

## 3D Scene Management Method Combined with Scene Graphs

AuXiang Wang,<sup>1</sup> Tao Shen,<sup>1\*</sup> Liang Huo,<sup>1\*\*</sup> Congnan Guo,<sup>1,2</sup> and Su Gao<sup>3</sup>

<sup>1</sup>School of Geomatics and Urban Spatial Informatics, Beijing University of Civil Engineering and Architecture, Beijing, 100044, China

<sup>2</sup>Beijing Institute of Geological & Prospecting Engineering, Beijing, 100048, China

<sup>3</sup>Yunnan Provincial Mapping Institute, Kunming, 650034, China

(Received July 21, 2021; accepted November 16, 2021; online published December 13, 2021)

**Keywords:** 3D scene management, adaptive quadtree, LOD, scene graph, hybrid index

The rendering of urban 3D scenes involves a large number of models, where the computer performance becomes a limitation. Arbitrarily putting all the models in a folder for storage significantly reduces the data processing efficiency when the models are called. There are also issues of storage redundancy and semantic fragmentation at the storage boundary. We propose a 3D scene management method based on an adaptive quadtree and scene graph (AQT-SG) that can solve the above problems. According to the spatial distribution characteristics of 3D scene data, this method adopts an adaptive quadtree for organizing complex 3D city scenes at the macro and meso scales, traversing the quadtree from bottom to top and calculating the geometric error at each level and in the middle. The node generates level of detail, builds a flexible multiscale 3D tile model, and uses scene graphs for the microscale organization and management of 3D scenes. We verified the proposed method with park data from a smart park management system. Large-scale complex 3D scene visualization and a comparison of the results of storage redundancy experiments verified that the data organization efficiency was optimized and the visual experience was improved by this method.

### 1. Introduction

Owing to their rapid and advanced urbanization, smart cities have increasingly urgent construction requirements. A smart city is analogous to a human body, where data is the flesh and blood. High-speed data collection and the development of modeling technology have led to a wealth of data products for smart cities, such as large-scale digital surface models, point cloud data, and 3D model data of urban building schemes. Compared with the past, we now have more data with better quality. This has led to the problem of managing massive data.

Three-dimensional urban models usually have the characteristics of wide-range, dense data and complex relationships between models. The characteristics of the data are an uneven spatial data distribution, different local consistency, non-unique attributes of the model, the complicated geometric topology of the model, and a large number and types of textures.<sup>(1)</sup> Using only a basic rendering pipeline for drawing scenes makes it difficult to guarantee real-time accuracy and fluency in 3D urban scene drawing.<sup>(2,3)</sup> Therefore, it is imperative to study the organization and

---

\*Corresponding author: e-mail: shentao@bucea.edu.cn

\*\*Corresponding author: e-mail: huoliang@bucea.edu.cn

<https://doi.org/10.18494/SAM.2022.3563>

management of 3D urban scene data reasonably and effectively to provide strong support for practical applications.<sup>(4,5)</sup>

It has been found that scenes can usually be categorized into two types in scene management: static and dynamic scenes.<sup>(6)</sup> Static scenes mostly use the spatial segmentation structure to organize scene data as this structure is advantageous for the subsequent visibility judgment and collision detection of the scene. Dynamic scenes use scene graphs to organize scene data, where the scene graph structures have more flexibility. Dynamic interactive operations of objects have the advantages of a short update time and high drawing speed. Simultaneously, a scene graph can accurately describe the position of each node in the scene and has a parent-child relationship and clear scene level with each entity indexed. For large-scale geographic scene visualization, more attention is focused on the scene presentation than the scene interaction. Specifically, the visualization of static scenes is the focus of attention. Quadtree division and grid division are mainly used in spatial division structures suitable for remote sensing images and digital terrain data such as quadtree and grid structures.<sup>(7)</sup> Quadtree division refers to the recursive division of a space in accordance with the rules for realizing scene aggregation by region.<sup>(8)</sup> However, in complex 3D city scenes, models often involve a wide range and large amounts of model data with complicated relationships. The quadtree structure does not consider the irregular shape of the buildings in the spatial distribution, which can cause the semantic fragmentation of the elements after segmentation. Grid division does not consider the density of the buildings, resulting in uneven space utilization. Zhu *et al.*<sup>(9)</sup> proposed a partition method based on an adaptive quadtree to reduce fragmentation to buildings at tile junctions. Although this method can partially solve the storage redundancy and semantic fragmentation problems, the inaccurate model spatial indexing due to the complex urban area remains unresolved. An improved scene graph is more suitable for the visual simulation of spatial information systems.<sup>(10)</sup> The main idea is to combine the scene graph and the bounding volume hierarchy (BVH). The group nodes in the scene graph provide the spatial division structure information of the object. This design is conducive for large-scale scene interactions. However, the tree will become increasingly unbalanced, and the efficiency will also decrease with increasing number of editing operations. Su and coworkers proposed a scene management method based on an adaptive binary tree and scene graph,<sup>(11,12)</sup> which limitedly expands the application field of scene organization and improves rendering efficiency. However, the adaptive binary tree is inefficient for organizing large-scale terrain because it overlooks the massive amount of urban 3D scene data in geographic scenes. Liu and Zhu<sup>(13,14)</sup> proposed an interactive tree concept based on dynamic scenes for scene management, which makes full use of object-oriented concepts and octree algorithms conducive for solving complex scene design in a real-time roaming state. However, this algorithm requires more labor and manual intervention by the scene designer, which increases the workload for complex scenes.

In this article, we aim to solve the semantic fragmentation problem of storage boundaries and storage redundancy. The self-adaptive quadtree with a simple structure can limitedly reduce the fragmentation of buildings at tile junctions. The scene graph provides organization flexibility and an accurate description of the position of each node in the scene and the parent-child relationship, where each geographic entity corresponds to a specific root node. It provides a

theoretical basis for solving building fragmentation and storage redundancy issues at node boundaries. Therefore, we propose an adaptive quadtree and scene graph (AQT-SG) 3D scene management method. As per the spatial distribution characteristics for the 3D scene data, this method uses an adaptive quadtree for dividing a complex 3D city scene at the macro and meso scales, traversing the quadtree from the bottom up, generating level of detail (LOD) for the intermediate nodes, building a flexible multiscale 3D tile model, and using scene graphs to organize the scene at the microscale. We have conducted experiments on a smart park management system using park data. In these experiments, the visualization of large-scale and complex 3D city scenes and the microscale use of scene graphs and an adaptive quadtree respectively demonstrate that the proposed method can optimize the data organization efficiency and resolve the issues of storage redundancy and semantic fragmentation of storage boundaries, and improve the overall visual experience.

## 2. Method

### 2.1 Overview of the method in this article

Figure 1 shows the idea of constructing a 3D scene management method combined with scene graphs in this study. We mainly use an adaptive quadtree for dividing vector data and terrain image data at the macro and meso scales, traversing the quadtree from bottom to top, generating LOD for intermediate nodes, and building a flexible multiscale 3D tile model. The child node of the adaptive quadtree is the root node of the scene graph. On the microscale, the scene graph organizes the 3D model data from the city. The adaptive quadtree has the advantages of a simple structure, high spatial efficiency, and high insertion and query efficiency. The same geographic entity is likely to be stored in multiple nodes in the splitting process of the quadtree, which leads to the inefficient use of index storage space. A scene graph is flexibly organized and can accurately describe the location and parent-child relationship of each node in the scene, so that each geographic entity can correspond to a specific child node. The proposed design combines the advantages and avoids the disadvantages of the two methods. This is conducive for rendering large-scale scenes and simultaneously avoiding the building fragmentation and storage redundancy at node boundaries at a microscopic scale.

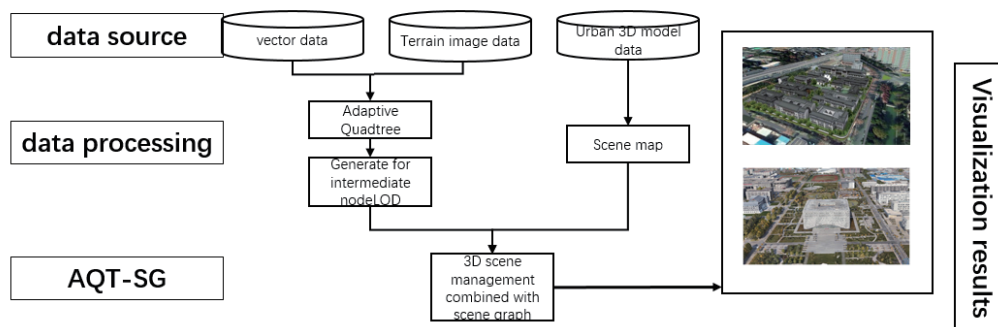


Fig. 1. (Color online) Idea behind AQT-SG method.

## 2.2 Adaptive quadtree

The self-adaptive quadtree structure is simple, and the data aggregation is by region through recursively dividing the space regularly. Therefore, we use an adaptive quadtree for organizing scene data at the macro and meso scales. Figure 2 presents a schematic diagram of adaptive quadtree partitioning.

Figure 3 shows the flow of the algorithm. Specifically, it includes the following four steps:<sup>(15)</sup>

- (1) Carry out the quadtree division of the urban area and judge whether some buildings and tiles intersect the dividing line.
- (2) If yes, appropriately enlarge the enclosing rectangle of the building.
- (3) Calculate all the building information in the four sub-areas.
- (4) Repeat the above steps for the four sub-areas until the number of models in each sub-region reaches the preset threshold.



Fig. 2. (Color online) Illustration of self-adaptive quadtree partitioning.

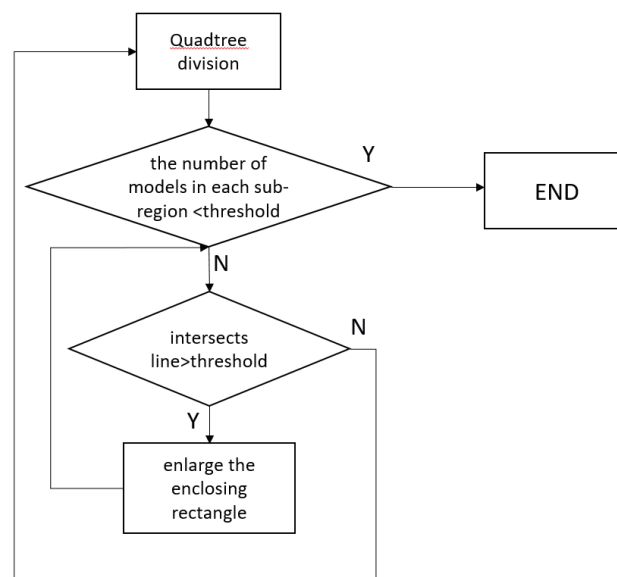


Fig. 3. Adaptive quadtree algorithm.

### 2.3 Scene graph division

A scene graph is a data management method that organizes scene data together in a graphical form. It uses a top-down hierarchical tree-like data structure to organize the spatial dataset and improve rendering efficiency. Moreover, a scene graph has three categories of nodes, namely, root nodes, group nodes, and leaf nodes.<sup>(16)</sup> A scene graph has a flexible organization and can accurately describe each node and its relationships. Consequently, each geographic entity corresponds to a specific root node. In the microscale division, this completely solves the problem of building model fragmentation due to the division of tiles. Unsurprisingly, the issues of model storage redundancy and load redundancy are also resolved.

Figure 4 illustrates the organization and management of a 3D scene adopting a hierarchical management method. This block diagram also shows the hierarchical relationship between objects, where each connected pair of objects has a corresponding hierarchical relationship. The upper object can be used as the reference coordinate system for the next connected object. Simultaneously, the position of the object and the parent node can also be spatially translated and rotated.<sup>(10)</sup>

### 2.4 Method of generating LOD for intermediate nodes

After the adaptive division of the urban area, the next step is to traverse the quadtree from the bottom to the top, generate LOD for the intermediate nodes, and calculate the geometric errors for each level.<sup>(17)</sup> The geometric error is the threshold of the LOD selection during rendering.

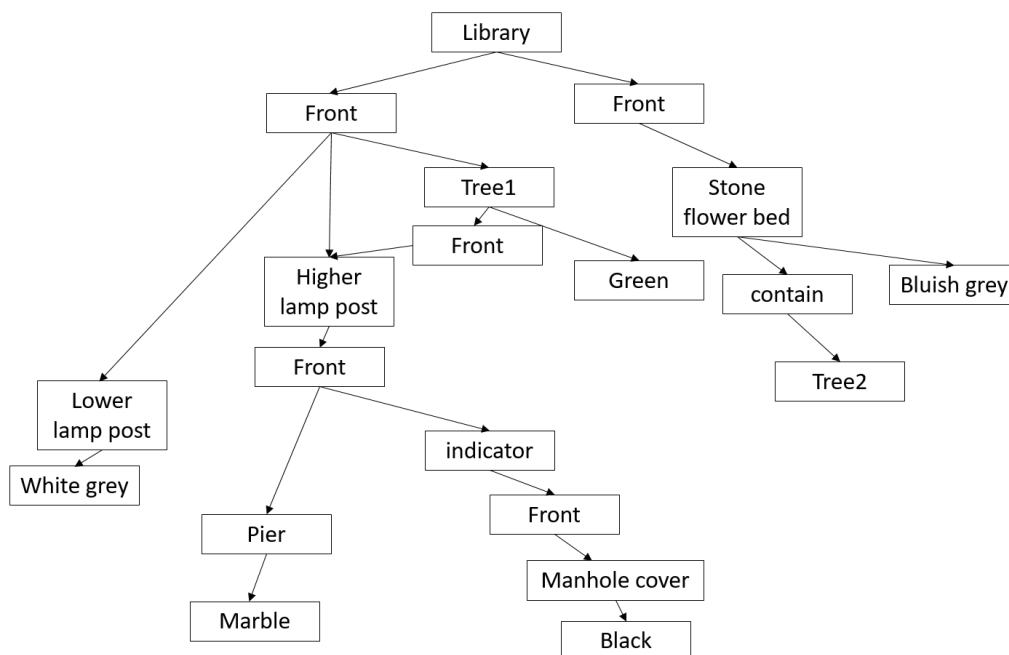


Fig. 4. Hierarchical management diagram of a scene.

The calculation starts at the most detailed leaf node and gradually merges from the bottom up to calculate the geometric error of the previous layer.<sup>(18)</sup> For simplicity, in this study, we assign the same geometric error value to the nodes at the same level in the scene tree.

LOD generation for an intermediate node first requires all the child nodes of the node to be merged and the use of a dynamic mesh simplification method based on a hierarchical structure to simplify the merged model.<sup>(19,20)</sup> The steps in the algorithm are as follows:

- (1) For the current node, first calculate the node error.
- (2) If the current node is a non-leaf node, first merge all child nodes and use a dynamic mesh simplification method based on a hierarchical structure to simplify the merged model.
- (3) Repeat steps (1) and (2) until the root node ends.

Building the scene tree from the bottom up achieves a smoother natural transition between models of different LODs.<sup>(21)</sup> The LOD model accuracy gradually increases from the root node to the leaf node. In online real-time applications, first, a rough model corresponding to the upper node is loaded to quickly obtain rough results, and then a fine model is slowly loaded for the progressive transmission of the 3D model.<sup>(22)</sup>

## 2.5 3D scene management method combined with scene graph

In this paper, we propose a scene organization and management method for improving scene graphs to reasonably and effectively organize and manage scene data. As shown in Fig. 5, the scene has only one scene tree. As per the spatial distribution characteristics of the 3D scene data, the adaptive quadtree divides the complex 3D city scene at the macro and meso scales, transverses the quadtree from the bottom up (after data organization) to generate LOD, builds a flexible multiscale 3D tile model, and uses the scene graph for organization and management at

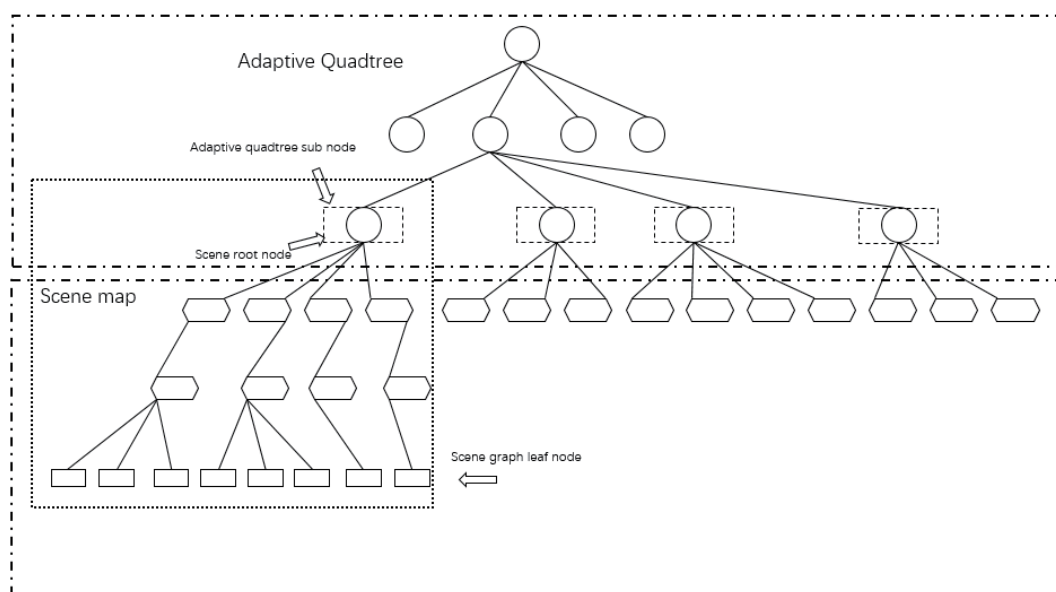


Fig. 5. (Color online) Data structure of the improved scene graph.

the microscale. It also reduces storage redundancy, makes indexing fast and accurate, improves scene rendering efficiency, and enhances visual rendering effects.

### 3. Using AQT-SG Method to Organize Scenes

#### 3.1 Fluency and stability of 3D visualization for urban scenes based on AQT-SG

We use the AQT-SG method to perform large-scale and complex 3D urban scene visualization experiments. Times T1 and T2 in Fig. 6 show the visualization effects of using an adaptive quadtree to organize the 3D city scene at the macro and meso scales. Time T3 shows the visualization effect of using the scene graph to organize the three urban scenes at the microscale. In viewing the roaming visual effects, the proposed method can better maintain the overall and local details, with no apparent stall or delay in the overall roaming process and adequate roaming fluency.

Figure 7 depicts the real-time frame rate when roaming using two different indexing methods, which was obtained by analyzing the change of the frame rate in the whole roaming

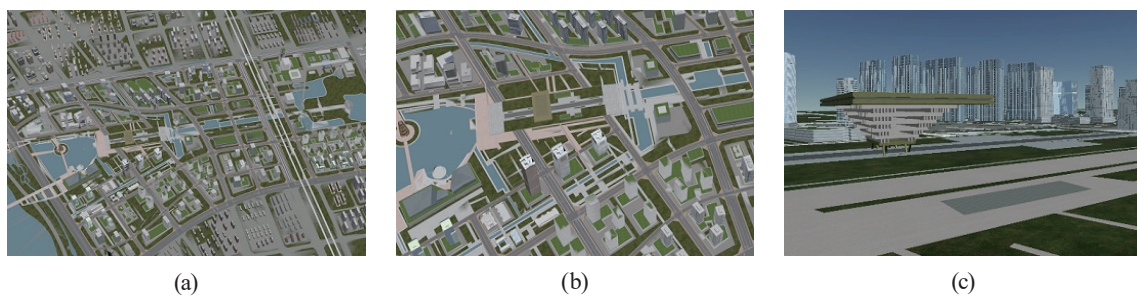


Fig. 6. (Color online) 3D visual effects for urban scenes at different scales. (a) T1, (b) T2, and (c) T3.

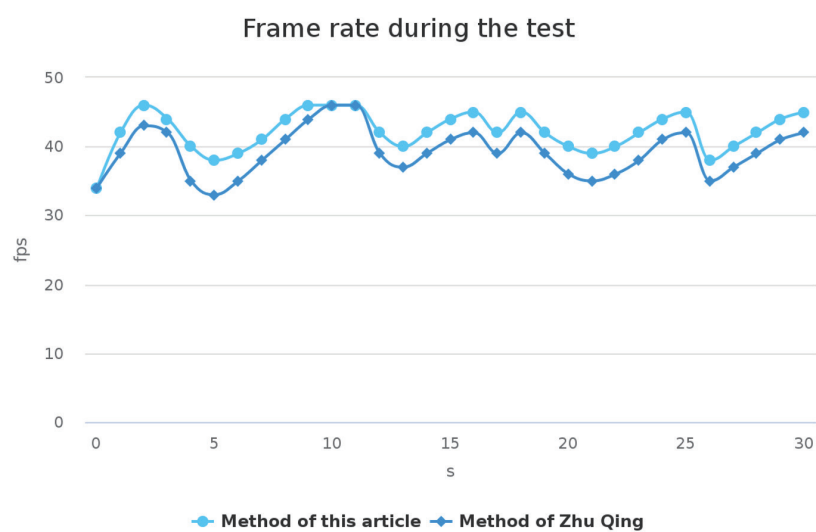


Fig. 7. (Color online) Real-time frame rate when roaming using two methods.

process. Both methods can ensure the smooth operation of the system in the whole roaming process. The rendered frame rate is always maintained between 34 and 46 frames per second (fps). However, compared with Zhu's method,<sup>(9)</sup> our proposed method can achieve a stable rendering state at a higher speed, such as at 0–3 and 5–10 s. This method has better anti-aging performance when the viewpoint moves rapidly, such as at 3–5 and 11–13 s. It can also ensure a better display effect when the viewpoint moves at a uniform rate, such as at 13–24 s. In summary, the proposed method is better than Zhu's method in the roaming process.

### 3.2 Index optimization experiments

We also performed comparative experiments in which the proposed method and the adaptive quadtree method are used for organizing a scene at the microscale, respectively. The proposed method for scene organization is used to verify the indexing efficiency.

The following is an explanation of the two indexing methods.

(1) Adaptive quadtree: Using an adaptive quadtree for scene organization can reduce semantic fragmentation of storage boundaries and storage redundancy to a certain extent. However, there will still be scenarios where the same building spans two nodes. Figure 8 depicts the model storage redundancy problem. The building in the red box of Fig. 8(a) should have been stored in node B on the right and loaded with it [Fig. 8(b)], but node A also stores it owing to the adaptive partition; thus, storage redundancy occurs. When node A is loaded, the two buildings belonging to node B will also be loaded, causing unnecessary computational overhead. For buildings at the boundary of nodes, there will also be a semantic fragmentation problem due to node divisions, i.e., part of the semantics is stored in node A.

(2) Proposed method: When the scene graph is used for scene organization at the microscale, each geographic entity corresponds to a specific child node. Therefore, for the controlled loading, the entity can be controlled instead of the area. As shown in Fig. 9, we can freely control the loading of each building, and a building that loads a node will not affect the surrounding child nodes. This can avoid an inaccurate model spatial index caused by the urban area complexity and fundamentally solve the problem of model storage redundancy, reduce wasteful rendering, improve visualization effects, and enhance user roaming experience.

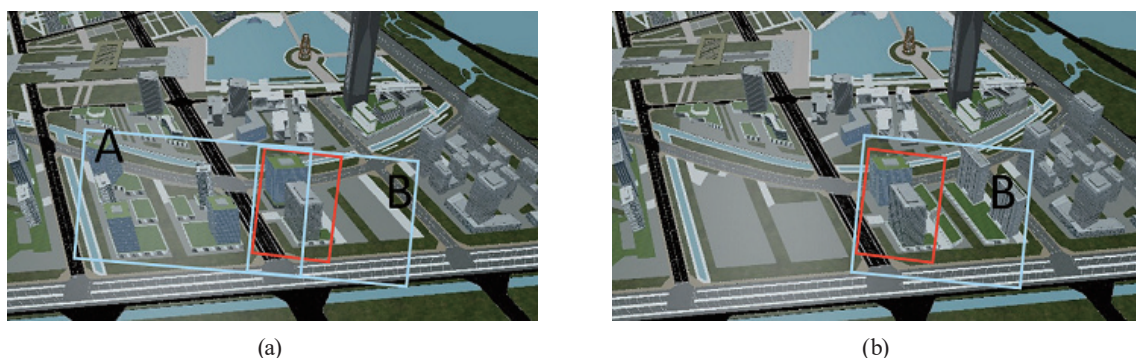


Fig. 8. (Color online) Microscale scene of adaptive quadtree organization.



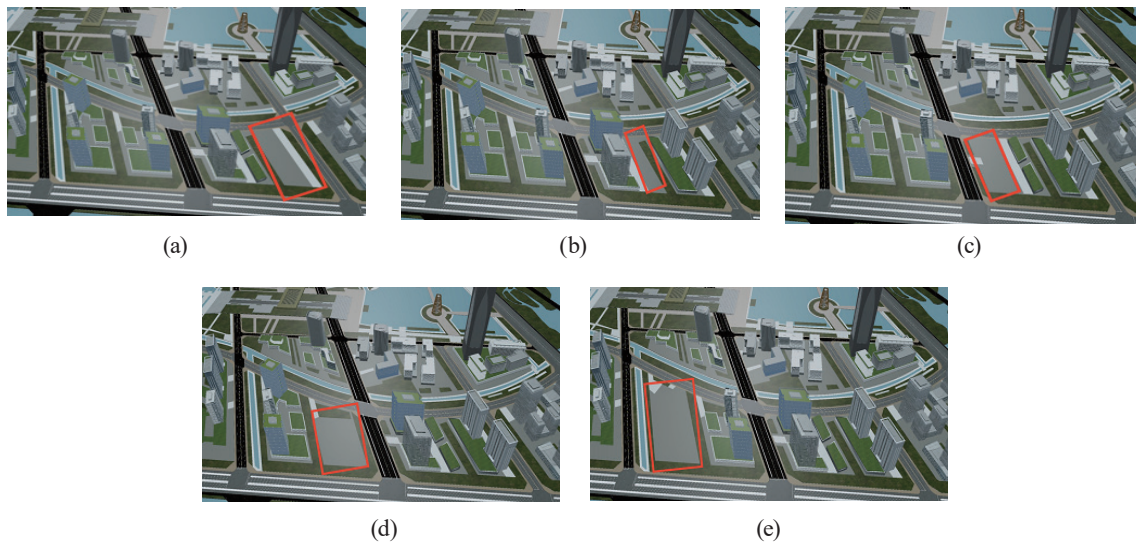


Fig. 9. (Color online) Use of scene graphs to organize microscale scenes.

#### 4. Conclusion

The aim of this study was to increase the efficiency of organizing massive data in large-scale and complex 3D city scenes and reduce redundant storage and fragmented semantics at boundaries for diversified 3D digital city and smart city applications. We studied and proposed an improved scene graph method combined with the adaptive quadtree. In experiments, this method was applied in a 3D visualization platform construction in a Chinese city. For large-scale urban scenes, we selected two scene organization methods based on three scales, i.e., macro, meso, and micro, and designed an overall management method for combining the two scene organization methods, which meets the performance requirements of the system (i.e., fluency and stability) and improves the flexibility of scene management (scene graph).

Experimental results show that the proposed method can better maintain global and local details, maintain stability in switching between different scales, and improve the overall visual rendering effect. Moreover, the superiority of the proposed method was demonstrated in a memory redundancy comparison experiment. However, owing to time constraints, there are still some problems in the proposed design. For example, this design only considers the index problem of 3D city scenes at different scales. No in-depth research on the dynamic scheduling of the hierarchical LOD under a certain scale was performed, especially for roaming in micro scenes. One possible approach to dynamic scheduling is to schedule different levels of LOD models by using the indexing advantage of scene graphs at the microscale and combining the viewpoint and moving direction.

## Acknowledgments

This work was supported by the National Key R&D Program of China (Project No. 2020YFD1100204).

## References

- 1 M. Sun, A. Ma, and J. Chen: *J. Remote Sens.* **2** (2002) 155. <https://doi.org/10.3321/j.issn1007-4619.2002.02.014>
- 2 T. Akenine-Möller, E. Haines, and N. Hoffman: *Real-Time Rendering* (CRC Press, 2019).
- 3 A. Bhojan, Z. Qiang, and A. L. Akkihebbal: *Proc. 3rd ACM Int. Workshop Interactive Multimedia on Mobile & Portable Devices* (2013) 37–42.
- 4 R. Neuville, J. Pouliot, F. Poux, and R. Billen: *Remote Sens.* **11** (2019) 236. <https://doi.org/10.3390/rs11030236>
- 5 C. Erikson, D. Manocha, and W. V. Baxter III: *Proc. 2001 Symp. Interactive 3D Graphics* (2001) 111–120.
- 6 C. A. Astudillo and B. J. Oommen: *Pattern Recognit.* **47** (2014) 96. <https://doi.org/10.1016/j.patcog.2013.04.012>
- 7 L. Wu, Y. Liu, and J. Zhang: *Method and Application* (2001) (in Chinese).
- 8 Z. Wang, X. Lv, and X. Han: *Comput. Sci.* **5** (2018) 4 (in Chinese).
- 9 Q. Zhu, X. Chen, Y. Ding, Min. Liu, H. He, W. Yang, and L. Chen: *J. Southwest Jiaotong Univ.* **52** (2017) 869 (in Chinese).
- 10 L. Zhao: *Scenario Management Based on Scenario Diagrams*, Yanshan University (2010) (in Chinese).
- 11 M. Su, R. Guo, H. Wang, S. Wang, and P. Niu: *Proc. 2017 IEEE Int. Conf. Information and Automation (ICIA) (IEEE, 2017)* 838.
- 12 M. Su, R. Guo, H. Wang, and S. Wang: *Proc. 2017 IEEE Int. Conf. Information and Automation (ICIA) (IEEE, 2017)* 838–842.
- 13 Y. Liu and C. Zhu: *Comput. Appl.* **11** (2001) 7 (in Chinese).
- 14 M. Su, R. Guo, H. Wang, and S. Wang: *Proc. 2017 IEEE Int. Conf. Information and Automation (ICIA) (IEEE, 2017)* 35–40.
- 15 S. Zhou: *Image Hybrid Modeling and Visual Lossless Rendering Technology for Large-scale Urban Scenes*, Ocean University of China (2014) (in Chinese).
- 16 Y. Xue: *Design and Implementation of Compiler for 3d Visualization Running Service Platform in Data Center*, University of Chinese Academy of Sciences (Shenyang Institute of Computing Technology, Chinese Academy of Sciences) (2017) (in Chinese).
- 17 J. Wu: *Comput.-Aided Des.* **2** (2018) 72.
- 18 J. Zhang and Z. Yao: *J. Southwest Jiaotong Univ.* **48** (2013) 666 (in Chinese).
- 19 M. Guthe, P. Borodin, and R. Klein: *4th Int. Conf. Virtual Reality and Its Applications in Industry* **5444** (2004) 428.
- 20 Y. Wang, J. Zheng, and H. Wang: *Sci. Programm.* **2019** (2019). <https://doi.org/10.1155/2019/4926190>
- 21 K. Bahirat, C. Lai, R. P. McMahan, and B. Prabhakaran: *Proc. 8th ACM on Multimedia Systems Conf.* (2017) 50–61.
- 22 J. Gong, S. Ke, Q. Zhu, and Y. Zhang: *Acta Geodaetica et Cartographica Sinica* **44** (2015) 570. <https://doi.org/10.11947/j.AGCS.2015.20130520>

## About the Authors



**Xiang Wang** received his B.S. degree in surveying and mapping engineering from Beijing University of Civil Engineering and Architecture in 2017, where he is currently pursuing his M.S. degree in architectural heritage and protection. His research interests include 3D geological information visualization. (18511053719@163.com)



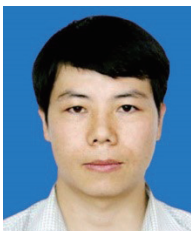
**Tao Shen** graduated from Beijing Normal University with a doctorate in science in 2007. At present, he is mainly engaged in the teaching and research of remote sensing and geographic information, and his research directions include remote sensing information processing, ecological environment remote sensing, spatial information analysis, and spatial information system evaluation. He has written one monograph and won three provincial and ministerial science and technology progress awards. (shentao@bucea.edu.cn)



**Liang Huo** is an expert in spatial information system software evaluation of the Remote Sensing Center of the Ministry of Science and Technology. At present, he is mainly engaged in the research and teaching of cartography and geographic information engineering, and his research direction is UNINFO, a geographic information system software platform with independent intellectual property rights. (huoliang@bucea.edu.cn)



**Congnan Guo** received his B.S. degree in surveying and mapping engineering from Beijing University of Civil Engineering and Architecture in 2017, where he is currently pursuing his M.S. degree in architectural heritage and protection. His research interests include data processing and trend analysis of land subsidence. (18612968148@163.com)



**Su Gao** received his B.S. degree in cartography from Wuhan University in 1999. He is currently a senior engineer of Yunnan Provincial Mapping Institute, where he is mainly engaged in geographic information systems and mapping. (649950505@qq.com)