depth estimation refers to MiDaS depth estimation and Refinenet network structure to obtain depth estimation through a pre-trained depth estimation model [2][11][12]. Eliminating the dependence on stereoscopic images or multiple images as input makes our method more widely applicable to all existing photographs.

2. Related works

2.1 Depth generation from single images

David et al. proposed the idea of recovering a depth map from a single image, which is one of the pioneering works using deep learning to do depth estimation [3]. The network is divided into global rough and local precise estimates, and a scale-invariant loss function is proposed to estimate the depth from rough to precise. The background reconstruction in our research also needs to use depth estimation. Our network structure about depth estimation is divided into two layers: MiDas and Refinenet [2][11][12].

2.2 3D reconstruction based on point cloud

Most extant works are using deep networks for 3D modeling with the 2D image datasets for the input data. However, this representation leads to a compromise between sampling resolution and net efficiency. Point clouds are a simple, unified structure that is easier to learn, and point clouds can be manipulated more easily during geometric transformations and deformations because connectivity does not need to be updated. Haoqiang Fan and others first proposed point cloud generation through single image 3D reconstruction [4]. A loss function Min-of-N loss (MoN) for point cloud network was constructed.

However, in Haoqiang Fan's work, the 3D reconstruction of the objects ignores their background. While, our final goal is to reconstruct complex background information.

2.3 Use the Triangular Mesh for a single view or multiple view 3D reconstruction

The triangular mesh method for 3D reconstruction does not need point cloud, depth, or other richer data, but directly obtains 3D mesh from a single color image. The usual framework is to initialize any input image into an ellipsoid shape. The network is divided into two parts. One uses low resolution to extract features to get a rough model, while the other is used to fill in details to enrich information. In the method proposed by Nanyang Wang et al. [5], they defined four kinds of loss to constrain the deformation of the grid. PIFU, which is the framework used for character reconstruction in PifuHD [1], also uses the network structure of end-to-end and coarse-to-fine. In addition, the PIFU framework also combines voxel information to judge whether each information point is inside the human body, so as to achieve the effect of predicting the back of the character.

3. Our Proposed Methods

Our research will solve the problem of the lack of background reconstruction in 3D reconstruction. We regard the person as the subject of the image, and the data used is the photo containing the person. In addition, we separated the background and the characters and carried out 3D reconstruction, respectively. For the character part, PifuHD [1], which has a great effect at present, is used to obtain a triangular mesh model. For the background part, point cloud generation is used for 3D reconstruction and the mesh information is obtained by surface reconstruction. And then combined with the human body. The whole process uses a single image as the only input. In the part of background reconstruction, it is necessary to solve the problems of sparse point detection and density enhancement, as well as the normal vectors required by surface reconstruction.

4. Image background segmentation

We input a single image O and then use Mask R-CNN [10] to separate the image's background from the character part of the main body and fill the separated character with white pixels to the size of the original image to obtain the image H. Then we used LaMa [6] network to repair the separated background image and obtained our fixed background image B.

$$L_{final} = \kappa L_{Adv} + \alpha L_{HRFPL} + \beta L_{DisPL} + \gamma R_1 \quad (1)$$

L_{Adv} and L_{DisPL} are responsible for the generation of naturally looking local details while L_{HRFPL} is responsible for the supervised signal and consistency of the global structure.

And to improve the image repair effect, we expanded the mask part of the figure outward to some pixels to expand our repair scope and prevent the edge information of the figure from affecting the image repair.

Next, we estimate the depth of the original image O and the inpainting background image B to obtain the depth map O_depth and B_depth by using MiDaS depth estimation method. But to obtain a richer depth map of hierarchical information, we adapt the Refinenet frame to run the MiDaS [2][11][12]. And then, we divide it into background and character for processing.

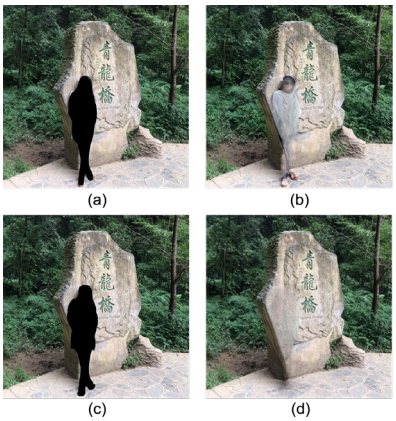

Figure 1: The images (a)(c) are the background after removing the characters, and the images (b)(d) are the background image B after inpainting the background with LaMa. Images (a)(b) are the character removal effect after Mask R-CNN is used. Images (c)(d) are the character removal effect after we expand the Mask edge.

5. 3D reconstruction

For the character (foreground) part, we used the PifuHD [1] model and use Blender to create our triangular mesh character model H_model. PifuHD is an end-to-end model that uses a two-layer network, one that uses low-resolution images to get a rough model of the person, and one that uses high-resolution images to get detailed information such as hair and clothing folds.

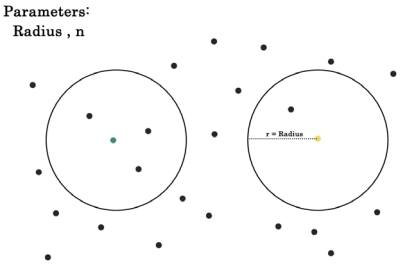

Figure 2: Sparse point detection method -- solitary point filtering. When there are less than n neighboring points in the circular domain of the specified radius RADIUS, the point is regarded as a sparse point. For example, when n=3, only the yellow point is removed.

As to the capturing environment, we combine the background image B with the depth image B_depth to generate a point

cloud B_pcl, and combine the image H with the depth image O_depth to generate a point cloud H_pcl. Then remove the white points in the space, leaving only the point cloud of the character part. After that, as shown in Figure 2, the sparse points are detected by solitary point filtering [7], and the density of the sparse points is enhanced. Because when a point cloud surface is reconstructed, the denser the points, the better.

However, this part does not pay special attention to the points. Therefore, considering the time consumption, we only need to randomly generate some points with the same color as the sparse points on the cross-section.

$$New_{Point} = Point + r \times \left(\frac{d}{|d|}\right) \quad (2)$$

$$d = \frac{a - a \cdot b \times b}{|b|^2} \quad (3)$$

Here is the limited random radius r of the generating point, in any direction on the tangent plane of the point, $\frac{d}{|d|}$ is a random 3D vector, r is the normal direction of the point. With this formula, multiple points can be randomly generated around sparse points, which is fast and easy to calculate. It is much faster than the upsampling method using the network.

Finally, the background is reconstructed by Poisson surface[8] to obtain the triangular mesh model of the background.

Poisson surface reconstruction is a very intuitive approach. Its core idea is that the point cloud represents the position of the object's surface, and its normal vector represents the direction of inside and outside. An estimate of a smooth object surface can be given by implicitly fitting an indicator function derived from the object.

Poisson's solution is L2 projection. They define the set of octree nodes as O. The vector space V can be approximated as:

$$\vec{V}(q) = \sum_{s \in S} \sum_{o \in Ngbr_D(s)} \alpha_{o,s} F_o(q) s\vec{N} \quad (4)$$

$$\tilde{\chi} = \sum_o x_o F_o \quad (5)$$

Here, $Ngbr_D(s)$ is the eight nearest leaf nodes of s, $\alpha_{o,s}$ is the weight of trilinear interpolation, and both V and χ can be represented in function space.

6. Point cloud registration

We scale the human model H_model and the human Point cloud $H_pcl$and use the ICP (Iterative Closest Point) method [9] to match their positions. Then we get the position of the separated character model in the reconstructed background model. Since the time consumption of the ICP algorithm is mainly spent on calculating the corresponding point set, the efficiency of the ICP algorithm can be improved if the time consumption of this step can be reduced. Therefore, before ICP operation, we manually moved the maximum XYZ axis coordinates of H_model and H_pcl to a similar position, and then used ICP for fine-tuning.

In the process of each iteration, the ICP algorithm logs each point of the data point cloud and finds the closest Euclidean point in the model point cloud as the corresponding point, and minimizes the objective function s^2 through this set of corresponding points:

$$s^2 = min \sum_{i=1}^{N} \|Q_i - (RP_i + t)\|^2 \quad (6)$$

According to the formula, we can obtain the optimal four-dimensional transformation matrix (including translation and rotation), apply the four-dimensional transformation matrix to the point cloud data, and obtain the new data point cloud and bring it into the next iteration process.

7. Evaluation

Because the same geometry may be represented by different point clouds to the same degree of approximation. Therefore, the Chamfer Distance (CD) evaluation method is mainly adopted in this paper to evaluate our final generated results. We first arranged the object scene from the virtual scene and obtained the groundtruth of its real point cloud information. Then, a single image was captured from the 2D perspective of the 3D scene to obtain the input data. Finally, the obtained point cloud results were compared with the groundtruth on CD.

Chamfer distance: We define the Chamfer distance between $S_1, S_2 \subseteq R^3$ as:

$$d_{CD}(S_1, S_2) = \frac{1}{S_1} \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2^2 + \frac{1}{S_2} \sum_{y \in S_2} \min_{x \in S_1} \|y - x\|_2^2 \quad (7)$$

In the strict sense, d_{CD} is not a distance function because triangle inequality does not hold. We nevertheless use the term "distance" to refer to any non-negative function defined on point set pairs. For each point, the algorithm of CD finds the nearest neighbor in the other set and sums the squared distances up. Viewed as a function of point locations in S_1 and S_2, CD is continuous and piecewise smooth. The range search for each point is independent, thus trivially parallelizable. Also, spatial data structures like kd-tree can be used to accelerate the nearest neighbor search. Though simple, CD produces reasonable high-quality results in practice.

Since different scenes may have different coordinate ranges of point clouds, we define that when d_{CD} is less than

$(density_{S_1} + density_{S_2})/2$, it is a qualified result; otherwise, it is regarded as a poor result. We want the point clouds to coincide as much as possible, so that the value of d_{CD} approaches zero. But we also have to think about how much error we can tolerate. We hope that the passing result is that at least half of the points are nearly identical, that is, the shortest distance is 0, and the shortest distance between the other half points and the target point will not exceed the average density of their point cloud, that is, even if the target point is added to the point cloud, the point cloud will not consider it as its "different point". However, the evaluation part has not been started yet because we lack the ground truth for comparison.

8. Experiments

8.1 Environment

Our experimental environment consisted of a Google Colab and a laptop with a 3070 GPU. The additional software needed was Blender software for coloring the character models. It is best to use snapshots with the full body of the character without obscurants as input data.

By using the default parameters of our system, we still could get pretty good results. But if camera parameters are available, we could obtain better results with more details.

In the table.1, *f* means focal length, and *c* means principle point (all units are in pixels). And we experiment with an image of 640 pixels in width to get our results.

Table 1. The camera parameters

parameters	width	height	fx	fy	cx	cy
default	640	480	600	320	550	550

8.2 Results

As shown in Figure 3, our experimental results prove the feasibility and robustness of our strategy, and we can achieve such a great effect with only a single snapshot.

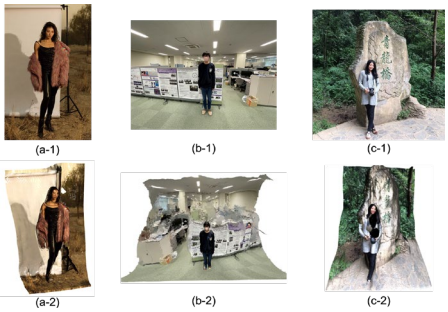

Figure 3: The experimental results show that our reconstructions apply to different image sizes and can be reconstructed not only for outdoor images but also for indoor images with good results.

9. Conclusion

Our research has realized the content of single snapshot 3D modeling in this paper and has had preliminary results. The whole project has begun to take shape. In the future, we'll improve the correct approach, for example images work best when people are standing up and using full-body shots. Other poses have poor modeling effects. Moreover the floor of the background is not modeled as flat as it should be. Occasionally, the character model still has some problems with the background model, and so on. These are the areas that we need to further strengthen. Enhancing the robustness of our project is our next goal.

References

[1] Saito, Shunsuke and Simon, Tomas and Saragih, Jason and Joo, Hanbyul. "PIFuHD: Multi-Level Pixel-Aligned Implicit Function for High-Resolution 3D Human Digitization." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (June 2020).

[2] Shih, Meng-Li and Su, Shih-Yang and Kopf, Johannes and Huang, Jia-Bin. "3D Photography using Context-aware Layered Depth Inpainting." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[3] Eigen D, Puhrsch C, Fergus R. "Depth Map Prediction from a Single Image using a Multi-Scale Deep Network[C]" *International Conference on Neural Information Processing Systems*. MIT Press, 2014.

[4] Fan H, Hao S, Guibas L. "A Point Set Generation Network for 3D Object Reconstruction from a Single Image[C]" *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE*, 2017.

[5] Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., Jiang, YG. "Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images." *ECCV*. 2018.

[6] Suvorov R, Logacheva E, Mashikhin A, et al. "Resolution-robust Large Mask Inpainting with Fourier Convolutions[J]". 2021.

[7] Open3D http://www.open3d.org/

[8] Kazhdan M. "Poisson surface reconstruction[C]" *Eurographics Symposium on Geometry Processing*. 2006.

[9] Arun K S."Least-Squares Fitting of Two 3-D Point Sets[J]". *IEEE Trans Pattern Anal Mach Intell*, 1987, 9.

[10] He K , Gkioxari G, Dollar P, et al. "Mask R-CNN[C]" *International Conference on Computer Vision. IEEE Computer Society*, 2017.

[11] Ranftl R, Lasinger K, Hafner D, et al. "Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer[J]." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[12] Li Z, Snavely N. "MegaDepth: Learning Single-View Depth Prediction from Internet Photos[C]" *IEEE CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE*, 2018