RESEARCH ON VISUALIZATION TECHNOLOGY OF CYBERCITY 3D SCENE

Zhou Yang, Xu Qing, Lan Chaozheng , TanBing
(Institute of Surveying and Mapping, ZhengZhou 450052,China)
zhouyang@chxy.com

PS WG V/6
Visualization and Animation

Key Words:, GIS, Analysis, Modelling, Visualization, Data Structures, DEM/DTM, Texture, System

Abstract:

3D visualization is a Key Technology in CyberCity. It has been applied in many fields. In this paper, we discussed and studied several key techniques of 3D visualization in CyberCity. These key techniques contain Oriented Object 3D Data Model, Advanced Modeling Techniques which contain Particle System and Billboard, 3D Scene Accelerate Rendering Techniques, Information Inquire techniques in 3D scene and so on. The 3D Scene Accelerate Rendering Techniques contain Object Culling, Back Face Culling and LOD (level of detail) technique. Based on these techniques, we designed and implemented an advanced 3D Rendering Engine. At the end of paper, we give some results created by the 3D Rendering Engine. The results show that the 3D engine can render highly reality 3D scene and user can obtain highly immersion by rambling in 3D scene and inquiring information based on database.

1 Introduction

The conception of CyberCity is derived from digital earth and is the important part of digital earth. As a new technique, CyberCity can be regarded as expansion and development from CIS. By the studying Visualization Technology of CyberCity 3D scene, which apply the 3D Visualization technique in construction of CyberCity, we can change the description of city scene's actuality and programming in terms of 2D map and 3D solid model into in terms of 3D space representation of computer.

We think there are mainly several key techniques in CyberCity 3D scene Visualization Technology. They are as follows:

① Data structures: Data structures are a base of representation for a solid object. A satisfactory data structure described a solid model may even make it possible to generate instructions automatically for computer-controlled machine tools to create that object. So whether the data structure described a solid model is advanced determine the performance of a 3D engine.

② Large dataset Accelerate Rendering technique: Highly detailed geometric models are necessary to satisfy a growing expectation for realism in Cybercity 3D scene rendering. But the computer graphic hardware' developing is limited and a large dataset 3D scene makes real time rendering difficult. To try to attain real time frame rate, we must take some strategy to accelerate the 3D scene rendering. Currently, the accelerate Rendering technique include manage disk paging of geometry and texture, level of detail (LOD) selection for texture blocks, LOD for triangle geometry, culling to the view frustum, and triangle stripping and so on.

③ Advanced modeling: Advanced modeling can describe many natural phenomena efficiently. For example, fog, smoke, burst etc are efficiently represented by advanced modeling and vegetation can be rendering by Billboard.

④ Information Inquire and analysis in 3D scene. Information inquire and analysis based database in 3D scene is one of key technique in 3D GIS and it is the character differed from the general software of 3D rendering. User can gain more interest information by the function of inquire and analysis.

In this paper, we discus and research these techniques Based on these researches, we design and implement an advanced 3D Rendering Engine PowerCity3D. The PowerCity3D is applied in many fields and get some good results. At the end of this paper, we give some examples created by the 3D Rendering Engine and draw some conclusions.

2 Space Object Reconstruction

2.1 Oriented Object Data Structure

The space object' 3D Modeling is an important technique to Visualization in CyberCity 3D scene. In order to reconstruct the object quickly and reliably, we must design an advanced data model. Because of the variety of space object, we use Oriented Object's idea to design space object' data structure.

In order to adapt to city 3D scene's rendering and information inquire and analysis, we use the follows space object data model to describe the city object:

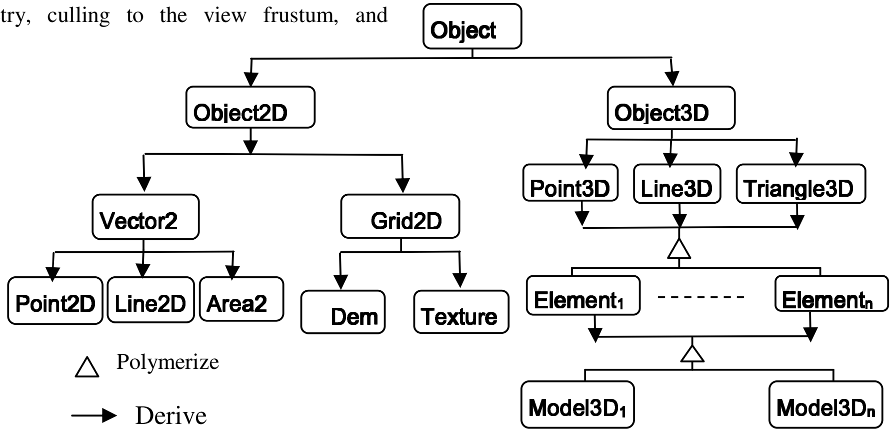


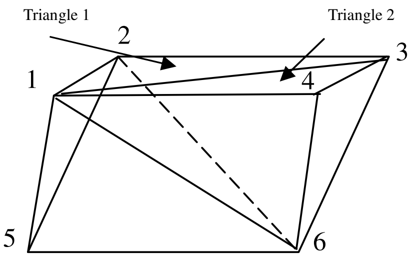
Fig.1 Oriented Object's data structure

As shown in Figure 1, we divide the object into Object2D including Vector2D, Grid2D and Object3D including Point3D Line3D, Triangle3D. The description of Object2D is same as tradition 2D GIS, but the description of Object3D is different. Because any space object can be fit by a lot of triangles, so we use the Triangle3D take the place of the Area3D in Object3D.

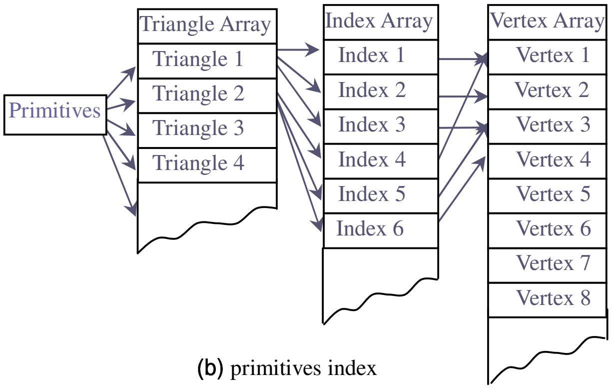
2.2 Descriptions of Object Models

Descriptions of space object models always are focus in 3D GIS. Object model can be described in several ways: Primitive instancing, Spatial enumeration, Cell decomposition, Sweep method, Boundary representation (B-Rep), Constructive Solid Geometry (CSG)[Zheng 1999]. The two most important type, Boundary representation and Constructive Solid Geometry, are feasible [Chang 2000].

B-Rep describes an object in terms of its surface boundaries, vertices, edges, and faces. Some B-Rep are restricted to planar, polygonal boundaries, and may even require faces to be convex polygons or triangles.[Foley 2002] As shown in Figure 2, we define a roof by B-Rep.



(a) primitives



(b) primitives index

Fig.2 Description of primitives

In CSG, simple primitives are combined by means of regularized Boolean set operators that are included directly in the representation. An object is stored as a tree with operators at the internal nodes and simple primitives at the leaves (Figure.3). Some nodes represent Boolean operators, whereas others perform translation, rotation, and scaling. So we define nine primitives: Cylinder, Extrusion, Helix, Polygon, PolygonMesh, Revolve, Skin, Sphere and Refer.

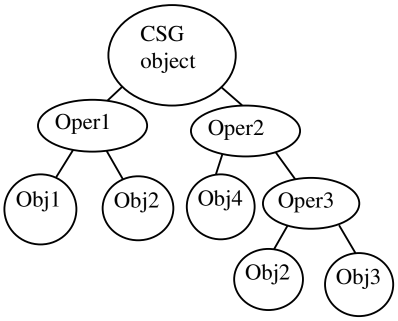
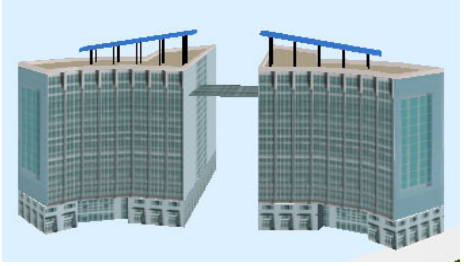
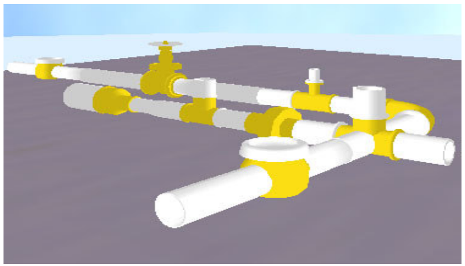


Fig.3 object defined by CSG and its tree

In this paper, we combine the characteristic of B-Rep and CSG by using B-Rep to describe the primitives and using CSG to describe the object consisted of some primitives. Figure 4 shows some of visualization products defined by CSG and B-Rep.



(a) Build



(b) Tube

Fig.4 visualization products defined by CSG and B-Rep

3. Large Scene Accelerate Rendering

Modern graphics workstations allow the display of thousands of shaded or textured polygons at interactive rates. However, many applications contain graphical models with geometric complexity still greatly exceeding the capabilities of typical graphics hardware.[Lindstrom 1996] This problem is particularly prevalent in applications rendering large city 3d scene.

Because there are large datasets that should be rendered in large city 3d scene, so we need apply some algorithms to reduce datum which need to be rendered and accelerate 3D scene rendering and improve the quality of visualization.

3.1The Accelerate Rendering strategy

A complete system to display views of large datasets at high frame rates consists of components to manage disk paging of geometry and texture, level of detail (LOD) selection for texture blocks, LOD for triangle geometry, culling to the view frustum, and triangle stripping.

In this paper, we applied **Object Culling, Back Face Culling ,** and **LOD** （level of detail）technique in rendering of large area 3D scene (Figure.5). [Woodl 1999][Rossignac 1993][Duchaineau 1997]

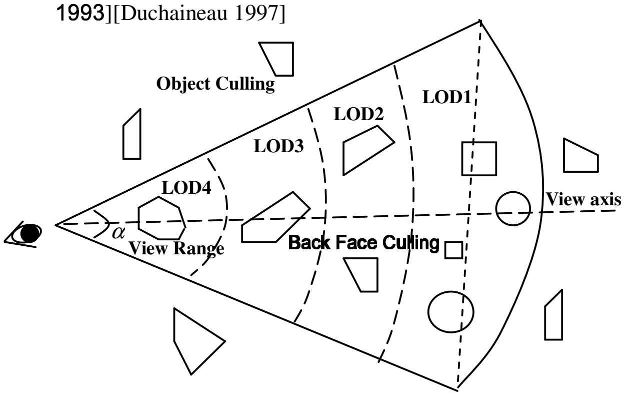


Fig.5 3D Scene Accelerate Rendering strategy

Using object culling, those sightless objects will be culling and only those visual objects can be rendering. For the object's visibility, we can use triangle to describe the view range (Figure.5) and judge whether the object is in triangle. The object's visibility can be defined as:

$$\prod_{i \neq j} \begin{vmatrix} x_i & y_i & 1 \\ x_j & y_j & 1 \\ x & y & 1 \end{vmatrix} = \begin{cases} v > 0 & inside \\ v < 0 & outside \end{cases} \quad (1)$$

$$(i = 1,2,3 \quad j = 1,2,3)$$

Where, (x, y) is the coordinate of object, (x_i, y_j) is triangle vertex coordinate.

For those visual objects, we can give it a weight coefficient according to its weightiness and its position away from viewpoint to decide its detail. So we use a function $D(O,d,m)$ to describe object detail.

$$D = \begin{cases} 0 & d > d_0 \\ D(O,d,m) & d \leq d_0 \end{cases} \quad (2)$$

Where, O is object id, d is distance which object away from viewpoint, m is object weight coefficient. D is object detail.

According to D, we can use different level of detail（LOD）to describe and render the object.

3.2 LOD Terrain Rendering

In Large 3D scene, we will spent a lot of resources on rendering terrain. The terrain data include digital elevation model (DEM) and texture. Currently, methods of LOD terrain rendering can be divided into two parts, GRID LOD and TIN LOD[Hamann 1990][Hoppe 1996][Hoppe 1998][Duchaineau 1997]. the GRID terrain data are simplified in bintree or quadtree.

In this paper, we divided the GRID terrain into some patches which its size must be power(2),for example ,32×32,64×64. we first judge whether the patch is visible, then use bintree to splitting the visible patch(Figure 6). When the terrain be rendering, the detail of level can be decided by the weight of bintree node calculated in advance and the distance from node to viewpoint.

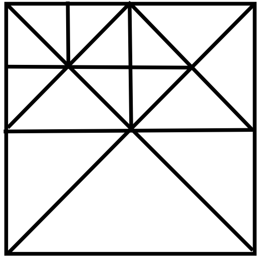


Fig6 bintree partition

Binary Triangle Trees are used to store the splitting process. Binary Triangle Trees can be defined as:

```
struct TriTreeNode
{       TriTreeNode *LeftChild;
        TriTreeNode *RightChild;
        TriTreeNode *BaseNeighbor;
        TriTreeNode *LeftNeighbor;
        TriTreeNode *RightNeighbor;
};
```

Figure.7 shows the topologic relationship of Binary Triangle Trees.

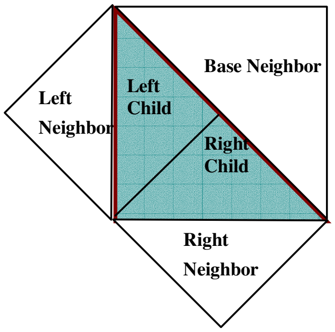


Fig.7 topologic relationship of binary triangle trees

The following Figure.8 is LOD DEM patches render based on view range culling.

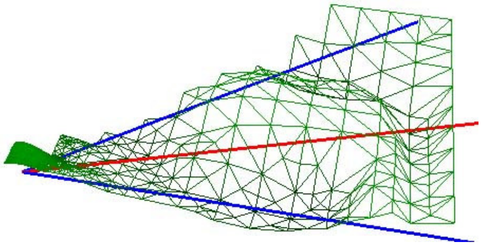


Fig.8 DEM patches culling

Figure 9 shows a birds-eye view of the LOD terrain.

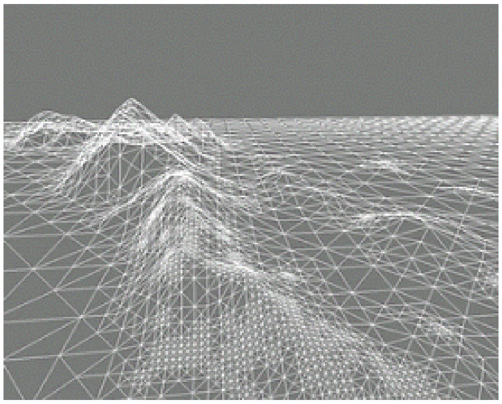


Fig9 terrain LOD rendering

4.Design and Implement of 3D Rendering Engine

Base on some computer graphics algorithms and 3D GIS principles, we designed and implemented an advanced 3D visualization engines: **PowerCity3D**. The graphic API is OpenGL. Using **PowerCity3D,** users can produce a good quality of 3D city scene and inquire object information base on Access data-base.

4.1 advanced modeling techniques

In **PowerCity3D,** Some advanced graphics algorithms was applied.

① Particle systems

A particle system is defined by a collection of particles that evolves over time. The evolution is determined by applying certain probabilistic rules to the particles: they may generate new particles, they may gain new attributes depending on their age, or they may disappear from the scene. They also may move according to either deterministic or stochastic laws of motion. Each object in particle systems has the following attributes: position, velocity, color, lifetime, age, shape, size and transparency. In our 3D engine, Particle systems are used to model fire, fog, smoke, and water.

Figure 10 shows the procedure of particle is generated. Figure 11 shows the fountain based on particle systems in our 3D engine.

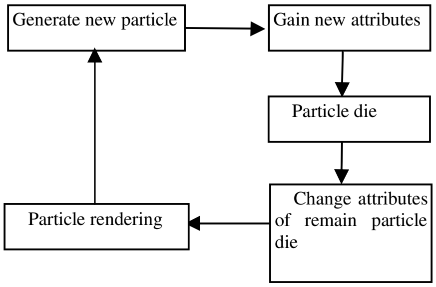


Fig10 the procedure of particle is generated

Fig11 particle system model fountain

② Billboard modeling

Billboard is a new approach for extreme simplification in the context of real-time rendering. Using Billboard, 3D models are simplified onto a set of planes with texture and transparency maps. Xavier use Billboard cloud to simplified 3D model. [Xavier 2003]. Aleks realized interactive vegetation rendering with a simplified image-based rendering approach based solely on alpha-blend textured polygons [Aleks 2000].

In this paper, we use billboard to rendering plants, street lamps, smoke, buildings and so on. We first replace large sets of faces by a texture plane which is generated by projecting a texture on a plane (Figure 12). Then we rotate the plane to adapt to the viewpoint (Figure 13).

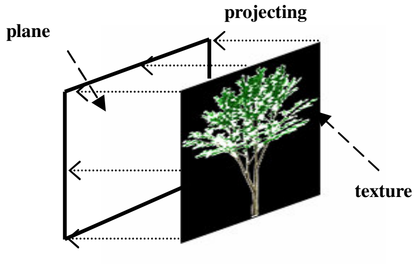


Fig12 Billboard projecting

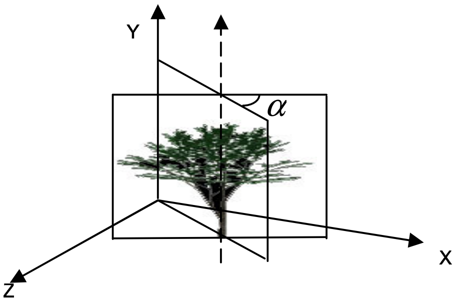


Fig13(a) Billboard rotate

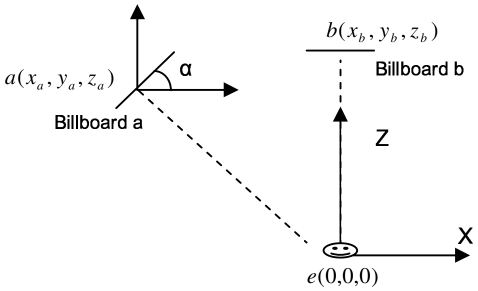


Fig13(b) Billboard rotate in XZ

Fig13 view-dependent Billboard rotate

The α can be defined as:

$$\alpha = 90^o - \tan^{-1}(\frac{x_a}{z_a}) \qquad (3)$$

Where, (x_a, y_a, z_a) is the Billboard' coordinate.

Figure 14 shows the billboard rendering in 3D scene..

Fig14 Billboard in 3D scene

4.2 Information inquire in 3D scene

Information inquire in 3D scene is one of characters of 3D GIS. Because of the complexity of 3D graphics projection, information inquire of 3D GIS is more difficult than 2D GIS. In this paper, we first obtain the 3D coordinate from the 2D screen coordinate the untransform methods. Then we implement 3D information inquire and analysis in 3D scene Figure 15 shows the procedure of coordinate transformation from model coordinate to screen coordinate.

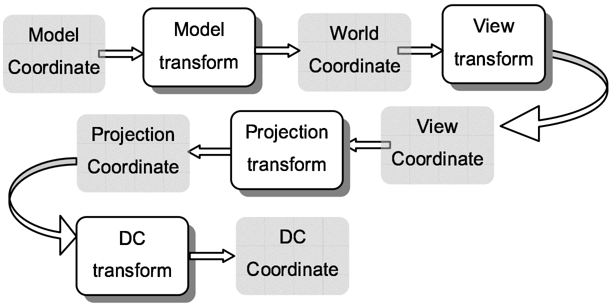


Fig15 coordinate transform pipeline

The transform can be defined as:

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = m_m m_v m_p m_d \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \qquad (4)$$

Where m_m is model transform matrix, m_v is view transform matrix, m_p is projection transform matrix, m_d is DC transform, matrix, $(x, y, z, 1)$ is DC homogeneous coordinate, z is z buffer, $(X, Y, Z, 1)$ is model homogeneous coordinate.

When we inquire information in 3D scene, we can obtain DC coordinate directly from computer screen. If we known the transform matrix, the model coordinate can be obtained by untransform. So the key is the transform matrix obtain. Because it is OpenGL API to be used to render the 3D scene, we can obtain these matrixes by matrix stack in which store the transform matrix during the procedure of rendering. Figure 16 shows the OpenGL matrix stack operation:

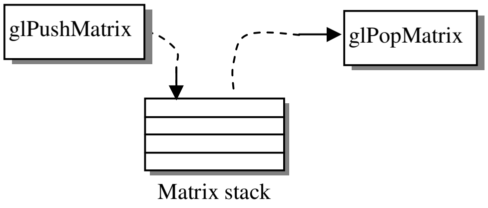


Fig 16 matrix stack operation

If the DC coordinate and transform matrix is known, we can calculate the model coordinate and complete the information inquire. Figure 17 shows the procedure of inquire coordinate.

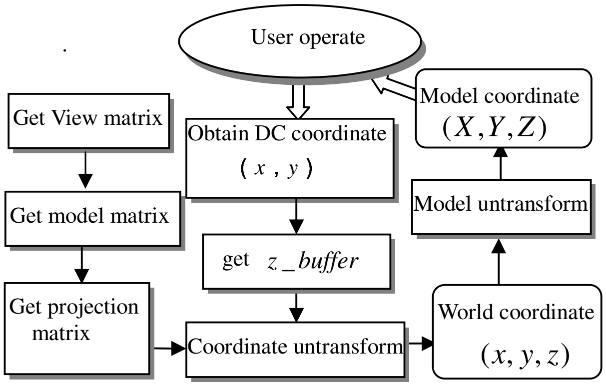


Fig17 Coordinate inquire

4.3 3D Rendering Engine

According to the 3D rendering algorithms, we design and implement 3D rendering engine in OO(oriented object) idea. In PowerCity3D, user can use following functions: object modeling, scene creating and rendering, ramble, advanced modeling, collision detection and respond, information inquire and analysis based database and so on. Figure 18 shows the structure of 3D engine.

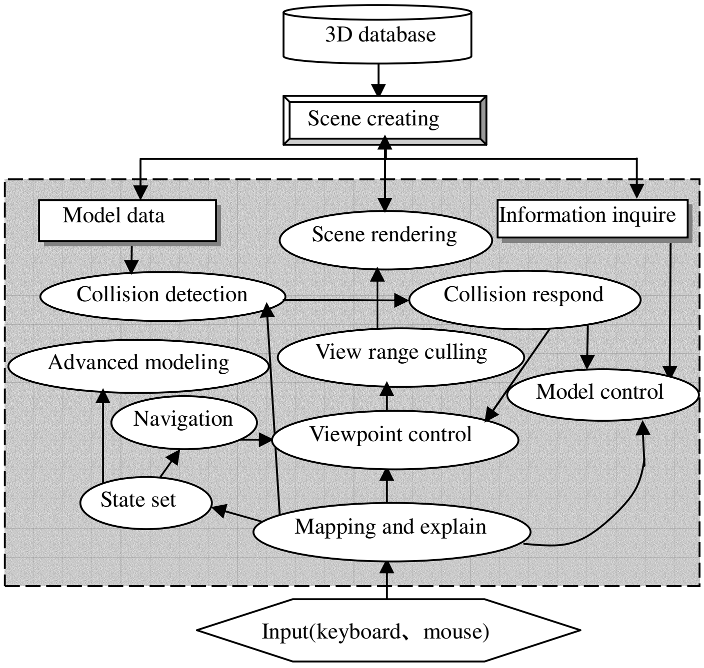
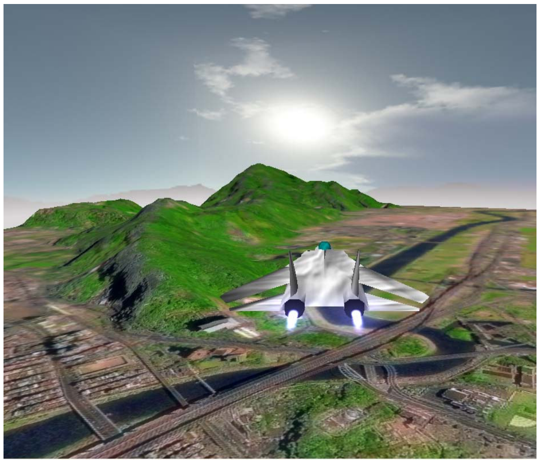


Fig18 Structure of 3D engine

5.Results and Conclusions

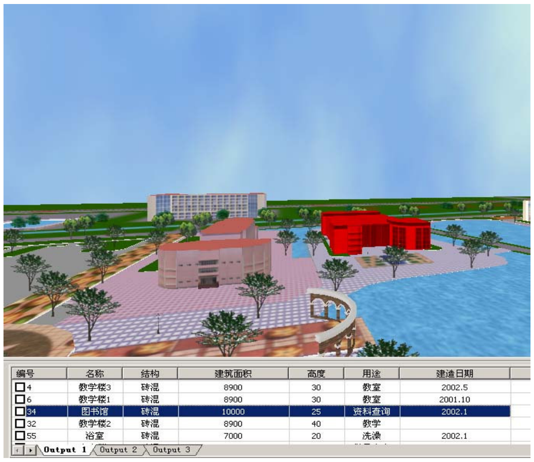
Currently, 3D visualization is a hot technology and it be applied in many field. In this paper, we discuss and research some key techniques in 3D visualization. Based on these researches, a 3D rendering engine PowerCity3D be designed and implemented. The 3D engine be applied in many project. Figure 19 shows some results created by PowerCity3D.

The results show that the 3D engine can render highly reality 3D scene and user can obtain highly immersion. User can ramble in 3D scene and inquire information based on database. In the future, we will pay more attention to study in large dataset accelerate rendering and object action simulation.


(a) Terrain rendering


(b) Smoke rendering


(c) city scene


编号	名称	结构	建筑面积	高度	用途	建造日期
☐ 4	教学楼3	砖混	8900	30	教室	2002.5
☐ 6	教学楼1	砖混	8900	30	教室	2001.10
☐ 34	图书馆	砖混	10000	25	资料查询	2002.1
☐ 32	教学楼2	砖混	8900	40	教学	
☐ 55	浴室	砖混	7000	20	洗澡	2002.1

(d) Information inquire

Fig 19. 3D scene in PowerCity3D

REFERENCES

References from Journals:

Chang Ge, Qian Zengbo, Xu Qing, 2000, *Virtual City Modeling for Visualization*, Proceeding of ICIG'2000, pp. 689-695.

Lindstrom Peter et al, 1996, Real-Time, *Continuous Level of Rendering of Height Fields*, Proceeding of ACM SIGGRAPH 96, pp.109-118.

Rossignac,J. , Borrel,P.,1993, *Multiresolution,3D Approximation for Rendering Complex Scene* , Second Conference on Geometric Modeling in Computer Graphics，.pp.453-465.

Duchaineau M. et al. , 1997, *ROAMing Terrain: Real-time Optimally Adapting Meshes*, Proceedings of the 8th conference on Visualization '97, pp.81-87.

Hamann B.A，1990，*A data reduction scheme for triangulated surface*. Computer Aided Geometry Design ,11(2).pp.197-214.

Hoppe H, 1996, *Progressive meshes*, Computer Graphics(SIGGRAPH Proceeding),pp.99-108.

Hoppe H, 1998, *Smooth View-Dependent Level-of-Detail Control and its Application to Terrain Rendering*, IEEE Visualization 1998, pp.35-42.

Xavier Décoret et al, 2003., *Billboard Clouds for Extreme Model Simplification*, Proceedings of ACM SIGGRAPH 2003, pp. 689 – 696

Aleks Jakulin, 2000, *Interactive Vegetation Rendering with Slicing and Blending*, Proceeding of EUROGRAPHICS 2000 (short presentations).

References from Books:

Zheng Nanling, 1999 *Computer Vision and Pattern Recognition*, National Defence Industry Press,Beijing

Foley James D. et al., 2002, *Computer Graphics: Principles and Practice Second Edition in C*, China Machine Press, Beijing.

Woodl, W. and Anaheim, Dr. ，1999, *Fundamentals of Real Time Modeling*， MultiGen,Inc， 2536, CA 92801,USA