

The Interactive Modeling Method of Virtual City Scene Based on Building Codes

Wei-long Ding*, **Xiao-jie Zhu**, **Bin Xu**, **Yan Xu**, **Kai Chen**, and **Zang-xin Wan**

College of Computer Science & Technology
Zhejiang University of Technology
Hangzhou, 310026, China

[e-mail: wlding@zjut.edu.cn, zxj_zhuxiaojie@163.com, xubin@qq.com, xuyan@163.com,
rhzhckhs@163.com, 593735243@qq.com]

*Corresponding author: Wei Long Ding

*Received August 28, 2020; revised October 13, 2020; accepted November 17, 2020;
published January 31, 2021*

Abstract

For higher-level requirements of urban planning and management and the recent development of "digital earth" and "digital city", it is urgent to establish protocols for the construction of three-dimensional digital city models. However, some problems still exist in the digital technology of the three-dimensional city model, such as insufficient precision of the three-dimensional model, not optimizing the scene and not considering the constraints of building codes. In view of those points, a method to interactively simulate a virtual city scene based on building codes is proposed in this paper. Firstly, some constraint functions are set up to restrict the models to adhere to the building codes, and an improved directional bounding box technique is utilized to solve the problem that geometric objects may intersect in a virtual city scene. The three-dimensional model invocation strategy is designed to convert two-dimensional layouts to a three-dimensional urban scene. A Leap Motion hardware device is used to interactively place the 3D models in a virtual scene. Finally, the design and construction of the three-dimensional scene are completed by using Unity3D. The experiment shows that this method can simulate urban virtual scenes that strictly adhere to building codes in a virtual scene of the city environment, but also provide information and decision-making functions for urban planning and management.

Keywords: Building Codes, Virtual City Scene, Interactive Modeling Method, 3D Model, Virtual Reality

1. Introduction

With the rapid development of virtual reality (VR) technology, three-dimensional digital models have emerged, and play a vital role in many fields such as medical treatment [1, 2], militaries [3], games [4], and urban planning [5, 6]. Three-dimensional digital city models have the ability to perform perspective, lighting calculation and three-dimensional measurement. For example, the planning solution can be reproduced intuitively, the distribution of urban areas can be planned rationally, and facilities for fire prevention and emergency response can be arranged. Moreover, for higher-level requirements of urban planning and management and the recent development of "digital earth" and "digital city", it is urgent to establish protocols for the construction of three-dimensional digital city models. It is still difficult to build a three-dimensional city model which meets the need of various aspects of city management and planning using the current digital technology. Therefore, efficient and high-quality 3D scene modeling technology is required to realize the digital construction of urban scenes.

In recent years, city scene modeling has become a hot topic in the research for a digital city. Takaichi et al. [7] developed a 3D urban map based on 2D GIS and built a large-scale three-dimensional city model, but Sun Min et al. [8] believed that this model only could express regular buildings, and it was difficult to construct complex urban scenes. Gan et al. [9] constructed 3D scenes based on the outlines of 2D buildings, but the accuracy of the model needs to be improved. Yuan et al. [10] proposed a method to create a traffic scene based on the technology of scene stages, and it would be better if the semantic details of the scene were enriched. Sinha et al. [11] built an image-based interactive system for modeling architectural scenes. But there was no mechanism in the system to correct the image sequences for color balance, so there were visual defects in the image-based textures. Liang Jing et al. [12] used a construction method based on the oblique images from an unmanned aerial vehicle (UAV) and reconstructed a digital campus. But the details of the model were poor and need to be repaired manually. In general, the methods proposed by these scholars still have some problems, such as the inability to construct complex urban scenes and insufficient precision of the three-dimensional model.

Zhou Shengchuan [13] proposed a method to build a large-scale scene which combines inverse procedural modeling with the technique of level of detail (LOD). Nevertheless, this method was not able to model irregular buildings. Parish [14] generated three-dimensional urban roads based on the principle of L-systems, but it was difficult to generate the axiom and productions suitable for the building codes. Zhou Changjiang [15] used the mixed data model of digital elevation model (DEM) and MultiPatch to construct a three-dimensional landscape. This method did not take into account the constraints of building codes, nor did it optimize the scene. Ren et al. [16] proposed an automatic layout optimization method for outdoor scenes, and implemented a system of 3D model retrieval and assembly based on sketches. Such approaches, however, neglected the constraints of building codes. To summarize, although the digital technology for three-dimensional city modeling has made considerable progress, some problems still exist including insufficient precision of the three-dimensional model, not optimizing the scene, and especially not considering the constraints of building codes.

Building codes, as national mandatory standards in city management and planning, provide guarantee for the quality of buildings. In the real environment, if a building does not meet national standards, such as the distance between the building and the road being less than the prescriptive distance, it will cause some problems such as traffic safety and noise. The virtual city scene should also comply with building codes. For non-professionals, if they do not

follow building codes, the virtual city scenes they construct will not meet national standards. It will not only fail to achieve the desired effect but also bring a lot of unnecessary operations for future scene modification. Therefore, how to make building codes as constraints and integrate it into the modeling of the three-dimensional virtual city scene to effectively guide model layouts has practical significance.

Aiming at the above-mentioned problems, a method to interactively simulate a virtual city scene based on building codes is proposed in this paper. Some special functions are designed to place restrictions from the building codes on the city models. To solve the problem that geometric objects may intersect in a virtual city scene, the technique of directional bounding box is improved. We propose a strategy to convert the two-dimensional layouts to a three-dimensional urban scene, and use a Leap Motion hardware device to interactively place the 3D models in a virtual scene. The method we proposed can provide visual information and also will be helpful for decision-making in urban planning and management.

2. Methodology

The virtual city scene is not only a virtual three-dimensional scene but also the mapping of a city on the computer. It can assist related personnel to complete city planning and city management by establishing virtual city scenes. Therefore, it is a problem to be solved to complete the construction of virtual urban scenes through somatosensory interaction.

In this paper, a system to interactively simulate the virtual city scene by using Unity3D is developed. The system flow chart is shown in [Fig. 1](#). The first step is to process pictures collected on the spot and city maps obtained from Baidu map. Unity3D and Photoshop are used to obtain terrain blocks and texture maps. Many elements in the urban scene, such as building models, tree models, street models and other models, should be constructed by using 3dsMax. Then they should be stored in the database after format conversion. Secondly, invoking corresponding diagrams through the user's choices to complete designs of two-dimensional layouts. And invoking related files in the database to complete the rendering of the three-dimensional virtual city scene. During the process of modeling the scene, the constraints of the model will be displayed in the interface, and collision detection will also be performed. A Leap Motion hardware device can be used to position and rotate three-dimensional models in a virtual scene. In the end, aesthetic knowledge is used to optimize the three-dimensional virtual city scene. After the above steps are completed, the three-dimensional virtual city scene constructed by the user will be generated.

For this system, an interactive test is designed to allow users to participate in two ways: mouse and keyboard interaction and Leap Motion interaction. Participants were aged 22 to 28. Among them, there were eleven players playing 3D games for more than two years, and there were eight people who had somatosensory interactive gaming experience. Twenty-four participants were required to complete the translation and rotation of 3D models in the above two ways. The time would be recorded when the participant completed the task and gave a stop gesture. Each user evaluates this system five times. These tests showed that all these participants completed the interactive modeling tasks successfully. The average time spent using the method of Leap Motion interaction was 252s, and the average time spent using mouse and keyboard interaction was 378s. Therefore, we could conclude that interaction based on Leap Motion is more effective.

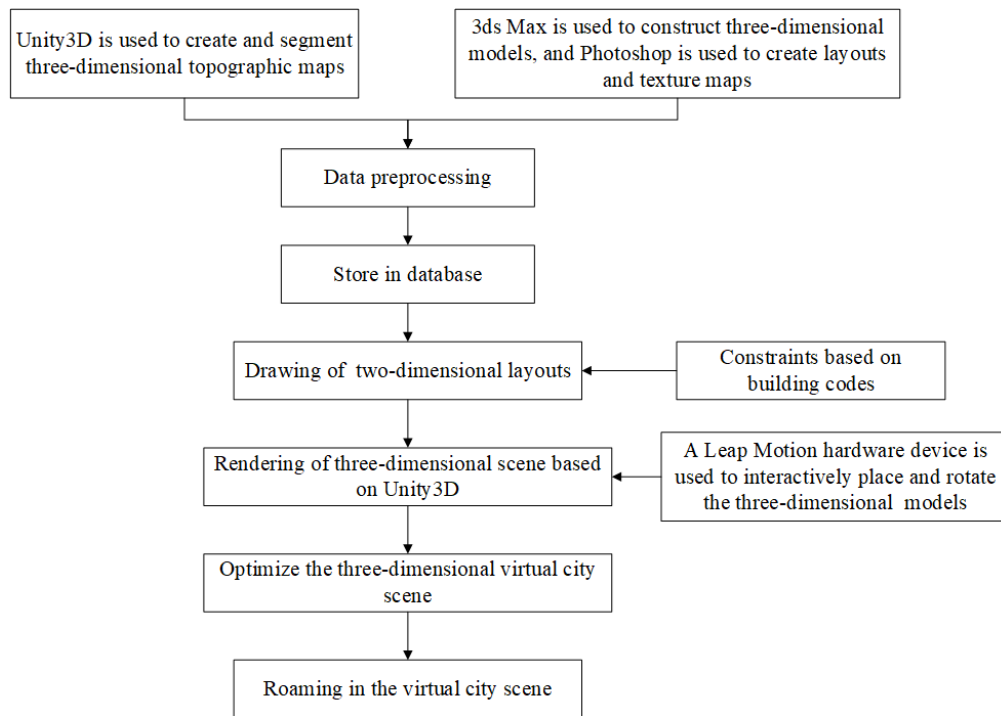


Fig. 1. System flow chart

3. Layout constraints of 3D city models based on building codes

There are many elements in a virtual city scene, such as building models, street models, tree models, etc.. Among them, models and layouts are the most critical elements, so managing them is a basic problem. Relational databases are used to store and manage models and layouts.

In the simulation of the virtual city scene, users often place models at will, which will violate building codes. As national mandatory standards, building codes have played a guiding role in urban planning. A city that complies with building codes can keep the elements of the city coexisting while the city expands. Whether a simulated 3D city scene adheres to the building codes or not is a critical factor to judge the modeling quality of a virtual scene. To be a plausible virtual city scene, the distance between virtual buildings and the distance between virtual buildings and virtual urban roads should also meet the requirements of fire protection, traffic safety, disaster prevention, greening, and engineering construction, as well as relevant requirements formulated by urban and rural planning authorities [17]. So how to restrict the different building models to adhere to the building codes is a key problem that we considered. Collision detection technology of directional bounding box is improved to generate a RealityBox (RB for short) based on the building model and a VirtualBox (VB for short) which is based on building codes, as shown in **Fig. 2** and **Fig. 3**.

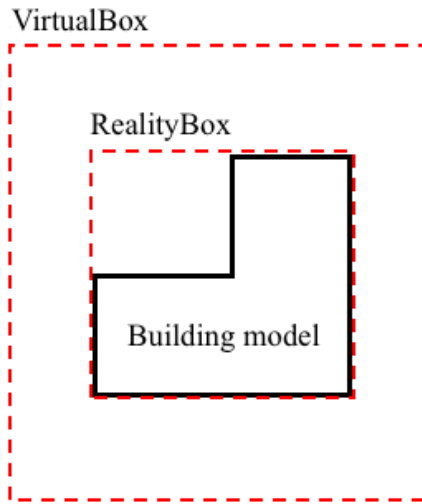


Fig. 2. Schematic diagram of a RealtyBox and a VirtualBox for a building

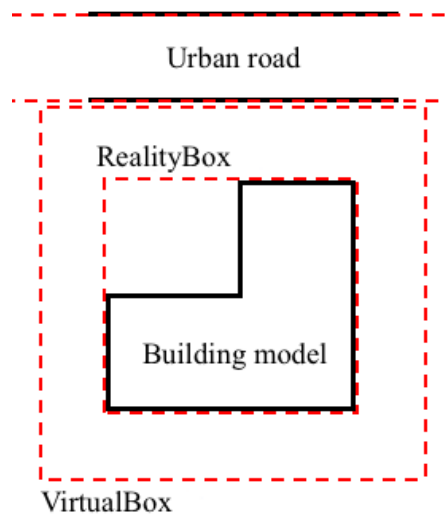


Fig. 3. Schematic diagram of a RealtyBox and a VirtualBox for a building next to a road

Buildings are divided into Low-rise Building (buildings less than or equal to 10 meters in height), Mid-rise Building (buildings greater than 10 meters and less than or equal to 24 meters in height), High-rise Building (buildings greater than 24 meters in height) and Special Building (theaters, playgrounds, shopping malls, etc.) [17]. Urban roads can be divided into Arterial Road (skeleton roads in the urban road network), Secondary Trunk Road (regional arterial roads connected to main roads in urban road networks) and Branch Road (in the urban road network, roads outside the main road that link secondary trunk roads or roads which are used only within a region) [17]. These are key elements of the urban road network which can be used to form an urban road system. We use the setback line to represent the distance between virtual buildings or the distance between virtual buildings and virtual urban roads. In this paper, according to the above-mentioned regulations, seven constraint functions are defined, which constrain the layout of the virtual urban scene from the relationship between buildings and buildings and the relationship between buildings and roads. The definition of each constraint function is given as follows for the example building code we used.

Constraint 1: The setback line of the Low-rise Building should not be less than 3 meters on each side. The formula is

$$VB_l - RB_l \geq 3 \quad (1)$$

where VB_l represents the boundary line of the VirtualBox of the Low-rise Building, and RB_l represents the boundary line of the RealityBox of the Low-rise Building.

Constraint 2: The setback line of the Mid-rise Building should not be less than 7.5 meters on each side. The formula is

$$VB_m - RB_m \geq 7.5 \quad (2)$$

where VB_m represents the boundary line of the VirtualBox of the Mid-rise Building, and RB_m represents the boundary line of the RealityBox of the Mid-rise Building.

Constraint 3: The setback line of the High-rise Building should not be less than 12 meters on each side. The formula is

$$VB_h - RB_h \geq 12 \quad (3)$$

where VB_h represents the boundary line of the VirtualBox of the High-rise Building, and RB_h represents the boundary line of the RealityBox of the High-rise Building.

Constraint 4: The setback line of the Special Building should not be less than 20 meters on each side. The formula is

$$VB_s - RB_s \geq 20 \quad (4)$$

where VB_s represents the boundary line of the VirtualBox of the Special building, and RB_s represents the boundary line of the RealityBox of the Special building.

Constraint 5: The distance between the Arterial Road and the Low-rise or the Mid-rise Building should not be less than 10 meters, but as for the High-rise or the Special Building, the distance between the Arterial Road and them should not be less than 15 meters. The formula is

$$AR_s \geq \begin{cases} RB_l + 10, & \text{Low-rise Building} \\ RB_m + 10, & \text{Mid-rise Building} \\ RB_h + 15, & \text{High-rise Building} \\ RB_s + 15, & \text{Special Building} \end{cases} \quad (5)$$

where AR_s , RB_l , RB_m , RB_h and RB_s represent the boundary line of the bounding box of the Arterial Road, the boundary line of the RealityBox of the Low-rise Building, the boundary line of the RealityBox of the Mid-rise Building, the boundary line of the RealityBox of the High-rise Building and the boundary line of the RealityBox of the Special Building respectively.

Constraint 6: The distance between the Secondary Trunk Road and the Low-rise or the Mid-rise Building should not be less than 7 meters, but as for the High-rise or the Special Buildings, the distance between the Secondary Trunk Road and them should not be less than

10 meters. The formula is

$$SR_s \geq \begin{cases} RB_l + 7, \text{ Low-rise Building} \\ RB_m + 7, \text{ Mid-rise Building} \\ RB_h + 10, \text{ High-rise Building} \\ RB_s + 10, \text{ Special Building} \end{cases} \quad (6)$$

where SR_s is the boundary line of the bounding box of the Secondary Trunk Road.

Constraint 7: The distance between the Branch Road and the Low-rise or the Mid-rise Building should not be less than 5 meters, but as for the High-rise or the Special Building, the distance between the Branch Road and them should not be less than 10 meters. The formula is

$$BR_s \geq \begin{cases} RB_l + 5, \text{ Low-rise Building} \\ RB_m + 5, \text{ Mid-rise Building} \\ RB_h + 10, \text{ High-rise Building} \\ RB_s + 10, \text{ Special Building} \end{cases} \quad (7)$$

where BR_s is the boundary line of the bounding box of the Secondary Trunk Road.

4. Improved collision detection technology of directional bounding box

We now consider the problem of arranging buildings in space subject to these seven constraints. To solve the problem that geometric objects may intersect with each other in a virtual city scene, an improved collision detection technique using directional bounding boxes is proposed in this paper.

Suppose the set of bounding boxes is $N = \{N_1, N_2, \dots, N_m\}$, which needs to satisfy the formula $N_i \cap N_j = \emptyset$ if $i \neq j$, $i, j = 1, 2, \dots, m$. When a new model named N_{m+1} is inserted, collision detection will be performed with all subsets in the set named N to determine whether $N_i \cap N_{m+1} = \emptyset$, $i = 1, 2, \dots, m$. We utilized the technique of oriented bounding boxes (OBBs) for this collision detection. Because the models built in this paper are all rigid bodies, using OBBs can greatly reduce the number of bounding boxes participating in the intersection test. Moreover, the precision and the time performance of OBB are better than axis aligned bounding boxes or bounding spheres.

When a model bounding box needs to be drawn in real time, the center point, the length and the width of a model are obtained through the stored information, and then a RealityBox based on this model is generated. A VirtualBox of the model is generated according to the constraints of the building codes. For two adjacent buildings, we will judge whether they intersect or not. For example, suppose two bounding rectangles are to be tested for intersection. The adjacent and orthogonal sides of the first rectangle are selected as the projection axes, which are the S axis and the T axis. As shown in Fig. 4, the projection interval of the first rectangle is $[s1, s2]$ on the S axis, and $[t1, t2]$ on the T axis. Similarly, the projection interval of the second rectangle is $[s3, s4]$ on the S axis and $[t3, t4]$ on the T axis. If either $[s1, s2]$ and $[s3, s4]$ are disjointing, or $[t1, t2]$ and $[t3, t4]$ are disjointing, then the rectangles are disjointing. If both of these pairs of projection intervals do not intersect, the intersection test result of these two rectangular bounding boxes is disjointing. If these projection intervals intersect, we will

choose the adjacent and orthogonal sides of the second rectangle as the projection axes, and repeat these intersection tests. If we find an axis along which the projections do not intersect, then the rectangles are disjointing, and if not, then the rectangles intersect. When the user operates this system, it will display the information of a RealityBox and a VirtualBox in the form of a red dashed rectangle in real time. When the user drags the model to the chosen position and releases the mouse, the system will generate the symbols of different models according to the placement position in the layout. A model that has been placed in the layout will also generate a RealityBox and a VirtualBox according to the building codes.

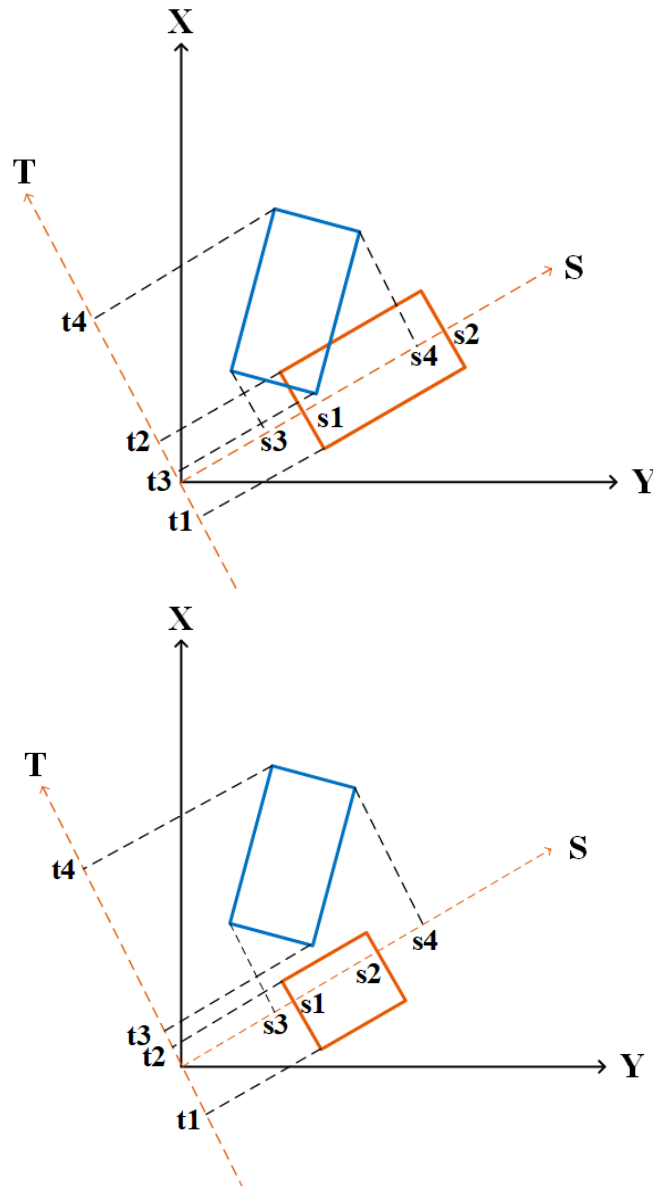


Fig. 4. Schematic diagram of improved directional bounding box intersection detection

5. Modeling the three-dimensional urban scenes

5.1 Algorithm from two-dimensional layouts to three-dimensional urban scenes

The two-dimensional layout of different city models is the basis to generate a three-dimensional virtual city scene. However, because dimensions and coordinate systems of two-dimensional layouts and three-dimensional virtual city scenes are different, the model symbol in the two-dimensional layout cannot be directly mapped to the three-dimensional scene through converting two-dimensional vector coordinates to three-dimensional vector coordinates.

Firstly, some model symbols are designed to represent different buildings, roads, and trees, etc.. According to the front view of a three-dimensional model, the model symbols are set, and the labeling information used to associate symbols with 3D models is established. These symbols are abstract representations of three-dimensional models. These symbols should have the function of expressing basic characteristics of three-dimensional models. For example, when users see a certain symbol, they can understand the model corresponding to this symbol. As shown in **Fig. 5**:

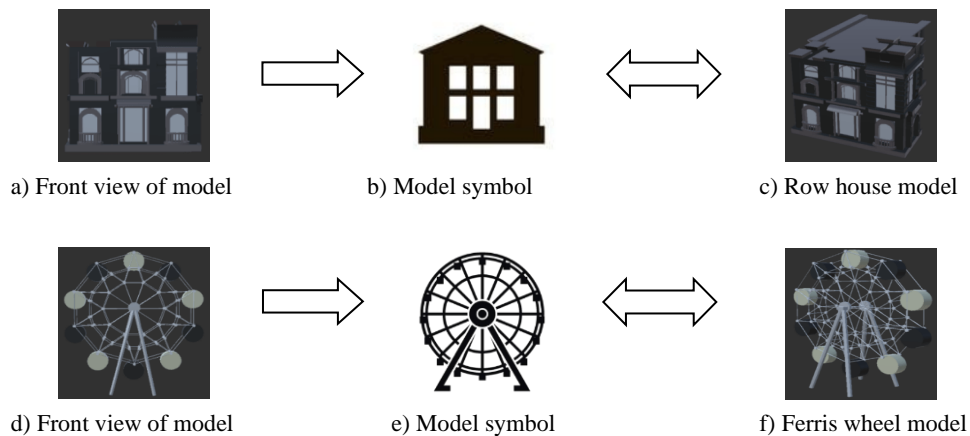


Fig. 5. Model symbols are associated with 3D models

In this paper, an algorithm is proposed to convert two-dimensional coordinates to three-dimensional coordinates based on anchor points. And the three-dimensional coordinates in the three-dimensional virtual city scene corresponding to model symbols are obtained by using the following algorithm:

Step 1: Set up the anchor points which are top-left, top-right, bottom-left, and bottom-right of a layout, and store their coordinates as $(X_{1-4}, Y_{1-4}, Z_{1-4})$.

Step 2: The coordinates $(X'_{1-4}, Y'_{1-4}, 1)$ which are top-left, top-right, bottom-left, and bottom-right of the screen coordinates system are now known, and then calculate S_{x11} , S_{y22} , S_{z33} , T_{x31} and T_{y32} which are used to convert coordinates from the world coordinates system to the screen coordinates system. The formula is

$$(X, Y, Z) \times \left(\begin{bmatrix} S_{x11} & 0 & 0 \\ 0 & S_{y22} & 0 \\ 0 & 0 & S_{z33} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ T_{x31} & T_{y32} & 0 \end{bmatrix} \right) = (X', Y', 1) \quad (8)$$

where S_{x11} , S_{y22} , S_{z33} , T_{x31} and T_{y32} are unknowns which need to be calculated. S_{x11} , S_{y22} and S_{z33} are the scaling of X, Y and Z respectively. T_{x31} and T_{y32} are the translation of X and Y respectively.

Step 3: The user clicks a symbol button of a certain model in the interface and drags it to the layout whose coordinate is (X_5, Y_5) .

Step 4: Convert the two-dimensional coordinate (X_5, Y_5) into the three-dimensional coordinate $(X_5, Y_5, 1)$, and multiply it with the inverse of the sum of two matrices to get the new coordinate (X'_5, Y'_5, Z'_5) that is in the world coordinates system. The formula is

$$(X, Y, 1) \times \left(\begin{bmatrix} S_{x11} & 0 & 0 \\ 0 & S_{y22} & 0 \\ 0 & 0 & S_{z33} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ T_{x31} & T_{y32} & 0 \end{bmatrix} \right)^{-1} = (X', Y', Z') \quad (9)$$

5.2 Interactive modeling of urban scenes based on Leap Motion

In this paper, Leap Motion hardware and Unity3D software are used to position and rotate the models to finish the interactive modeling through gesture definition and data processing. The detection space of Leap Motion is an inverted pyramid-shaped space from 2.5cm to 60cm above the device. When the user starts operating, they should put their hands in the detection space of Leap Motion before proceeding. Leap Motion will track the movement of the user's hand at 200 frames per second, but it has lower recognition of gestures with the palm facing down. Here three sets of user interaction gestures are designed based on the recognition characteristics of Leap Motion. As shown in **Fig. 6**: a) When the user opens five fingers, the system starts to read the user's hand movement data; b) When the user's fingers are pinching, the system will capture the 3D model. And the orientation conversion of the 3D model in the 3D scene is completed through the rotation of the hand; c) When the user's hand is in the shape of a fist, the system stops interacting.



Fig. 6. Definition of user's gestures

When the user begins operating, the data of hands will be stored in a frame array. In this frame array, key data such as fingertip coordinates (named TipPosition) and palm vector (named PalmVector) are stored. In this paper, the data will be read and added to coordinates and rotation angles of the three-dimensional model to enable users to manipulate 3D models through the gestures of his hands. The formula is

$$\begin{cases} (X, Y, Z) = (X + \Delta x, Y + \Delta y, Z + \Delta z) \\ R(X, Y, Z) = R(X, Y + \Delta ry, Z) \end{cases} \quad (10)$$

where (X, Y, Z) and $R(X, Y, Z)$ represent the coordinates and rotation angles of the model in the three-dimensional scene. Δx , Δy and Δz are respectively the increments of the 3D model on the X, Y, and Z axes after user's operation. Δry is the increment of the rotation angle of the three-dimensional model on the Y axis. In the 3D model, the change of the rotation angle in the Y axis corresponds to the change of the orientation in the virtual scene. In order to ensure the authenticity of the three-dimensional model, rotation angle increments are not set up in the X-axis and Z-axis to avoid tilting the building.

5.3 The management and reality enhancement of a 3D scene

Scene management in this paper includes scene changing, scene stitching and loading a scene dynamically. The former is to bind a pre-designed script of the scene changing to a button. In this script, various functions used in scene changing are designed. When a button is clicked, a certain function of the script will be triggered, and then the scene will change. Scene stitching is performed when pre-processing large-scale terrain. Multiple feature points for record cutting are stored in a linked list and numbered one by one. After a terrain block is generated, according to the order of terrain blocks in the linked list, stitching with other terrain blocks is completed through feature points. In order to solve these problems such as too many drawing scenes, too slow loading speed, and not running smoothly, we use the method of loading scene dynamically [18] to reduce the number of surfaces and details of non-important objects. Then rendering operations will be more efficient.

In order to make the three-dimensional scene more realistic, aesthetic knowledge is used to optimize the three-dimensional virtual city scene. Lighting that includes parallel lights, point sources, spotlights, regional lights and so on is indispensable in a three-dimensional virtual city scene. In Unity3D, lighting is real-time lighting by default, and different positions of objects in the lighting will produce different visual effects. Therefore, the RGB value and direction of the lighting should be changed. In this paper, we use soft lighting with RGB (244, 231, 199), which can bring better visual effects, as shown in Fig. 7.



Fig. 7. Illumination comparison

In this paper, the foreground, middle-ground and background are combined during the creation of terrain blocks. When the user begins roaming, the foreground is composed of plants and buildings on the roadside and the middle-ground is the High-rise Building in the distance and the background is the dense mountains and the blue sky. The superposition of these scenes will increase the level of the scene. Skybox technology is used to realize the sky in the distance, and six special textures are placed at the edge of the scene, so that the scene borders will perfectly blend with the sky to make it more realistic, as shown in Fig. 8.



Fig. 8. SkyBox renderings

6. System development and discussion

6.1 System development

In this paper, a system is developed to interactively simulate the virtual city scene by using Unity3D. The hardware includes Quad-core CPU 3.40GHz, 8G memory, Graphics NVIDIA GeForce GTX 1060, 3GB video memory. The system works well on operating systems such as Windows and OSX.

Taking Hangzhou Beishan Street as an example, pictures collected on the spot and city maps obtained from Baidu map should be processed. First, Photoshop is used to extract feature information such as roads, greening, mountains, the surface of the water, etc.. Second, topographic maps of the city obtained from a Baidu satellite map are divided into multiple topographic maps of 200 meters by 200 meters. Third, we use Unity3D to export it in *.terrain format. Finally, these files are stored in a database for management and invocation.

The model display interface is divided into three components, as shown in Fig. 9: the model symbol area on the left of the interface, the 3D model display platform on the middle of the interface, and the models' information area on the right side of the interface. This interface which includes a model schematic list and relevant information about the models is used to visually display 3D models and model information to the user. When a user clicks on a model symbol, a corresponding three-dimensional model will be drawn in the scene and the relevant information will be displayed in text in the information bar on the right side of the interface. In this system, when a user begins operating, the bounding box will be displayed in the form of a red dashed rectangle in real time. When the user drags the model to a chosen position and releases the mouse, the model will be generated according to the placement position in the layout. Models that had been put in the layout before will also generate bounding boxes according to building codes. Fig. 10 shows three-dimensional virtual city scenes generated according to different layouts. It can be seen that in this system, a model of an urban scene can be built, which not only conforms to the real scene but also strictly follows building codes.

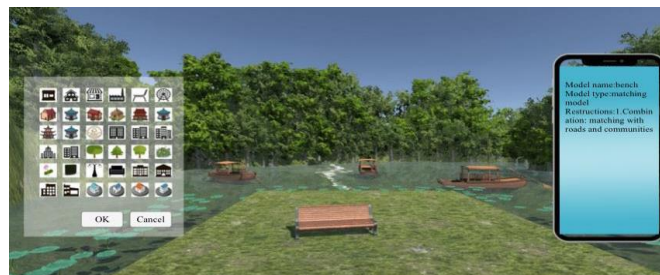


Fig. 9. Model displayed interface



a) Hangzhou Shangri-La Hotel



b) Hangzhou Yuewang Temple



c) Hangzhou Youth Palace

Fig. 10. Simulated results

6.2 Discussion

In this paper, a method to interactively create virtual city scenes based on building codes is proposed, which provides information and decision-making functions for urban planning and management. Compared with previous work, users can simulate urban virtual scenes that strictly adhere to building codes in a virtual scene of the city environment through somatosensory interaction to complete the design and construction of three-dimensional scenes. However, our method has several limitations: 1) In order to solve the problem that geometric objects may intersect and collide in a virtual city scene, an improved technique of the directional bounding box is proposed in this paper. However, due to the characteristics of the directional bounding box, its intersection detection will be more complicated. 2) A strategy to convert the two-dimensional layouts to a three-dimensional urban scene is proposed, but this strategy requires manually labeling symbols of three-dimensional models. If we use the technology of combining context information instead, it can be more efficient. 3) We adopt an interactive method based on Leap Motion. This method requires users to suspend their hands in the detection space of Leap Motion for a long time. It will cause the user physical energy consumption and arm pain. 4) Unfortunately, the system does not provide an interface to change building codes, so it cannot be used in other countries. If building codes have changed, it can be realized by changing the code.

7. Conclusion

In this paper, we propose a method to interactively simulate a virtual city scene based on building codes and implement an interactive virtual city scene system based on this method which will automatically prevent placement that produces collisions of the Virtual Boxes. In the system, the user can click a model symbol to complete the drawing of a 3D model, and dynamically change the position and orientation of the 3D model through gestures. The method we proposed will be helpful for decision-making in urban planning and management. In the future, a collision detection technology based on hybrid hierarchical bounding box can be considered instead of the collision detection technique of the directional bounding box. Besides, designing a gesture with a single finger or a tool can be taken into account to provide users a more comfortable interaction method.

Acknowledgment

This work was supported by the National Natural Science Foundations of China (61571400) and Zhejiang public welfare technology research plan / social development project (LGF21F020015). The authors are grateful to Professor Nelson Max from University of California Davis, Wen-xiu He from Zhi-jiang College of Zhejiang University of Technology, and the anonymous reviewers whose comments helped to improve this paper.

References

- [1] W. J. Greenleaf, "Virtual reality applications in medicine," in *Proc. of WESCON '95*, p. 691, 1995. [Article \(CrossRef Link\)](#)
- [2] H. Li, "Application and Expectation of Virtual Reality Technology in Medical Field," *China Digital Medicine*, vol. 4, no. 5, pp. 50-55, 2009. [Article \(CrossRef Link\)](#)
- [3] B. Sun, "Review of the Application of Virtual Reality Technology in American Army," *Computer Simulation*, vol. 35, no. 1, pp. 1-7, 2018. [Article \(CrossRef Link\)](#)
- [4] M. Virvou and G. Katsionis, "On the usability and likeability of virtual reality games for education: The case of VR-ENGAGE," *Computers & Education*, vol. 50, no. 1, pp. 154-178, 2008. [Article \(CrossRef Link\)](#)
- [5] T. Bai, "The Virtual Reality Technology and Its Application in the Urban Planning and Management," *Science & Technology Information*, vol. 7, pp. 54-55, 2010. [Article \(CrossRef Link\)](#)
- [6] Y. Sun, X. Zhao, Y. Wang, F. Li, and X. Li, "Study on the visual evaluation preference of rural landscape based on VR panorama," *Journal of Beijing Forestry University*, no. 12, pp. 104-112, 2016. [Article \(CrossRef Link\)](#)
- [7] Y. Takaichi, K. Yudai, and O. Yoshinori, "Three-Dimensional Object Modeling in a Three-Dimensional Urban Map," in *Proc. of M3 '98 International Workshop on Urban MultiMedia/3DMapping*, 1998. [Article \(CrossRef Link\)](#)
- [8] M. Sun, A. Ma, and J. Chen, "Review on Three-Dimensional City Model Research," *Journal of Remote Sensing*, vol. 6, no. 2, pp. 155-160, 2002. [Article \(CrossRef Link\)](#)
- [9] Y. Gan, J. Weng, Z. Sui, and X. Ji, "Fast constructing large scale city 3D building scene based on 2D building outline," in *Proc. of 2010 18th International Conference on Geoinformatics*, pp. 1-4, 2010. [Article \(CrossRef Link\)](#)
- [10] J. Yuan, Y. Li, H. Pan, Z. Cui, and Y. Liu, "3D Traffic Scenes Construction and Simulation based on Scene Stages," in *Proc. of 2018 Chinese Automation Congress*, pp. 1334-1339, 2018. [Article \(CrossRef Link\)](#)

- [11] S. N. Sinha, D. Steedly, R. Szeliski, M. Agrawla, and M. Pollefeys, "Interactive 3D Architectural Modeling from Unordered Photo Collections," *ACM Transactions on Graphics*, vol. 27, no. 5, 2008. [Article \(CrossRef Link\)](#)
- [12] J. Liang, Y. Li, X. Dai, F. Yang, and Q. Shi, "3D Reconstruction of Digital Campus Based on Oblique Imaged of Unmanned Aerial Vehicle," *Geomatics & Spatial Information Technology*, vol. 41, no. 8, pp. 139-145, 2018. [Article \(CrossRef Link\)](#)
- [13] S. Zhou, C. Ma, and G. Chen, "An Inverse Procedural Modeling and Hybrid Rendering Approach for Large-Scale Urban Scenes Visualization," *Journal of Computer-Aided Design & Computer Graphics*, vol. 1, pp. 88-97, 2015. [Article \(CrossRef Link\)](#)
- [14] Y. Parish and P. Müller, "Procedural Modeling of Cities," in *Proc. of ACM SIGGRAPH*, pp. 301-308, 2001. [Article \(CrossRef Link\)](#)
- [15] C. Zhou, "Study on the Digital City 3D Scene Modeling and Visualization Technique," *China University of Mining and Technology*, 2014. [Article \(CrossRef Link\)](#)
- [16] P. Ren, Z. Wang, Y. Fan, M. Zhou, and G. Du, "A Rapid Modeling Method for 3D Architectural Scene," in *Proc. of 2016 International Conference on Cyberworlds*, pp. 9-16, 2016. [Article \(CrossRef Link\)](#)
- [17] The PRC Ministry of Construction, Code of Urban Residential Areas Planning & Design (GB50180-93), Beijing, China: China Architecture & Building Press, 2002. [Article \(CrossRef Link\)](#)
- [18] H. Li, "The analysis of 3D scene data parallel processing strategy and model simplification," *Hunan University of Science and Technology*, 2017. [Article \(CrossRef Link\)](#)



Wei-Long Ding received the B.S. degree and the M.S. degree in mechanical design engineering from “AnHui University of Science & Technology” respectively in 1998 and in 2001, and the Ph.D. degree in pattern recognition and artificial intelligence from “University of Science & Technology of China” in 2004. He was a visiting scholar of University of California Davis in 2013. This author became a Member (M) of IEEE in 2009 and became a member of China Society of Image and Graphics in 2014, and a professor of college of computer science and technology, Zhejiang University of Technology, Hangzhou, China. His research interest includes virtual reality, computer graphics, and intelligent system.



Xiao-jie Zhu received the B.S. degree in computer science and technology from Zhejiang A&F University, Hangzhou, China, in 2019. She is currently pursuing the M.S. degree in computer technology at Zhejiang University of Technology, College of Computer Science & Technology, Hangzhou, China. Her research interest includes virtual reality, computer graphics.



Bin Xu received the M.S. degree in computer technology from Zhejiang University of Technology, Hangzhou, China, in 2018. His research interest includes virtual reality, computer graphics.



Yan Xu received the M.S. degree in software engineering from Zhejiang University of Technology, Hangzhou, China, in 2018. His research interest includes virtual reality, computer graphics.



Kai Chen received the B.S. degree in electrical and information engineering from Zhejiang University of Technology, Hangzhou, China, in 2015, and the M.S. degree in software engineering from Zhejiang University of Technology, Hangzhou, China, in 2018. His research interest includes intelligent improvement of virtual characters, behavior simulation.



Zang-xin Wan received the B.S. degree in software engineering from Zhejiang University of Technology, Hangzhou, China, in 2017. He is currently pursuing the M.S. degree in software engineering at Zhejiang University of Technology, College of Computer Science & Technology, Hangzhou, China. His research interest includes virtual reality, computer graphics.