

EziCity: A Rapid Landscape Design Framework of Virtual 3D City

Jianliang Lan¹ and Youdong Ding²

Abstract. A framework of virtual 3D city aiming at rapid, interactive and intuitive landscape prototype design is presented in this paper. The conception or assembling phase of a virtual city is usually a laborious process, because the layout and manipulation of 3D scene objects often does not yield the expected results due to the lack of auxiliary 3D information with the condition that only 2D user interfaces are available. With the help of spatial layout information and elementary spatial constraints, rapid prototype construction of virtual city landscape is possible.

Keywords: 3D design · Virtual city · Scene by sketch · Spatial constraints

1 Introduction

With the rapid development of CAD (Computer Aided Design), VR (Virtual Reality), AR (Augmented Reality), 3D user interfaces have become a new form of information exhibition and human-computer interaction, widely used in tourism, construction, real estate and many other industries. While as the prerequisites of these technologies, 3D Content design and assembling in CAD or VR systems are often laborious processes, not only requiring experienced technicians but also consuming lots of time. A fundamental issue is that users are inconvenient to manipulator scene objects in a 3D space, like moving objects to specified position or rotating objects to special orientation, without the help of 3D spatial position and orientation awareness devices.

In this paper, a virtual city landscape design framework named *EziCity* is introduced. *EziCity* aims at providing people a rapid prototype of virtual 3D city land-

¹ Jianliang Lan (*)

School of Computer Engineering & Science, Shanghai University, Shanghai, China
e-mail: lanjianliang@126.com

² Youdong Ding

School of Film & Television Art & Technology, Shanghai University, Shanghai, China

scape design or modelling. It involves many common aspects like regular CAD or VR systems, such as models management, illumination, terrain & sky module, weather & ambient module, control & roaming module, selection, collision detection etc., making it a self-contained system for city landscape design or city planning.

EziCity exploits the scene descriptions provided by a scene sketch which would be translated into a well structured XML file to simplify the design progress. The scene sketch contains the basic 2D scene layout, initial roaming path and other ambient information, thus could be used to help positioning scene objects, optimizing roaming paths and setting up other aspects of scene environment, which enable users to concentrate their time and efforts to design and assemble the desired virtual scene as soon as possible.

To alleviate the difficulty of manipulating objects in 3D space with 2D user interfaces only, elementary spatial constraints, like the DOF (degrees of freedom) of object are also taken into consideration during the design phase. There has to be six DOFs for a rigid object in a 3D space, but it's obvious that we can confine them down to less DOFs. For example, buildings and trees must be placed exactly on the ground, allowed with yawing around the vertical axis, which means they have three DOFs, so that the manipulation to these objects could be confined to a lower dimension space.

2 Related Work

Various forms of virtual city modeling or planning have been researched for years, and still remaining an active research area [6]. Researches usually focus on some specific topic like modeling city roads, street blocks, residential blocks, layout maps etc., because it is too difficult to modeling all the aspects of a city. Procedural modeling for the virtual city or digital urban is the hotspot recent years [7][12], and lots of approaches and methodologies have been emerged. Nobuko Kato etc. [10] propose an artificial life L-system approach to automatic modeling the block shapes, road shapes and graphical topology of a virtual city. Nobuko Kato etc. [11] also consider the city to be composed of several city blocks, and each block has its own feature values, such as average building height, which are determined using a genetic algorithm. Chen and Liu [5] have analyzed the layout, architectural distribution and traffic system of a city, and propose a fractal city model to simulate process of city evolvement through fractal mechanism. Two sub-processes involve in the whole process, iterative process and fractal process, which are recursively executed until forming a rational city layout. Merrell etc. [1] have done some excellent work in procedural modelling of residential building layouts. They are motivated by a methodology for building layout design commonly encountered in real world architectural practice, using a Bayesian network trained on real world data to synthesize an architectural program, with a set of high level requirements. Their approach could procedurally generate a set of opti-

mized floor plans and a set of decorated 3D model, proved to be useful in modeling the virtual city.

Many researchers engage in exploiting the benefits of virtual 3D city, to serve other industries like city tourism, real estate, building architecture and traffic simulation and so on. Virtual 3D scene provides people an intuitive city landscape, while it is also easy to make people lost in the complex scene. Huang etc. [9] present a dynamic labelling method in 3D scene base on the full study of 2D cartography principles and existing 3D labelling methods. Tahrani and Moreau [14] study the solar effects in real and virtual world and therefore daylight ambience by walking in an immersive and interactive virtual city scene. Reika [13] carries out a study on users operating the virtual reality system to investigate their responses of “like” or “dislike” towards the real time adjustments of design effect, pave the way for future research on monitoring psychological impacts on users of virtual reality system during design process.

There are many ways and toolsets to build a virtual city, while the dilemma people faced with is about flexibility or functionality. As other large scaled 3D scene system, two most common methodologies for building virtual 3D city are network-based architecture or native-based architecture. Network-based system provides the ability of accessing one copy of scene data from different clients, asynchronously maybe, saving a lot of storage space and efforts for dealing the distributed data, with the prerequisite that the network latency is tolerable. Fan etc. [8] leverage VRML in virtual city construction on the platform of Web 3D, while the interactions between users and the VRML interfaces may also be confined due to the VRML’s limitation. Most of the mature commercial modelling or designing software, for instance, Maya, 3D Studio Max etc. are native-based programs, providing maximum functionality but less flexibility such as remote collaborative working. Open source libraries for graphic rendering or dynamic mechanical analysis are available in the open source communities. OSG (Open Scene Graph) [4] is a powerful library for building and rendering the scene graph, which could be considered as a primary toolset for virtual 3D city implementation.

3 EziCity

Ezicity is a virtual 3D city landscape design framework aiming at rapid prototype generation. It composes three sub-systems, whose basic functions and details are described as follows.

- Sketch generation and translation system
- Scene generation and modification system
- Roaming and capturing system

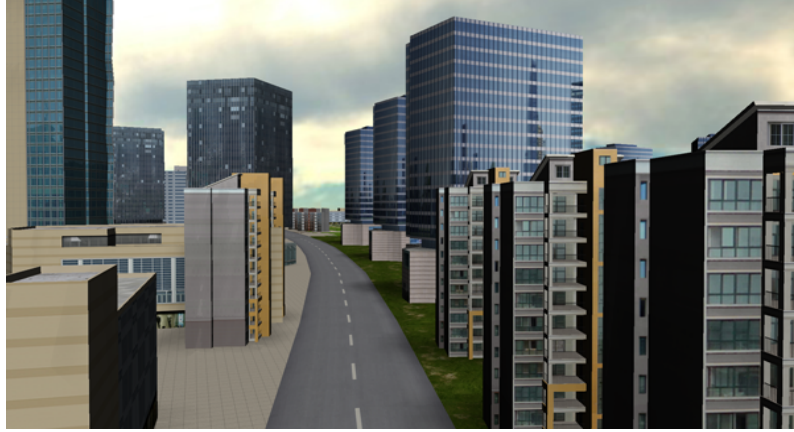


Fig. 3 A virtual 3D city landscape generated by *EziCity*

At the landscape designing phase, designer could depict the scene to a planar sketch with necessary information in the form of labelling, providing *Ezicity* the basic scene layout and spatial constraints. The system could translate the sketch to a formalized and structured XML document, facilitating other sub-system to generate or modify the scene.

From the sketch, all necessary scene information is collected. The information includes but is not limited to illumination, weather, terrain, sky, objects' location & orientation, object's spatial constraints, roaming path and so on. *Ezicity* uses this information to construct the whole scene, and after that, it also enables designers to modify the scene.

Roaming in the scene is a very import function for a landscape design framework. *Ezicity* extracts roaming paths from the scene sketch, and generates animation paths of the viewing camera. Roaming system could also capture the landscape to an image or a video during the roaming period.

3.1 Scene Sketch

Scene sketch is a simplification and abstraction of a virtual 3D city scene, depicting the scene entities in a 2D space. Every entity has its own property or constraint which is labeled in the sketch. There are mainly three types of scene entities as follows.

- Scene ambient, such as illumination, weather, terrain, sky etc.
- Scene objects, such as buildings, plants, pedestrians etc.
- Roaming paths

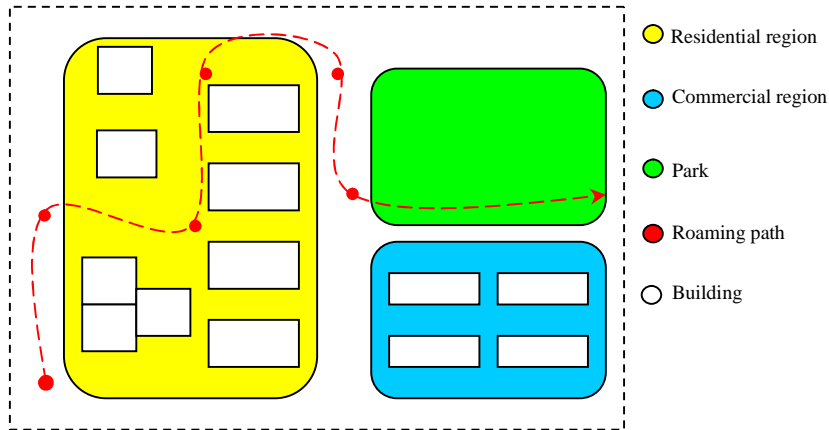


Fig. 3.1 Scene sketch example of some virtual city regions and a roaming path. The different shapes and colors are only to better demonstrate the various city regions and entities. Properties of entities are not showed in is example.

Scene objects are the most important entities of a scene. Designer could easily add an object such as a building or a pedestrian with mouse or writing pen at some position of the sketch, and then assign it with some properties like building type or pedestrian type.

In order to make it easier for designer to interact with scene objects with the condition that only 2D user interfaces are available, some elementary spatial constraints could be set to the scene object's properties. For instance, buildings, plants, pedestrians must be located right above the terrain, confined to three DOFs, which are two directions of translation on the terrain plane and one rotation around the vertical axis. These constraints make the object's manipulation more intuitive and reasonable [2][3].

3.2 Scene Generation

EziCity manages the whole scene graph in the form of directed acyclic graph [4]. The scene root node has four significant child nodes: terrains, models, lights and the sky-box nodes. Data flows or events are passed and handled by node visitors and call-back functions. All the leaf nodes in the scene graph represent geometry to be rendered, and non-leaf nodes establish hierarchy and scene manipulations.

EziCity takes the remote collaborative working into account, so it is based on network architecture. All scene materials (mainly models) are stored in a remote data server, and *EziCity* acts as the role of client design frontend. There are many benefits to do this way, such as isolating the data and operations, saving storage space, allowing multiple users to design different scenes asynchronously.

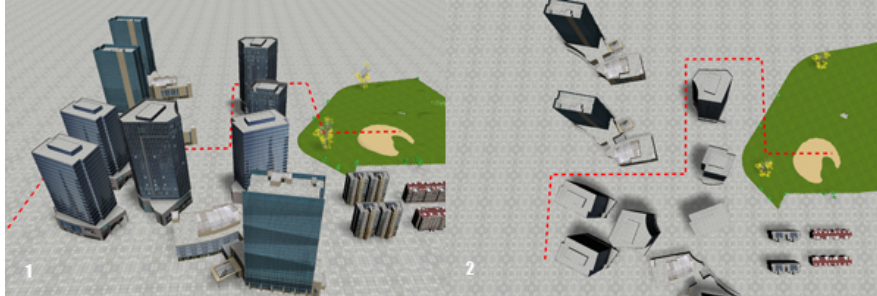


Fig. 3.2 Scene generate procedurally by the previous sketch example. The red stippled line indicates the raw roaming path, and could be hid after design phase.

Scene generation procedure could be briefly described in pseudo instructions as follows.

1. Load the translated sketch XML file, and parse it to get the scene descriptions.
2. Set up the ambient entities like illumination, weather, terrain, sky.
3. Set up the scene root node, and do the steps from 3.1 to 3.3.
 - 3.1. If there is no other object to add to the scene, go to step 4.
 - 3.2. Get the object's type or name, search it from remote data server, if hit, download it and add it to the scene. If there are many candidates, offer the users a selection list.
 - 3.3. Position the downloaded model by its spatial properties. Most of the objects should be put onto the terrain plane except those floating objects like birds or balloons. After that, go to step 3.1.
4. Enter scene modification mode.

3.3 Scene Roaming

Scene roaming is a necessary function for a modern landscape design system, improving the realistic sense of immersion. *EziCity* enables designer to capture the landscape during roaming process to a quality lossless image or a high quality video, which is rather handy for designer's to generate further purpose materials like advertisements.

A raw roaming path is composed of a series of control points configured in the scene sketch, assuming that there are n control points. Each control point P_k ($0 \leq k < n$) is defined as a vector $(X_k, Y_k, Z_k, T_k)^t$, where $(X_k, Y_k, Z_k)^t$ is the point's Cartesian coordinates, and T_k is the timestamp of P_k indicating how soon the camera will arrive at this point from the beginning of the path.

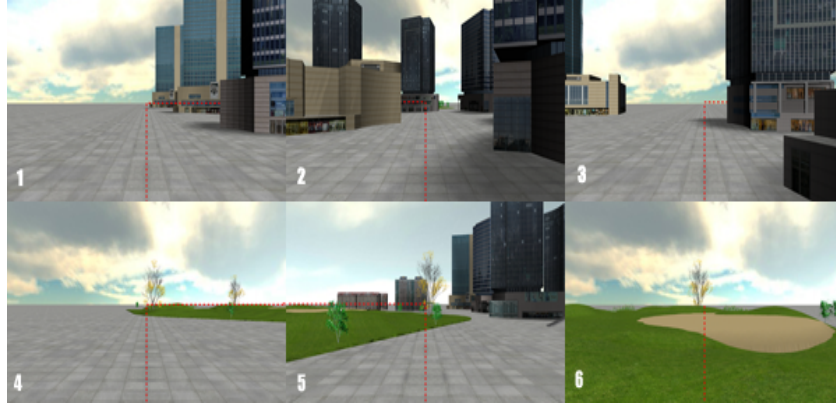


Fig. 3.3 Different views when roaming in the virtual 3D city. The red stippled line indicates the raw roaming path with which the current camera follows.

In the scene sketch phase, control points are set where the roaming path turns corners. To avoid the scenario that camera turns too suddenly when arriving at a control point, we have to optimize the raw roaming path. Spherical interpolation is a good option to smooth the camera's motion. But if the interpolations are carried out between two raw control points, the optimized path may deviate too much from the raw path, especially when these two raw control points are far from each other. In order to address this issue, auxiliary control points are added to the raw path. For every P_k ($0 < k < n$), one auxiliary control point is defined as P_{k-} , and $P_{k-} = P_{k-1} + (1-d) \cdot (P_k - P_{k-1})$, where d ($0 < d < 1$) is a curvature factor about 0.1 is smooth enough for usual cases. After that, we get the new series of control points $P_0, P_{1-}, P_1, P_{2-}, P_2, \dots, P_{n-1-}, P_{n-1}$. Spherical interpolations are then carried out between P_{k-} and P_k , where $0 < k < n$, producing the smooth path for camera's motion.

4 Conclusions

Ezicity, a framework of rapid landscape prototype design of virtual 3D city is introduced in this paper. This framework is aiming at an intuitive and interactive system for modelling the virtual 3D city landscape for real industries purpose. The scene could be generated procedurally from a sketch with necessary properties labels. To improve user's the realistic sense of immersion, optimized roaming paths could be configured. User can also capture the landscape during the roaming process for further purposes.

Acknowledgment

This research is mainly supported by *Shanghai Educational Committee Leading Academic Discipline Project (No.J50103)* and *Shanghai Science & Technology Key Research Project (11511503300)*.

References

1. Merrell, P., Schkufza, E., Koltun, V. (2010). Computer-generated residential building layouts. *ACM Transactions on Graphics*, Vol29.6, No.181, NY, USA. doi: 10.1145/1882261.1866203
2. Smith, G., Salzman, T., Stuerzlinger, W. (2001). 3D scene manipulation with 2D devices and constraints. *Graphics Interface*. Morgan Kaufman Publishers, 135-142.
3. Zanden, B., Myers, B., Giuse, D., Szekely (1994). Integrating pointer variables into one-way constraint models. *ACM Transactions on Computer-Human Interaction*, 161-213.
4. Burns, D., Osfield, R. (2004). Tutorial: Open scene graph A: introduction tutorial: Open scene graph B: examples and applications. *IEEE International Conference on Virtual Reality*. 265. doi: 10.1109/VR.2004.1310100
5. Chen, J., Liu, H. (2007). Modelling virtual city based on fractal. *ICPCA, 2nd International Conference on Pervasive Computing and Applications*, 78-83. doi: 10.1109/ICPCA.2007.4365416
6. Deng J., Zheng, X., Lv, L. (2010). Virtual urban — A scientific tool and platform for urban research. *International Conference on Audio Language and Image Processing (ICALIP)*,1268-1273. doi: 10.1109/ICALIP.2010.5685106
7. Engel,J., Pasewaldt, S., Trapp, M., Dollner, J. (2012). An immersive visualization system for virtual 3D city models. *International Conference on Geoinformatics (GEOINFORMATICS)*, 1-7. doi: 10.1109/IV.2009.101
8. Fan, W., Shi, Y., Peng, Z., Liu, S. (2009). Research on application of VRML in virtual city Construction. *International Joint Conference on Artificial Intelligence*, 598-601. doi: 10.1109/IJCAI.2009.82
9. Huang, Y., Kong, D., Zhang, Y. (2012). Study on dynamic labelling of building in 3D virtual city. *IEEE Symposium on Electrical & Electronics Engineering (EESYM)*, 156-162. doi: 10.1109/EESym.2012.6258613
10. Kato, N., Okuno, T., Okano, A., Kanoh, H., Nishihara, S. (1998). An ALife approach to modeling virtual cities. *IEEE International Conference on Systems, Man, and Cybernetics*, vol.2, 1168-1173. doi: 10.1109/ICSMC.1998.727864
11. Kato,N., Okano,A., Okuno,T.(1999). Modeling virtual cities using genetic algorithms. *Proceedings of the IEEE Midnight-Sun Workshop on Soft Computing Methods in Industrial Applications*, 134-139. doi: 10.1109/SMCIA.1999.782722
12. Poullis, C., You, S. (2009). Automatic Creation of Massive Virtual Cities. *IEEE Virtual Reality Conference*. 199-202. doi: 10.1109/VR.2009.4811023
13. Reika Sato-Wang. (2006). Cognitive psychological for urban plan & architecture using the virtual reality system. *IEEE International Conference on Systems, Man and Cybernetics (SMC)*, vol.3, 2045-2050. doi: 10.1109/ICSMC.2006.385161
14. Tahrani,S., Moreau, G.(2007). Analyzing urban daylighting ambiences by walking in a virtual city. *IEEE International Conference on Virtual Reality*. 227-230. doi: 10.1109/VR.2007.352487