

An Occlusion Detection Algorithm for 3D Texture Reconstruction of multi-View Images

Ming Li, Bingxuan Guo, and Weilong Zhang

Abstract—Occlusion detection is one of the key technologies for 3D texture reconstruction of multi-view images. In this paper, an efficient occlusion detection algorithm is proposed aiming at the existence fragments, seams in 3D texture mapping form multi-view images. Firstly, we reconstructed the 3D surface model form multi-view images. Then, based on the projection relationship between object point and image point, we proposed an occlusion detection algorithm based on OpenGL. At last, compared with the algorithm of occlusion detection based on sparse grid, the experimental results show that our method can better solve the problems of occlusion detection in 3D texture reconstruction based on multi-view images.

Index Terms—Z-buffer, OpenGL, sparse grid, multi-view image.

I. INTRODUCTION

In the texture mapping of 3D surface model, to establish the relationship between every grid element (triangular patch) of 3D surface model and its corresponding visible texture images is a fundamental and necessary work [1]. In the analysis of corresponding relationship among them, the Z-buffer algorithm [2], [3] and Ray-tracing algorithm [4], [5] are two representative common solutions. Reference [6] reviewed and evaluated the Z-buffer algorithm proposed by Edwin, and obtained a lot of research results and conclusions. Reference [7] studied on Ray-tracing algorithm, and did many experiments, to solve some problems of itself, deepen the understanding of the algorithm. The Z-buffer algorithm has the dependency of image resolution, specifically, it must distinguish two class resolutions. One is the resolution of visual direction, which depends on the type of value stored in the buffer, such as integer, single precision floating point and double precision floating point, and it limits the ability to distinguish the different surface in close proximity in the visual direction. Another resolution dependency refers to the size and number of pixels in the buffer, which limits the ability to distinguish the vertical overlapping surface in small scope, in the process of Z-buffer, a smaller inclusive interval than the pixel size cannot be detected. In order to solve the above problems, we need to increase above two class resolutions to achieve these goals, but the memory overhead

will be significantly increased. Ray-tracing algorithm determine whether the patch shaded through a light defined by a pixel of the image space and the camera's projection center in the object space. It need to use the camera's interior and exterior orientation elements, when the light project from the projection center to the object space, the visible object is the first object which intersects the light. On the contrary, it is shaded. Although they are simpler and easier to implement GPU acceleration, the processing and operation of them are pixel by pixel, which will lead to the fact that the true geometric properties of objects depend on the size of pixel at a certain resolution in the visibility analysis. Meanwhile, the traditional Z-buffer algorithm and Ray-tracing algorithm are more suitable for the laser point cloud for visible patch selection application with simple 3D model, and to the occlusion detection of 3D model obtained by the multi-view images, it cannot ensure the entity objects can be reconstructed in 3D model due to resolution and other factors, this will lead to the occlusion relationship between the objects in the image and the occlusion relationship of the objects in the reconstructed 3D grid model is inconsistent, at last, the former algorithm will cannot perform good in occlusion detection for 3D texture reconstruction of multi-view images. So, in this paper, we will put forward an efficient occlusion detection algorithm in 3D texture mapping form multi-view images.

II. DATA PREPROCESSING

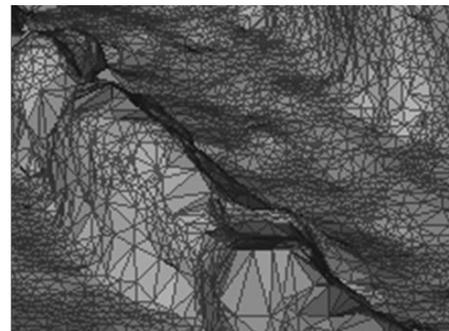


Fig. 1. The local 3D model of experimental Area.

In this paper, the procedure of 3D terrain surface model reconstruction is as followed. Firstly, extract and match feature points in the multi-view images utilizing SIFT (Scale Invariant Feature Transform) [8] combing with calculating the geometrical relationship between the multiple views. Then, calculate the matching result by SFM (Structure From Motion) [9], [10] to get the scene information such as the camera position and then reconstruct sparse by the obtained 3D point cloud. After that, combine the CMVS (Cluster Multi-view

Manuscript received July 24, 2017; revised October 13, 2017. This work was supported in part by the National Key Research and development Program of China (2016YFB0502201 and 2016YFB0502202), the NSFC (91638203), and the State Key Laboratory Research Expenses of LIESMARS.

The authors are with the State Key Laboratory of Information Engineering in Surveying Mapping and Remote Sensing, Wuhan University, Wuhan, Hubei, 430074 China (e-mail: lisouming@whu.edu.cn, 00201550@whu.edu.cn, Zhangweilong@whu.edu.cn).

Stereo) and PMVS (Patch-based Multi-view Stereo) technique to acquire the reconstructed 3D dense point cloud [11], [12]. Ultimately, construct the 3D surface model with high precision used for texture mapping by the optimized triangular algorithm (expressed by TIN). Fig. 1 shows the local 3D model of experimental Area.

III. METHODOLOGY

To determine a triangular patch form the 3D surface model can be visible in which images is an important problem in 3D texture mapping. 3D texture mapping based on multi-view images is not a simple process from theory to realization, due to the limitations of the real environment (hardware, time, etc.), the process must be accelerated from former algorithms, like as pixel by pixel, or grid by grid. In the existing acceleration modes, there are two categories: hardware acceleration and software acceleration (or CPU acceleration and GPU acceleration); software acceleration must be based on the hardware, in this section, we will first introduce CPU acceleration algorithm (occlusion detection algorithm based on sparse grid) [13], and then, without considering the addition of new computing devices, proposed a GPU acceleration algorithm (occlusion detection algorithm based on OpenGL) on the basis of using the additional computing power of GPU.

A. Occlusion Detection Based on Sparse Grid

Occlusion detection occurs in 3D space, how to combine depth information to quickly obtain the occlusion situation form the 2D image is the key of the problem. The basic idea of occlusion detection based on sparse grid is consistent with Z-buffer, but at this time the processing unit becomes a triangular patch, and it introduces the sparse grid to create the index structure of the neighborhood in order to speed up the traversal of the triangular patch on the 3D surface model [13].

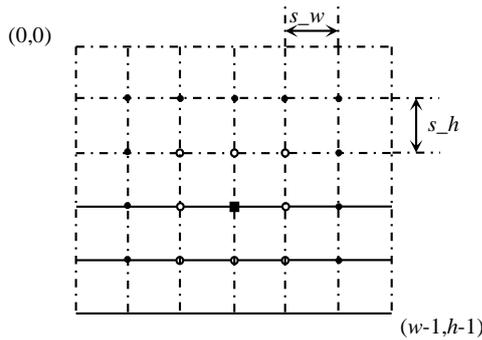


Fig. 2. The sparse grid of image.

Fig. 2 is the index structure of the neighborhood, in the image, the grid spacing is set to $s_w \times s_h$ grid node to record all the mapped triangular patch in its neighborhood. According to the principle of image imaging, occlusion only within the scope of the neighborhood of the point. As shown in fig. 2, triangular patch in \blacksquare is only necessary to determine the depth information of triangular patches in its neighborhood \circ and the node of itself. The occlusion and non-occlusion surfaces can be calculated by the space intersection of computing, specific core operation steps are as follows:

- 1) Find the intersection of the triangular patch, formula (1), (2), (3) are the mathematic expressions.
- 2) To determine whether the point is in a triangular plane, the judge condition is whether the sum of area of three new triangular patches that composed of the point and the any other two points from the triangular vertex is equal to the area of the original triangular patch.
- 3) If the point inside the triangular patch, we can know the occlusion condition through the intersections of multiple objects in the triangular patch produce by connecting the photographic light with the point, and combine with the depth information.

$$P_a = P_1 + u_a(P_2 - P_1) \quad P_b = P_3 + u_b(P_4 - P_3) \quad (1)$$

$$u_a = \frac{(x_4 - x_3)(y_1 - y_3) - (y_4 - y_3)(x_1 - x_3)}{(y_4 - y_3)(x_2 - x_1) - (x_4 - x_3)(y_2 - y_1)}$$

$$u_b = \frac{(x_2 - x_1)(y_1 - y_3) - (y_2 - y_1)(x_1 - x_3)}{(y_4 - y_3)(x_2 - x_1) - (x_4 - x_3)(y_2 - y_1)} \quad (2)$$

$$x = x_1 + u_a(x_2 - x_1) \quad y = y_1 + u_a(y_2 - y_1) \quad (3)$$

where P_1, P_2 and P_3, P_4 are the ends of the straight line segments respectively, and set $P_a = P_b$, P_a, P_b are the intersections on the two lines.

B. Occlusion Detection Based on OpenGL

OpenGL implements the computer simulation of 3D objects, has a powerful function of 3D scene rendering. One most important function of OpenGL is to generate 2D images corresponding to 3D objects on the computer screen. The photogrammetry takes the image as the main research object, extracts the 3D geometric information from the objects by interpreting the images. The perspective projection imaging process of OpenGL is same to the imaging process of photogrammetry in essentially. In addition, the OpenGL implementation process is also a simulation of people watching the real world, everything looks small in the distance and big on the contrary. So, it must implied the process of occlusion detection. Therefore, we can use the rendering process of OpenGL to complete the occlusion detection.

OpenGL undergoes three basic transformations in the process of window displaying with 3D scene, they are model transformation (**M**), projection transformation (**P**) and viewport transformation (**V**), which correspond to three coordinate systems respectively, namely the object coordinate system O_w-XYZ , camera coordinate system O_m-xyz , screen coordinate system O_m-uv , as shown in Fig. 3.

The role of model transformation is to map the object coordinate system to the camera coordinate system, the function of projection transformation is to map the camera coordinate system to unit cube twice as big by eye cone cropping. The viewport transformation is the final transformation to map the 3D coordinates to 2D display screen coordinates. Through these transformation matrixes that can transform the object coordinate system to the screen coordinate system, the mathematical expression is formula (4), and the formula of **M, P, V** is (5), (6), (7) respectively, we can change the object coordinate system (X, Y, Z) to the screen coordinate system (u, v) by these three transformation

matrixes. In these transformations, the perspective projection transformation (\mathbf{P}) is relatively complex, and its basic principle is the similarity proportion transformation.

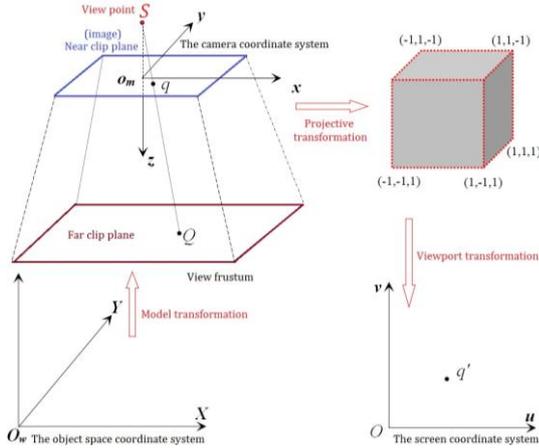


Fig. 3. OpenGL transform procession.

$$[u \ v \ 1]^T = V_{3 \times 4} \cdot P_{4 \times 4} \cdot M_{4 \times 4} [X \ Y \ Z \ 1]^T \quad (4)$$

$$M = \begin{bmatrix} R^T & -R^T T_s \\ 0 & 1 \end{bmatrix} \quad (5)$$

$$P = \begin{bmatrix} \frac{2f}{w} & 0 & -\frac{x_0}{w} \\ 0 & \frac{2f}{h} & -\frac{y_0}{h} \\ -\frac{F+N}{F-N} & 0 & -\frac{2FN}{F-N} \\ 0 & 0 & -1 \end{bmatrix} \quad (6)$$

$$V = \begin{bmatrix} \frac{w}{2} & 0 & 0 & \frac{w}{2} + x_0 \\ 0 & \frac{h}{2} & 0 & \frac{h}{2} + y_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

According to the relationships of between the OpenGL's transformation matrixes and the interior and exterior orientation elements in photogrammetry [14], [15], we can use the photogrammetry information to complete the expression of OpenGL's transformation matrixes, thus using the 3D scene rendering process of OpenGL to achieve a fast, efficient and accurate occlusion detection. Where the model transformation is substantially equivalent to the rotation and translation of the exterior in photogrammetry, \mathbf{R} , \mathbf{T}_s are the rotation matrix and translation matrix in photogrammetry respectively. The projection transformation is equivalent to the 3D clipping + similar transformation + affine transformation, (f, x_0, y_0) is the interior orientation elements, (w, h) is the width and height of the camera. The viewport transformation is equivalent to the 2D Euclidian transformation.

IV. EXPERIMENTAL ANALYSIS

A. Experimental Environment and Data

The experimental computer environment is Windows 7 operating system, with 32G computer memory, Intel core 7. Fig. 4 shows the five-view images of the experimental area.



Fig. 4. Experimental multi-view images.

B. Results Contrast

TABLE I: DATA CONTRAST OF OCCLUSION DETECTION EFFICIENCY

Algorithms	Time Cost (s)	
	Sparse Grid	OpenGL
3117	0.126	0.676
9638	0.231	4.005
19275	0.313	5.917
38552	0.798	9.662
77103	2.375	16.985
154221	8.431	25.155
308454	29.909	39.942
616926	120.249	68.875
1233885	531.941	119.168
2467864	2839.813	213.163

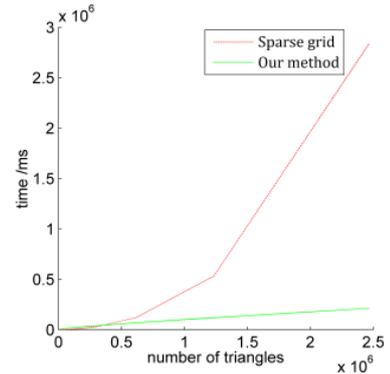


Fig. 5. Contrast curves of occlusion detection efficiency.

We use the occlusion detection algorithm of Sparse Grid based on Z-Buffer [13] as the reference object. The time complexity of traditional Z-Buffer is $O(n^m)$, the improved algorithm of Sparse Grid can be reduced to $O(n)$, here m is the number of average pixels corresponding to each triangle patch, n is the number of triangle patches. For the data set of massive triangular patches in large-scale city scene 3D texture reconstruction, the spatial complexity of the improved algorithm mentioned above is no-uniform linear growth. Fig. 5 shows the comparison result of the Sparse Grid algorithm and our occlusion detection algorithm proposed in this paper. It can be seen from Fig. 5 that although the algorithm is not dominant when the amount of data is small, due to the optimization of OpenGL off-screen rendering detection mechanism and GPU acceleration, the linear steady growth of processing time can be ensure with the increase of the amount of triangular patches. Especially, when the number of triangular facets is more than 300,000 (shown in Table I), the time consumption of the algorithm based on Sparse Grid shows abnormal jump growth with the increasing amount of triangular facets, which reflects our proposed occlusion detection algorithm has a significant advantage of deal with

large data volume.

Fig. 6 shows the effects comparison of before and after the occlusion detection based on OpenGL in the 3D texture reconstruction from multi-view images. It can be seen from the red boxes that the occlusion phenomenon has been effective removal in the right side of the 3D texture model, guaranteed the texture disparity coherence and consistency of the reconstructed 3D texture model.



Fig. 6. Results comparison of occlusion detection.

V. CONCLUSION

In this paper, firstly, we analyzed the existing problems of traditional occlusion detection algorithms of Z-buffer and Ray-tracing in the 3D texture mapping. Then, according to the occlusion problems in 3D texture reconstruction based on multi-view images, we used the relationships between the transformation matrixes of OpenGL and the interior and exterior orientation elements of photogrammetry to propose an occlusion detection algorithm based on OpenGL. At last, compared our algorithm with the reference [13], the experimental results show that the proposed algorithm has higher efficiency and better effects, which means that it is more suitable for 3D texture mapping based on multi-view images.

ACKNOWLEDGMENT

Thanks to Master Xuan Liao for helping the experimental data collection and arrangement, meanwhile, she also participated in part of the experimental work in this paper. And also thanks to Professor Deren Li for his guidance and help on the research contents of this paper. Their help makes this paper can complete in time.

REFERENCES

- [1] Q. Luo, G. Zhou, and G. Zhang, "The texture extraction and mapping of buildings with occlusion detection," in *Proc. the International Geoscience and Remote Sensing Symposium*, 2015, pp. 3002-3005.
- [2] K. Akeley and J. Su, "Minimum triangle separation for correct z-buffer occlusion," in *Proc. the International Conference on Computer Graphics and Interactive Techniques*, 2006, pp. 27-30.
- [3] H. Yamachi, Y. Souma, and Y. Kambayashi, "Evaluation of a technique for collision and object detection with the Z-buffer in cyber space," in *Proc. Cyberworlds*, 2011, pp. 85-92.
- [4] A. Woo, P. Poulin, and A. Fournier, "A survey of shadow algorithms," *IEEE Computer Graphics and Applications*, vol. 10, pp. 13-32, June 1990.
- [5] J. S. Goldsmith and J. K. Salmon, "Automatic creation of object hierarchies for ray tracing," *IEEE Computer Graphics and Applications*, vol. 7, pp. 14-20, May 1987.
- [6] C. Frueh and A. Zakhor, "Construction 3D city models by merging ground-based and airborne views," in *Proc. the 2003 IEEE Computer Society Conference on CVPR*, 2003, pp. 306-313.
- [7] H. Lensch, W. Heidrich, and H. P. Seidel, "Automated texture registration and stitching for real world models," in *Proc. the 8th Pacific Conference on Computer Graphics and Applications*, 2000, pp. 317-452.
- [8] M. A. Brown and D. G. Lowe, "Automatic panoramic image stitching using invariant features," *International Journal of Computer Vision*, vol. 74, pp. 59-73, June 2007.
- [9] M. Goesele and S. Seitz, "Multi-view stereo revisited," in *Proc. Computer Vision and Pattern Recognition*, 2006, pp. 2402-2409.
- [10] J. J. Koenderink and A. J. Van Doorn, "Affine structure from motion," *Journal of the Optical Society of America A-optics Image Science and Vision*, vol. 8, pp. 377-385, Feb. 1991.
- [11] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multi-view stereopsis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 505-512, Aug. 2010.
- [12] J. Schoning and G. Heidemann, "Evaluation of multi-view 3D reconstruction software," in *Proc. Computer Analysis of Images and Patterns*, 2015, pp. 450-461.
- [13] C. Zhang, W. Zhang, B. Guo, J. Liu, and M. Li, "Rapidly 3D texture reconstruction based on oblique photograph," *Acta Geodaetica et Cartographica Sinica*, vol. 44, pp. 782-790, July 2015.
- [14] Z. Zhang, G. Su, and S. Zhen, "Relation OpenGL imaging process with exterior and interior parameters of photogrammetry," *Geomatics and Information Science of Wuhan University*, vol. 29, pp. 570-574, Jul. 2004.
- [15] S. Wei and Z. Nan, "OpenGL simulation for 3-CCD target positioning system," *Laser & Optoelectronics Progress*, vol. 51, pp. 368-379, Oct. 2014.



Ming Li was born in Hubei, China. He received the Ph.D. degree from Wuhan University, Wuhan, Hubei, China in 2016. His research interest includes computer vision, Photogrammetry, indoor vision positioning and geographic information application.

He is currently a postdoc researcher at State Key Laboratory of Information Engineering in Surveying Mapping and Remote Sensing of Wuhan University, and also an assistant researcher at collaborative innovation center of geospatial technology. He participated in three national and provincial scientific research projects, and he has also presided over the independent research project of doctoral students.

Ph.D. Li was awarded the national scholarship by ministry of education of the people's republic of China and the annual outstanding graduate student by Wuhan University, and also awarded aerial photogrammetry scholarship of Microsoft by Wuhan University and NavInfo incorporation, and so on.



Bingxuan Guo was born in Hebei, China. He received the Ph.D. degree from Wuhan University, Wuhan, Hubei, China in 2001. His research interest includes computer vision, Photogrammetry, 3D multi-view geometry and geographic information application.

He is currently a professor at State Key Laboratory of Information Engineering in Surveying Mapping and Remote Sensing of Wuhan University, and also a researcher at collaborative innovation center of geospatial technology.

Prof. Guo was awarded five provincial science and technology awards in China, and has undertaken several national and provincial scientific research projects.



Weilong Zhang was born in Shanxi province of China. He received the M.S. degree from Xidian University, Xian, Shanxi, China in 2016. His research interest includes 3D reconstruction, Photogrammetry, image processing and geographic information application.

He is currently a Ph.D. candidate at State Key Laboratory of Information Engineering in Surveying Mapping and Remote Sensing of Wuhan University.