RESEARCH ON 3D BUILDING VISUALIZATION BASED ON TEXTURE SIMPLIFICATION AND FRACTAL COMPRESSION

Xin Bao [1] , Guoqing Zhou [1], Tao Yue [1, *], Haoyu Wang [1], Wenxi Zhang [1] , Yongsheng Huang [1], Jinsheng Ni [2]

[1] Guangxi Key Laboratory of Spatial Information and Geomatics, Guilin University of Technology, No. 12 Jian'gan Road, Guilin, Guangxi 541004, China
[2] Spatial Data Service Center ,China Aerospace Science and Industry Corporation, No.8 Fucheng Road, Haidian District , Beijing 100048, China

KEY WORDS: Urban Architecture, Fractal Quadtree, Texture Compression, Texture Simplification, Texture Mapping, Visualization

ABSTRACT:

There are some problems in the visualization of 3D model of buildings at present, such as redundant texture data and large memory occupation of loading texture data, which pose challenges to the smooth loading of 3D building visualization. In this paper, we propose a method which using fractal quadtrees to simplify texture data and compress management. Firstly, this method uses the fractal self-similarity and the feature that the same texture has the same fractal dimension to screen all the textures, when the fractal dimension of multiple textures is within the threshold range. Radon transform will be performed on these textures to calculate their standard deviation and further simplify the texture. Secondly, we use fractal texture compression to create a multi-resolution texture data structure with quadtree structure, so that the building 3D model can achieve dynamic visualization of different scales. Finally, the parallel texture mapping of all faces of the 3D model of the building is implemented according to the texture calling rules. Experimental results demonstrate that the texture is managed by the method proposed in this paper, which shortens the loading time of texture and reduces the memory occupation by about 48.92%. Therefore, the method of texture simplification and compression proposed in this paper has a significant effect in the process of building 3D model visualization.

[*] Corresponding author: Tao Yue; E-mail: yuetao@glut.edu.cn

1. INTRODUCTION

With the increasing demand for 3D scenes, it is important to quickly manage and call fine textures for the establishment of high-quality 3D models. However, the finer the texture, the more the loading process will affect the speed of the program, especially when the number of texture images is very large. The research goal of this paper is to solve the problem of how to quickly manage and call textures and quickly realize the three-dimensional visualization of multi-scale urban buildings.

3D building model data is commonly simplified by the method of compression (Salomon, 2000) at present, but the simplified model will lose many feature points of the building (Mao and Ban, 2013), and the rapid visualization of large-scale buildings cannot be achieved by building simplification alone. At the same time, many researchers have proposed architectural generalization methods (Sester, 2000; Fan and Meng, 2012; Zhao, Zhu ,2012), which reduce the call data and improve the rendering speed by using multi-detail level (LOD) technology. Most building model generalization methods focus on the geometry of the model, and textures are rarely considered. However, Khan and Okuda (2006) believe that texture data takes up a lot of memory space compared to building geometry data. In the dynamic visualization process of urban 3D models, different perspective distances require different LOD models, so multiple data structures are needed to store textures of different resolution scales. At present, the storage and organization methods of texture data are mainly based on the spatial segmentation method and the object bounding box method to generate an index. The former method generally uses a top-to-bottom split to generate a scene segmentation tree, such as a quadtree, a KD tree, etc. (Zhou, D. and Z. Ye, 2010). These methods are stored in separate textures. Each texture image must first complete the data parsing, find the associated information, and finally read the texture data of the corresponding format. Smith J (1994) proposed a quadtree decomposition method based on wavelet feature. The algorithm uses quadtree decomposition method to extract texture features from the scale hierarchy of each image, and finally extracts uniform texture blocks for Database index. In 2014, Zhou et al. (Zhou, 2004; Zhou and Xiang, 2014) inserted LiDAR raw data into a Grid Digital Surface Model (DSM), using interpolation methods to flexibly add other building attributes such as texture information and wall information. This method can quickly and accurately obtain 3D building data, but it is too expensive to adopt this method for buildings, and the blind images generated by laser scanning need to be integrated with other data sources. Al-Saidi et al. (2016) proposed a new method combining Fractal Dimension (FD) with FIC scheme. This method can classify images according to the texture of the image in the database, and reduce the retrieval time of the query image. Many scholars at home and abroad have done a lot of research on this (Sander, 2001; Zhou, 2005; Xie and Zhou, 2006; Zhou, 2016). Although some methods (Lempitsky and Ivanov, 2007; Goldlücke , 2014)have been able to create high-reality 3D models, there are still several obvious problems:

(1) After the 3D model texture mapping, it still needs to be updated in real time. If the number of buildings is tens of thousands, the texture reading speed will be slower and the texture mapping time will be longer.

(2) When multiple building texture types and materials have the same part, the storage of texture data will generate redundancy, which increases the space complexity, resulting in reduced computational efficiency.

Therefore, this paper studies the fractal structure and texture of the quadtree organization method, and proposes a fractal quadtree texture management and calling method that can be oriented to complex buildings. This method has a simple process, which can reduce texture and reduce memory consumption, and can quickly manage and call textures, realizing dynamic visualization of 3D models of urban buildings.

2. FRACTAL QUADTREE TEXTURE DATA MANAGEMENT

Only a small part of the acquired oblique image is the target texture required by the building model, and most of the remaining textures can be called redundant textures. When the corresponding relationship between the point coordinates of building model and the texture coordinates of points is established, the target texture region in an image can be determined. When it comes to complex buildings with large scenes, there will be a large number of types of surfaces with the same material and the same texture data. Therefore, this paper proposes a brand-new term for fractal quadtree, that is, using the self-similarity of fractals and the same texture has the same characteristics of fractal dimension, using fractal dimension as the main and standard deviation as the auxiliary to simplify the texture. Then use the fractal theory to compress the texture and establish a quadtree structure of multi-resolution texture data, which can speed up the indexing and calling speed of the texture in the visualization process of the building 3D model.

2.1 Texture Simplification

In the texture management and organization process, to determine whether the two textures are the same, you need to set the classification threshold. Generally speaking, the energy of texture feature set (Smith and Chang, 1994) is selected as the classification threshold. However, in the case that the contrast between the target texture and the texture of other regions is very small, the loss of target texture or misclassification will occur(Yao, 2008). Since the fractal dimension can describe the roughness of the surface of the object and can be used as an effective parameter to distinguish different types of textures to overcome the shortcomings of traditional image segmentation, this feature can be used to set the difference of image fractal dimension as the texture classification threshold. Keller(1989) has confirmed that different textures may have the same fractal dimension, therefore, only the fractal dimension of the original image is not enough to describe the texture features of the texture image. Therefore, it is necessary to screen the texture with fractal dimension threshold. Then radon transform is conducted on the texture within the threshold range to calculate the standard deviation, further simplifying the texture and reducing the memory occupation. The main process of texture simplification is shown in Figure 1, where K refers to the fractal dimension threshold and T to the standard deviation threshold.

2.1.1 Texture Filtering Based on Fractal Dimension: The fractal dimension is used to screen the texture in a large range, and the selected texture has a high similarity, which can speed up the simplification of texture. There are many ways to calculate the fractal dimension of texture image. According to the comparative analysis of several methods in Yao (2008), this paper uses the box method to calculate the fractal dimension of texture image. Think of image I of size $M \times M$ as a surface in three dimensions, the length is M width is M height is L, where L is the pixel level of the image, I is divided into $R \times R$ size grid, and the "height" direction is divided into $R \times L/M$; in the grid of each $R \times R$ divided into, find the maximum pixel value u and the maximum pixel value b, the output from the minimum value to the maximum value, a total of several boxes can be covered, the number of boxes is recorded as $\mathrm{n}(i,j)$; sum the number of boxes for each $R \times R$, denoted as N, $N = \mathrm{sum}(n(i,j))$; in this case, the fractal dimension $D = -\log N / \log R$ is theoretically. When R tends to infinity, R is a finite value, so changing the value of R to find a set of N , applying a linear fit, the slope of the obtained line is the fractal dimension of the texture image.

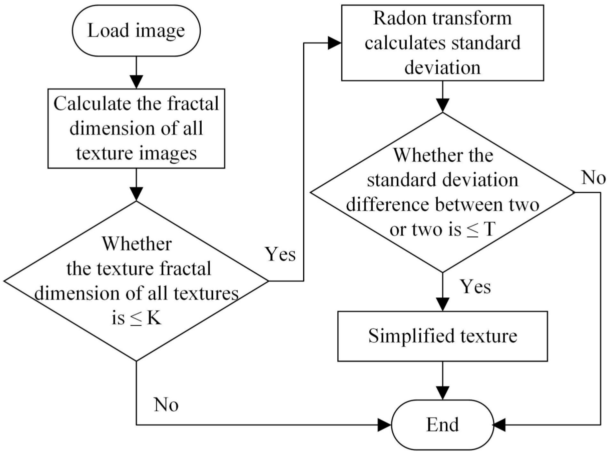

Load image
Calculate the fractal dimension of all texture images
Whether the texture fractal dimension of all textures is ≤ K
Yes
No
Radon transform calculates standard deviation
Whether the standard deviation difference between two or two is ≤ T
No
Yes
Simplified texture
End

Figure 1. Texture simplified flow chart

2.1.2 Texture Simplification Based on Radon Transform: When all textures are first classified by calculating fractal dimensions, set the texture image $\mathrm{f}_1(x,y)$ to rotate the φ angle to get the image $\mathrm{f}_2(x,y)$. According to the nature of the Radon transform, the Radon transform $P_2(\mathrm{r},\theta)$ of $\mathrm{f}_2(x,y)$ is (Wang Xuan, 2008):

$$P_2(\mathrm{r},\theta)=P_1(\mathrm{r},\theta+\varphi)$$

* MERGEFORMAT (1)

Where $P_1(\mathrm{r},\theta)$ is the Radon transform corresponding to the texture image $\mathrm{f}_1(x,y)$ .The cross-correlation function of $P_1(\mathrm{r},\theta)$ and $P_2(\mathrm{r},\theta)$ is:

$$C(\tau,\mathrm{t}) = \int_0^{2\pi} P_2(\mathrm{r},t)P_1(\theta+\tau,t)d\theta$$
$$=\int_0^{2\pi} P_1(\theta+\varphi,t)P_1(\theta+\tau,t)d\theta$$

* MERGEFORMAT (2)

Where t is time, τ is the angular offset as a function of time, so $\beta=\theta+\varphi$, then $\theta=\beta-\varphi$, $\mathrm{d}\theta = d\beta$, the above formula can be written as:

$$C(\tau,\mathrm{t}) = \int_0^{2\pi} P_1(\beta,t)P_1(\beta+\tau-\varphi,t)d\beta$$

* MERGEFORMAT (3)

It can be seen from equation (3) that for any value of t, when $\tau=\varphi$, the function $\tau=\varphi$ takes the maximum value. In order to reduce the complexity, m t values（t_i，1<i ≤ m）can be randomly selected. Calculate the τ value τ_i corresponding to $C(\tau,\mathrm{t})$ for each t_i , and its mean and variance are:

$$\mathrm{u} = \frac{1}{m}\sum_{i=1}^{m}\tau_i$$

* MERGEFORMAT (4)

$$\sigma^2=\frac{1}{\mathrm{m}}\sum_{i=1}^{m}(\tau_i-u)^2$$

* MERGEFORMAT (5)

It can be seen from equation (3) that if $\mathrm{f}_1(x,y)$ and $\mathrm{f}_2(x,y)$ are the same type of texture, the standard deviation σ will be small, so the decision rule is:

$$\begin{cases} \mathrm{f}_1(x,y) \text{ and } \mathrm{f}_2(x,y) \text{ are the same texture} & \sigma \le T \\ \mathrm{f}_1(x,y) \text{ and } \mathrm{f}_2(x,y) \text{ are different textures} & \sigma > T \end{cases}$$

Where T is the standard deviation threshold and can be determined by the accuracy requirement. The texture within the threshold of the fractal dimension is subjected to the Radon transform described above, and the standard deviation is calculated. If the calculation of the textures is within the standard deviation threshold, the textures are merged to further simplify the textures.

2.2 Texture Image Compression

2.2.1 Fractal-Based Texture Compression: After several textures are simplified, the texture images are compressed by using their self-similarity and self-affine properties. In recent years, people have continuously researched and developed various improved fractal image compression coding methods. Among them, the quadtree partitioning method has become one of the most popular segmentation methods due to its advantages of flexible block, high compression ratio and simple algorithm (Liu, B., 2010). The basic principle is as follows: let the original texture image correspond to the root of the quadtree, and when the matching error exceeds a predetermined threshold, divide the root of the tree into four sub-nodes. And the process is repeated until any one of the images can find a suitable matching block. The process of texture fractal compression is shown in Figure 2. For different viewing angles and distances, texture images of different resolutions can be generated from the compressed texture image.

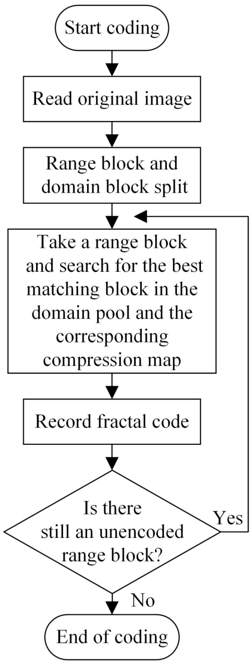

Start coding
Read original image
Range block and domain block split
Take a range block and search for the best matching block in the domain pool and the corresponding compression map
Record fractal code
Is there still an unencoded range block?
Yes
No
End of coding

Figure 2. Flow chart of fractal image compression

2.2.2 Multi-Resolution Texture Call: In order to manage and invoke texture more quickly, this paper adopts quadtree structure to organize texture hierarchy. Therefore, it is necessary to divide texture into layers and blocks according to its resolution size (Li, 2017). Then, the block result of each layer of texture is named and stored by coding. For the texture data of the same layer, the position code corresponding to each block is unique. And location coding actually recorded the attribute information of each block texture, using location coding can be established for each block index, so it is convenient to determine on a hierarchical model of a block of the next layer of the same spatial location of four block, making it easy to build a quadtree structure of texture data organization, as shown in Figure 3.

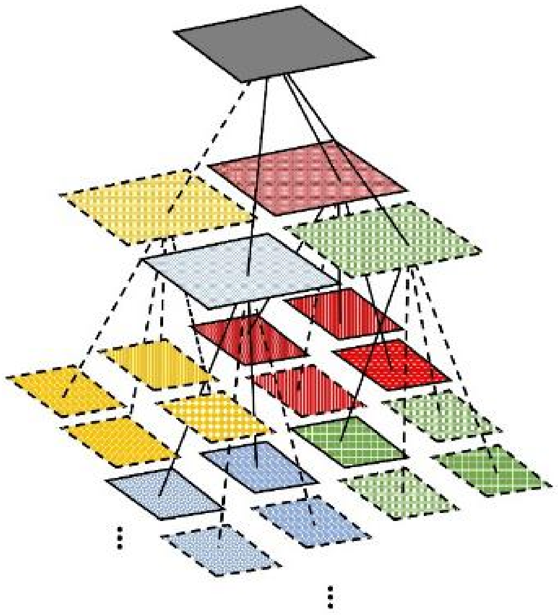


Figure 3. Schematic diagram of fractal quadtree texture

In order to determine the degree of compression of each façade, the distance and the angle of view are considered, mainly based on the distance from the viewpoint to the quadtree node (Figure 4) to select textures with different degrees of detail. The high resolution texture is selected when the viewpoint is closer to the node position, and the lower resolution texture is selected instead. The evaluation function based on the viewpoint distance is as follows (Wang Zhenwu,2018):

$$f = \frac{l}{d \times C}$$

* MERGEFORMAT (7)

The line of sight l is the distance between the viewpoint (x_0, y_0, z_0) and the target node in the quadtree, that is, the center point (x_1, y_1, z_1) , and is:

$$l = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2 + (z_1 - z_0)^2}$$

* MERGEFORMAT (8)

d represents the length of the block at the node; C represents a constant that controls the minimum resolution of the overall scene, and is used to adjust the constant threshold of the accuracy of the hierarchical detail model. The larger C , the more block nodes that need to participate in the current scene drawing, and the higher the resolution of the model of the current scene. An error control threshold τ is set, and a f value is calculated by the equation (7). If $f < \tau$, the block node is subdivided, that is, indexed to the child node of the block node until $f \geq \tau$.

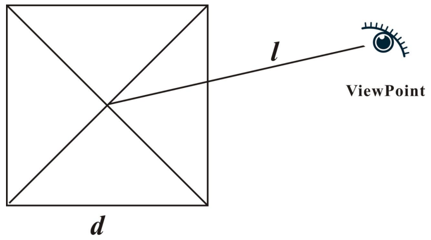


Figure 4. Perspective distance representation

In the process of texture mapping, this paper proposes a method of batch mapping of multiple identical walls with the same texture, that is, classifying the walls according to the texture attributes of building walls and then performing parallel operations on each surface. As shown in Figure 5, for example, the texture of the building is divided into a wall texture and a top texture, and it has a total of N texture attributes, including attributes n1, n2, m1, m2 etc. If there are wt1, wt2, wt4... in the wall texture, they have the same attribute n1. The top texture n1, rt1, rt2... have the same attribute m1. Then it uses texture n1 bindings wt1, wt2, wt4..., using texture m1 to bind rt1, rt2.... Then the texture of type n and type m will perform texture mapping at the same time. It takes parallel operation and the program runs once without saving all the texture data in memory in advance, which can reduce the memory consumption and improve the operation efficiency.

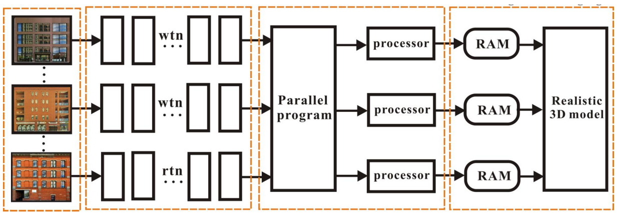


Figure 5. Multi-faceted parallel texture mapping flowchart

3. EXPERIMENT AND RESULT ANALYSIS

3.1 Research Scope and Data

This paper selects the Denver area of Colorado as a research area, which is the financial, industrial, commercial, and transportation center of the Rocky Mountain region. It includes building types with different structures and uses, such as housing, commerce, social culture and infrastructure. Therefore, the density of buildings in the area is large, the buildings are of different heights, the shapes are different, and the structure is relatively complicated. Verify the broad applicability of the texture management methods presented in this paper.

The experimental data used was taken by the RC30 aerial camera. The adjacent images covering the central city are dv1119 and dv1120, which have a 65% heading overlap. In order to demonstrate the advantages of the fractal quadtree proposed in this paper for building texture organization method, the A building is selected as an example in the 92 building models in the study area. The 3D model of the A building without map is shown in Figure 6. Enter 528 texture images, all image formats are .jpg and the size is 512×512. Figure 7 shows the texture of some buildings.

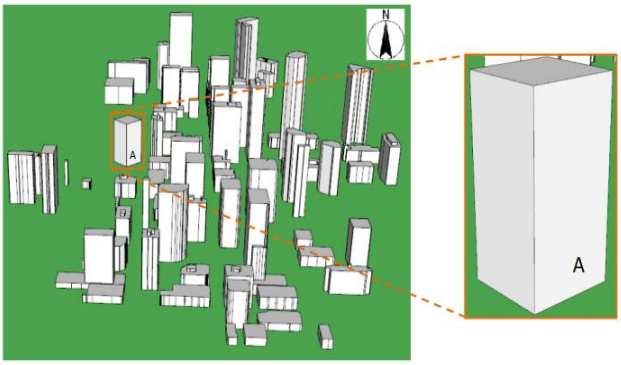


Figure 6. 3D model of an unmapped building

Figure 7. Partial texture of the building

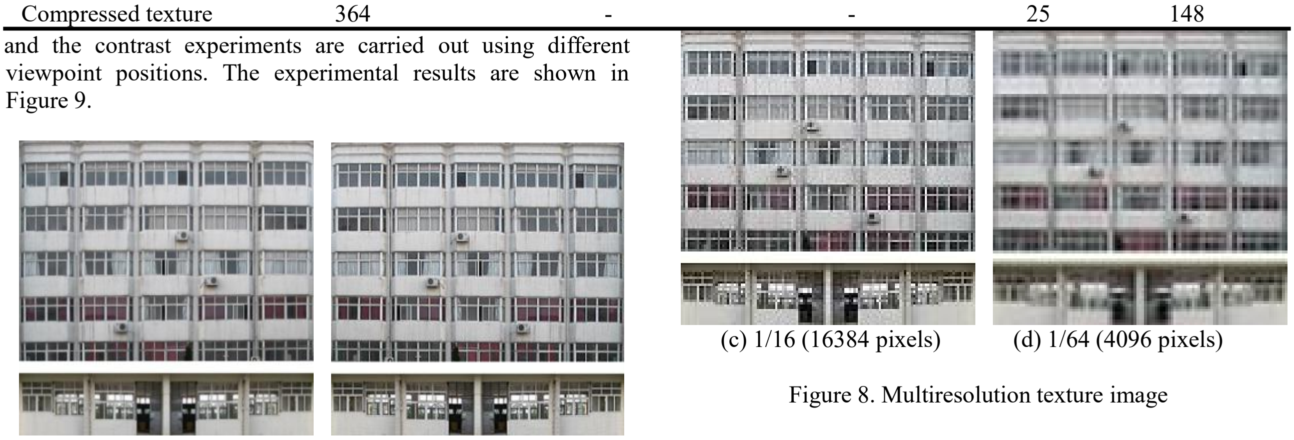
3.2 Analysis of Experimental Results

When we calculate the fractal dimension of 528 texture images, it takes 181.44s. Since the difference of the fractal dimension of the texture image is found to be within 0-0.0015 in many different experiments, the histogram similarity of the texture image is high, so the experiment sets the fractal dimension threshold K to 0.0015 for texture screening; Then, the selected 345 textures are into the Radon transform, and the standard deviation is calculated, which takes 6.45 s. The number of reduced textures is 181, further using the fractal compression texture, the value block size is set to 4×4, the definition domain block size is set to 8×8, the sliding step size δ=8 , and the matching parameters s and o are quantized by 5 bit and 7 bit, respectively. Take one of the faces of the building A (size512×512) as an example to perform texture fractal compression and decoding. The decoded image obtained by iterating four times is shown in Figure 8. The data related to the whole process of texture organization is shown in Table 1.

In order to show the superiority of the proposed method, the southeast, east and west directions of the A building are selected, and the contrast experiments are carried out using different viewpoint positions. The experimental results are shown in Figure 9.

Table 1. Data related to the texture simplification process

Process	Original texture quantity	Filter out the number of textures	Reduce the number of textures	Texture size /M	Time/s
Filter texture	528	345	-	48	210
Streamlined texture	345	-	164	34	12
Compressed texture	364	-	-	25	148



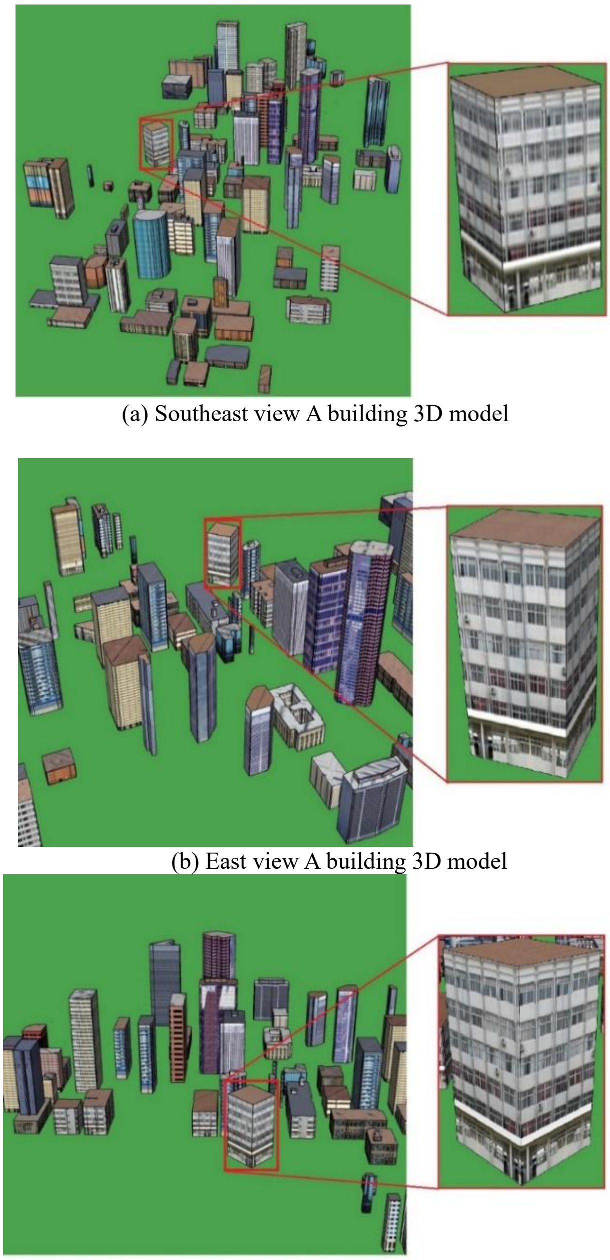
(a)Original (262144 pixels)

(b) 1/4 (65536 pixels)

(c) 1/16 (16384 pixels)

(d) 1/64 (4096 pixels)

Figure 8. Multiresolution texture image



(a) Southeast view A building 3D model

(b) East view A building 3D model

(c) West view A building 3D model

Figure 9. Three-view model diagram

According to Figure 9(a), it can be concluded that when the viewpoint position is in the southeast direction farthest from the A building, the lower resolution texture is selected. Many details in the building are not shown, so the generated 3D model is rough. Figure 9(b) shows that when the viewpoint position is in the east direction far from the A building, because the distance is slightly closer than the southeast direction, some details are displayed, such as the air conditioner outside the window and the curtains of the individual windows are clearly clear. It can be seen from Figure 9(c) that when the viewpoint position is in the west direction closest to the distance A, a higher resolution texture is selected, and substantially all the details are displayed, such as a window frame, a door frame, and various curtains. Obviously, as the line of sight changes, the choice of texture resolution is getting higher and higher, and the generated 3D building model is more realistic and beautiful.

4. CONCLUSION

This paper proposes a new method to realize the dynamic visualization of different scales of 3D models of buildings by using fractal quadtree for texture reduction and management. The method takes advantage of the fractal self-similarity and the same texture has the same fractal dimension, using fractal dimension as the main and standard deviation as the auxiliary to simplify the texture. Then use fractal compression to generate multi-resolution textures and store the data organization forming the quadtree structure, speeding up the indexing and reading speed of the texture according to different viewing angles. The experimental results show that the proposed method organize the texture before the visualization of the three-dimensional model of the building, which can improve the loading speed of the texture. Moreover, it reduces the occupied memory by about 48.92% without affecting the authenticity of the three-dimensional model of the building. And the more abundant the buildings and textures, the more effective the method is.

ACKNOWLEDGEMENTS

This paper is financially supported by the National Natural Science of China under Grant numbers 41431179,41961065; Guangxi Innovative Development Grand Grant under the grant numbers GuikeAA18118038, GuikeAA18242048; the National Key Research and Development Program of China under Grant number 2016YFB0502501 and the BaGuiScholars program of Guangxi (Guoqing Zhou).

REFERENCES

Al-Saidi, N. M. G., S. S. Al-Bundi and N. J. Al-Jawari,2016. A hybrid of fractal image coding and fractal dimension for an efficient retrieval method. *Computational & Applied Mathematics* ,37(2), 1-16. doi.org/: 10.1007/s40314-016-0378-9.

Fan, H. and L. Meng,2012. A three-step approach of simplifying 3D buildings modeled by CityGML. *International Journal of Geographical Information Science* ,26(6), 1091-1107.

Goldlücke, B., M. Aubry, K. Kolev and D. Cremers,2014. A Super-Resolution Framework for High-Accuracy Multiview Reconstruction. *International Journal of Computer Vision*, 106(2), 172-191.

Keller, J. M., S. Chen and R. M. Crownover,1989. Texture Description and Segmentation Through Fractal Geometry. doi.org/: 10.1016/0734-189X (89)90130-8.

Khan, I. R. and M. Okuda,2006: Texture simplification in 3D maps for visualization at low resolution. 2006 International Symposium on Intelligent Signal Processing and Communications, IEEE. doi.org/: 10.1109/ISPACS.2006.364827.

Lempitsky, V. and D. Ivanov,2007: Seamless Mosaicing of Image-Based Texture Maps. IEEE Conference on Computer Vision & Pattern Recognition.

Li Xinwei, Deng Fei, Li Xuefei, Wu Si. 2017. A Visualization Plan for Large-scale 3D Models with Oblique Photogrammetry Technology. *Bulletin of Surveying and Mapping* ,0(4),39-43.

Liu, B. and Y. Ying, 2010.An improved algorithm for image code based on quadtree fractal. *Microelectronics and Computer* ,27(5), 103-105.

Mao, B. O. and Y. Ban,2013. Generalization of 3D building texture using image compression and multiple representation data structure. *Isprs Journal of Photogrammetry & Remote Sensing* ,79(1), 68-79.

Salomo and David, 2000. Data Compression: The Complete Reference. Springer-Verlag New York, Inc.

Sander, P. V., J. Snyder, S. J. Gortler and H. Hoppe,2001. Texture mapping progressive meshes. Proceedings of the 28th annual conference on Computer graphics and interactive techniques, ACM.

Sester, M.,2000. Generalization based on least squares adjustment. *International archives of photogrammetry and remote sensing*, 33(B4/3; PART 4), 931-938.

Smith, J. and S.-F. Chang,1994. Quad-tree segmentation for texture-based image query. Proceedings of the second ACM international conference on Multimedia, ACM.

Wang Xuan, 2008. Research on scaling and rotation invariant analysis methods for images recognition, *Xidian University of Electronic Science and Technology*.

Wang Zhenwu, Lv Xiaohua and Han Xiaohui, 2018. Survey of Terrain LOD Technology Based on Quadtree Segmentation. *Computer Science* ,45 (4), 34-45.

Xie, W. and G. Zhou,2006, Urban 3D building model applied to true orthoimage generation. *1st EARSeL Workshop on Urban Remote Sensing.*

Yao, D., S.Bo, L.Sheng. 2008, Fractal-dimension based algorithm to detect computer graphics. *Communications technology* ,41(12), 256-258.

Zhao, J., Q. Zhu, D. U. Zhiqiang, T. Feng and Y. Zhang,2012. Mathematical morphology-based generalization of complex 3D building models incorporating semantic relationships. *Isprs Journal of Photogrammetry & Remote Sensing* ,68(none), 95-111.

Zhou,D.and Z.Ye,2010. A new large-scale three-dimensional city model real-time rendering texture organization method. Proceedings of 2010 International Conference on Remote Sensing (ICRS 2010) Volume 3.

Zhou, G., W. Chen, J. A. Kelmelis and D. Zhang,2005. A Comprehensive Study on Urban True Orthorectification. *IEEE Transactions on Geoscience & Remote Sensing* ,43(9), 2138-2147.

Zhou, G., C. Song, J. Simmers and P. Cheng, 2004. Urban 3D GIS from LiDAR and digital aerial images. *Computers & Geosciences* ,30(4), 345-353.

Zhou, G., Y. Wang, Y. Tao, S. Ye and W. Wei,2016. Building Occlusion Detection from Ghost Images. *IEEE Transactions on Geoscience & Remote Sensing* ,55(2), 1074-1084.

Zhou, G. and Z. Xiang,2014. Seamless Fusion of LiDAR and Aerial Imagery for Building Extraction. *IEEE Transactions on Geoscience & Remote Sensing* ,52(11), 7393-7407.