

Architectural Design Virtual Simulation Based On Virtual Reality Technology

Fang Li

Hubei Normal University, Huangshi, 435002, China
duoduo530835181@foxmail.com

Abstract

As the modeling of the architectural system is difficult and inaccurate, the 3D modeling of the light in the architectural system is studied by using Java3D technology. Using java3D perspective projection technique, the observation coordinate system is determined, and real-time viewpoint setting as well as roaming processing is realized. Through multiple view technique, we can observe the movement of building bodies from multiple aspects to obtain vivid simulation effects of light' behavior in the architectural system. Simulation results show that the virtual simulation platform developed based on java3D can improve the accuracy and naturalness of architectural system simulation.

Keywords: Virtual Simulation, Java3D, Architectural System

1. Introduction

Virtual simulation is to construct interactive, integrated and life-like 3D virtual environment by using computer hardware, software and various sensors [1]. It is widely used in areas like scientific computing visualization, aviation, medicine, construction, art and entertainment, design and planning, education, and so on [2]. With the rapid development of computer and visualization technology as well as the development from 3D virtual simulation to large-scale virtual simulation, Virtual simulation is more and more widely used in various areas, and different areas put much higher requirements on the real-time capability, verisimilitude, and reliability of computer simulation.

Traditional architectural design usually uses the method of transformation of thinking from two-dimensional graphics to three-dimensional graphics. The designers of the design objects lack full range of thinking and narrow field of vision, lack of integration of thinking skills. Collapse of the phenomenon of new design ideas often appears between the transformations of thinking from two-dimensional graphics to three-dimensional graphics for the designers. As the two-dimensional drawings or three-dimensional models reflect only part of information of the building, so that it often covers a number of design defects. Virtual reality technology which introduced into building design allows designers or owners immersive experience the building built. Designers can be fully in accordance with their own ideas to construct the "virtual" room, and can transform the point of view anyway in room which understanding the pros and cons of building design, and constantly make changes until satisfied.

Virtual reality technology allows the artistry of the building that can be emphasized and highlighted, and if the art features of the building are becoming pure, that is the building art. But in the building design process in the past, the building art is difficult to achieve. The building arts with virtual reality technology can be achieved in the field of architectural design. The simulation of virtual building model is always a hot research area in architecture simulation field [3]. Domestic scholars have studied on this area for a

long time especially towards near-architecture building environment virtual simulation, building detection scene simulation, scene simulation, building environment reality modeling, and starry sky knowledge teaching [4]. This paper develops a large-scale virtual simulation research platform of the architectural system by using the graphics software development kit (SDK) provided in Java3D, realizes key 3D browse technologies like viewpoint smooth movement, texture mapping, rotation and local zoom, and simulates the motion of building bodies in the architectural system.

2. Principle of Generation of Perspective Projection Vision Field

Java has advantages like good code portability, high programming efficiency, and fast execution speed [5]. Java3D provides interactive 3D graphic application programming interface (API), through which developers can build and operate 3D geometric objects from higher level without being limited by hardware and software. Users only need to consider graphic objects in themselves, without facing the problem of realizing the object from points, lines and planes, and without considering complicated graphic problems such as illumination and coloring either. Java3D also possesses features like hardware automatic acceleration and scene automatic management [6-7]. For example, with the change of observation points, some part is shown or hidden automatically.

The mapping between 3D points can be regarded as the rotation of a point in a solid object from one position to another position. The mapping from 3D virtual world scenes to 2D images is realized through projection transformation. Java3D provides parallel projection and perspective projection. The former is also known as orthogonal projection, which is carrying out along the coordinate axis from three directions to obtain projection transformation. The size of the object observed after adjusting viewpoints will not change with the distance from viewpoints, therefore perspective projection can be adopted to realize the mapping from 3D scenes to 2D graphic, and can reflect the size of objects as the distance from viewpoints changes as well as the distance between objects and observers. For example, generally objects closer to viewpoints look bigger than those far from viewpoints. So when using perspective projection, the size of objects changes with the distance from viewpoints to produce more vivid effects. The key to acquire good projection effects is to set proper relative data in vision field. The principle of perspective projection vision field is shown as Figure 1.

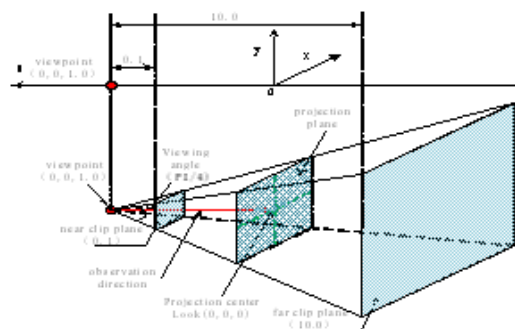


Figure 1. Generation Principle of Perspective Projection Vision Field

Use set Compatibility Mode Enable (true) of View object to start compatibility mode in Java3D. Java3D provides observation matrix and projection matrix to set relative

parameters for vision field. Observation matrix needs to set three parameters: viewpoint, observation direction (using vector to represent), and observation center. They are produced through create View(Canvas3Dcv, Point3d eye, Point3d look, Vector3d up) to satisfy different views, then look At(eye, look, up) method of viewing of Transform3D object is used to transfer relative parameters to viewing for storage, finally, set VpcToEc (Transform3D Viewing) of view object is used to set observation matrix. Projection matrix is set by using frustum (double left, double right, double bottom, double top, double near, double far) of projection of Transform3D project (coordinate (left, bottom, -near) is to define the lower left corner of near clip plane, (right, top, -near) is to define the top right corner of near clip plane, -far is to define the depth of far clip plane). Then set the projection matrix of left eye by using set LeftProjection (projection) of view object to pass relative parameters (Similarly, the projection matrix of right eye can be set using set RightProjection (projection)).

Use super structure of java3D scene graph to add View Platform nodes and Canvas3D into View object. Set physical body and physical environment nodes using related methods of view object. Then take View Platform as child node and add it to Branch Group node to complete the construction of scene graph. Using create (Canvas cv, Point eye, Point3d look, Vector3d up) to pass different parameters to cv canvas to form different views embodies the powerful advantage of virtual simulation.

3. Simulation of Building Bodies in the Architectural System

This paper uses sphere class of java3D to construct the model in local coordinate system. For example, the Architecture can be constructed by setting properties like radius in local coordinate system, then use Texture Loader class to load texture, and establish one-to-one relationship between object pixel and texture element to realize texture mapping through texture mapping technique provided by java3D. To better simulate the motion of light, and make it satisfy Kepler's laws (light move around the light, and the area the curve connected to the Light swept in unit time is same), the graphic content technique in java3D is adopted to construct the orbit. Taking Architecture's orbit as an example, the average revolution period of the Architecture is one year (365 days). Divide the orbit area into 3650, that's, set 3650 points in the Architecture's orbit. Based on given conditions and the standard elliptic equation

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1'$$

$\theta(t)$ of the corresponding point is calculated. Then the coordinate of the corresponding point is obtained through angle θ .

$$S_{X0SXt} = \frac{1}{2} ab \theta(t) - \frac{1}{2} cb \sin(\theta(t))$$

(where a, b is the long axis and short axis of the ellipse respectively, c is the x-coordinate of the ellipse focus, S_{X0SXt} is the area swept by a building in unit time). Based on this equation and known surface area, we can calculate the corresponding deflection angle $\theta(t)$ [8]. As the above equation is a transcendental equation, it is very difficult to calculate θ using ordinary methods. Here, we adopt the binary approximation algorithm to obtain an approximate solution, and set a precision value (PRECISION=0.0000000001 in this paper). The detailed steps of the algorithm are described below:

(1) Determine $f(\theta)$ (here,

$$f(x) = \frac{1}{2}ab\theta(t) - \frac{1}{2}cb\sin(\theta(t)) - \frac{\pi ab}{3650};$$

(2) Discuss the monotonicity of $f(\theta)$, $\theta \in (0, \pi)$, define $x_1=0$, $x_2=\pi$, $x_0=(x_1+x_2)/2$, continuously take the median value x_0 of angle θ to substitute into $f(\theta)$;

(3) Compare $f(\theta)$ and 0, if $f(\theta)>0$, then let $x_2=x_0$, otherwise let $x_1=x_0$;

(4) Use while loop to control the process, if the calculated x_0 is less than or equal to PRECISION, then it is the deserved approximate solution, otherwise, go back to step (3), repeat until the approximate solution is acquired;

(5) Calculate the deflection angle θ_1 of the area swept in the first unit time, and then calculate θ_2 of the area swept in the second unit time, and in a similar way, obtain all approximate solutions.

The calculated deflection angle $\theta(t)$ is not our ultimate goal. To simulate the Architecture's revolution orbit, we need to calculate the world coordinate of 3650+1 points on the orbit (each point is corresponding to each share). Then use java3D graphic content technique to connect and calculate the coordinate. This paper uses Transform Group class and Behavior to complete the simulation process. Behavior object defines the action to be executed when a behavior is activated. Behavior should be triggered by special objects under awaken condition. Using the behavior to conduct transformation in given time cycle can realize Architecture's revolution simulation.

As the calculated deflection angle $\theta(t)$ is not the same angle as φ in parameter equation $x=a*\cos\varphi$, $y=a*\sin\varphi$, this paper proposes an improved algorithm to handle this problem. As shown in Figure 3, draw an inscribed circle and circumscribed circle of the elliptic orbit respectively, connect OA, which intersects with the inscribed circle at point B, draw a line vertical to X axis through point A, and the intersection point is M. The x-coordinate of the ellipse moving point X_t is the x-coordinate of point A, and y-coordinate is the y-coordinate of point B. As the ellipse equation and the deflection angle $\theta(t)$ are known, it is easy to calculate the coordinates of point A and B, and thus calculate the coordinates of point X_t .

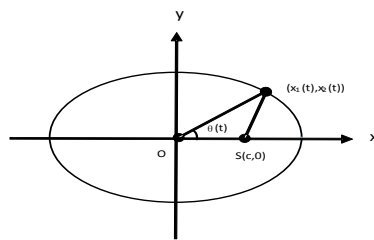


Figure 2. The Sketch Map of Architecture's

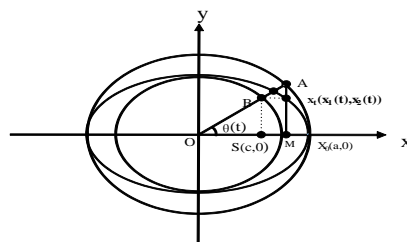


Figure 3. The Sketch Map of Improved Construction Algorithm Architecture' Orbit Construction Algorithm

Programming language of virtual architectural design teaching process and implementation.

Furthermore, in order to show the equipment work processes more vividly, such as internal flow channel and connectivity relationship of the pump and valve, three-dimensional arrow models which fit the flow channel are established by 3Dmax, with the texture of liquid flowing effect, through GPU programming method to define custom Shaders, so as to achieve showing the hydraulic oil trend by three-dimensional arrows with animated textures, Cg code is as follows:

CGPROGRAM

.....

v2fvert (appdata_base v) //vertex function

{

v2f o;

o.pos=mul(UNITY_MATRIX_MVP,v.vertex);

//view port transformation by projection matrix

o.uv=TRANSFORM_TEX(v.texcoord,_MainTex); //texture coordinates

return o;

}

float4 frag (v2f i) : COLOR //fragment function

{

float4 outp;

float4 defaultcolor = float4(0.6,0.6,0.6,0.26);

//define default color when texture is hidden

float4 texCol = tex2D(_MainTex,i.uv);

//define texture and make assignment

//_MainTex_ST.xy corresponds to Tiling , _MainTex_ST.zw corresponds to Offset

//changing _Lenth value by external scripts can realize texture animation effect

if(i.uv.y<_Lenth*_MainTex_ST.y)

outp =texCol;

//with in Lenth value display the texture

else

outp =defaultcolor;

//out of _Lenth value display the default color

Returnoutp;

}

ENDCG

4. Architectural Design Image Recognition and Registration

First make recognition objects, VuforiaAR SDK support local identification database and cloud identification database, these two methods have their own strengths, local database don't need network to support and complete recognition faster, but a limited number of objects can be identified and must be made in advance, in contrast, the cloud identification may support more than one million objects, but this method must rely on the high-speed internet connection, considering the conditions of ship using, decided to use local identification database, we need to upload the images or markers which need to associate with augmented contents to Qualcomm "Target Manager" and generate local identification data set.

Then put prefabs such as ARCamera, ImageTarget *etc.* into the scene according to need, delete the original MainCamera object, drag and drop the corresponding augmented contents to be sub-objects of ImageTarget and adjust the world coordinates and scaling, so that when the target image is identified the augmented content will be registered under the definition of the relative position of the scene, set the appropriate lighting to heighten the atmosphere, finally configure the parameters and write or modify C # script to complete the desired function, some parameters shown in Figure 4.

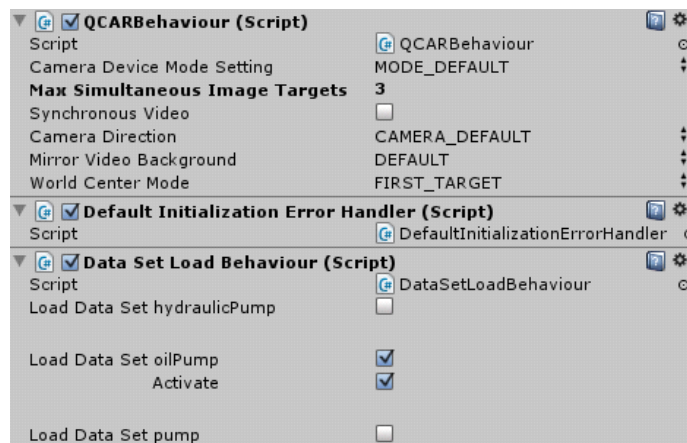


Figure 4. Some Important Parameters Configuration

The modeling process is completed in 3DSMAX. Drawing the map through AutoCAD, then making the concrete frame through the extrusion function in the 3DS MAX, and then adding object model into the scene. The system used Cult3D as interactive software, first installed three program of Cult3D on computer, then selected the export command in file menu in 3DS MAX, and derived type as C3d, In this way, the file format of the model and animation can be identified by Cult3D Designer. The C3d files are added to the Cult3D Designer with appropriate interaction, then output into a. Co file format, the specific steps are as follows:

First of all, in the file add Cult3D designer file menu by introducing just manyou.c3d file. When the Scene Graph (scene) window will appear in3ds Max in the production of all models of objects, such as cameras, each view, all the parts, materials and so on, can also be here to be motion the same object component, method is in need of group of an arbitrary object right-click on the mouse, by new dummy generates a virtual object, the object will be grouped into one. Below the Scene Graph (scene) of the window is Action window, this includes all Cult3D perform actions, we often use actions such as motion rotation, zoom, mobile *etc.* Set the mouse and keyboard interaction as Figure 5.

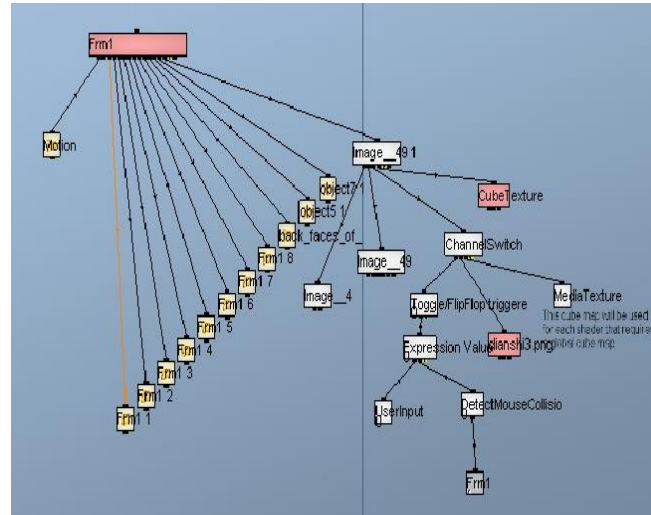


Figure 5. Interactive Interface

4.1. Architecture-Light Revolution Modeling

Architecture-Light revolution modeling has already been discussed. There is a $5^{\circ} 09'$ included angle between the architecture revolution orbit(ecliptic plane)and the light revolution orbit(lunar orbit). It's not clear to view vivid 3D orbit through direct simulation of their revolution orbits. Addition of included angle model into the scene can display true to life orbit effect. Use Geometry class

```
IndexedQuadArrayqa=new
IndexedQuadArray(10IndexedQuadArray.COORDINATES|IndexedQuadArray.NORMA
LS|IndexedQuadArray.COLOR_3, 24)
```

To generate an instance that can set vertex coordinate, normal vector, and vertex color to construct included angle model. The ecliptic plane is the bottom of the model. Set the first vertex as (x, y, z)=(0, 0, 0), the length of the hypotenuse is 1, the width is 1.8d. Calculate the coordinates of other vertex through the inclination angle $5^{\circ} 09'$ (Translate into 5.15°), as equation (1) shows:

$$\begin{bmatrix} \text{point}[0] \\ \text{point}[1] \\ \text{point}[2] \\ \text{point}[3] \\ \text{point}[4] \\ \text{point}[5] \\ \text{point}[6] \\ \text{point}[7] \\ \text{point}[8] \\ \text{point}[9] \end{bmatrix} = \begin{bmatrix} \text{point3d}(0,0,0) \\ \text{point3d}(\text{Math.cos}(5.15*\text{Math.PI}/180),0,0) \\ \text{point3d}(\text{Math.cos}(5.15*\text{Math.PI}/180),0,1.8\text{d}) \\ \text{point3d}(0,0,1.8\text{d}) \\ \text{point3d}(\text{Math.cos}(5.15*\text{Math.PI}/180),\text{Math.sin}(5.15*\text{Math.PI}/180),1.8\text{d}) \\ \text{point3d}(\text{Math.cos}(5.15*\text{Math.PI}/180),\text{Math.sin}(5.15*\text{Math.PI}/180),0) \\ \text{point3d}(0,0,0) \\ \text{point3d}(0,0,1.8\text{d}) \\ \text{point3d}(-\text{Math.cos}(5.15*\text{Math.PI}/180),0,1.8\text{d}) \\ \text{point3d}(-\text{Math.cos}(5.15*\text{Math.PI}/180),0,0) \end{bmatrix} \quad (1)$$

To better distinguish, use red to represent the lunar orbit, and blue to represent the inclined plane. At this time, the included angle model is a solid, covering related revolution orbit, thus the simulation effect is not good. Use the method of visual object provided by Transparency node:

TransparencyAttributes=newTransparencyAttributes(TransparencyAttributes.BLENDED,0.3f);Set the transparency of the included angle model as 0.3f, and a good effect of each layer and between layers is achieved. At the same time, light is also an important part of computer graphics. Vivid 3D effect needs light rendering. As there is a very long distance between the Light and the Architecture or other light, the lighting effect projected is similar to direction light. The simulation of light adopts direction light. Define DirectionalLight lightD = new DirectionalLight(); Set light color as white, and make light direction changes with architecture revolution through Behavior, the effect is shown in Figure 6.

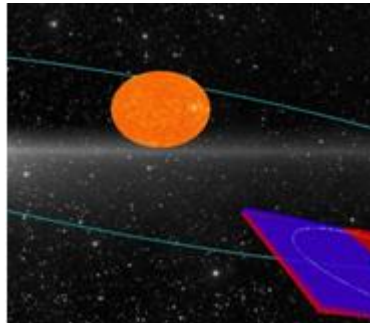


Figure 6. Simulation Effect with Enhanced Light Rendering

5. Conclusion

In this paper, Java3D as the technological foundation of modeling and simulation, this paper develops a multi-function architectural system simulation platform. Multi-view technique in Java3D is used to simulate the morphology and motion orbit of each building, which can display vivid 3D Configuration, moving mode and texture features of different building bodies in the architectural system. Besides, using Java3D can construct more complicated scenes to satisfy more demands and more professional applications.

However, we need to improve in the following ways in next study: With consideration of the development of current and future building technology, especially deep building exploration technology, it is necessary to build virtual simulation platform with more complete functionality and faster data processing speed and a platform should be provided with the function of spatial-temporal data modeling and efficient, fast and intelligent data processing.

Acknowledgment

This work was supported by The Education Department of Hubei Province (Research on humanities and social science research project NO. 14Q079)

References

- [1] Hong Z. Y. and D. Liang, "Computer graphics: Application of Java2D &3D", China Machine Press, (2008).
- [2] Ge S., "An Improved LOD Algorithm of Virtual Reality Scene Based on Java3D", Wuhan University of Technology, (2005).
- [3] C. J. Yong, "Java3D based Virtual Reality Modeling Method", Application Research of Computers, vol. 19, no. 5, (2002), pp. 33-35.
- [4] Xu Y., "Augmented reality application research of teaching", Chinese distance education, vol. 10, (2007), pp. 208-215.

- [5] L. Huaming, "A Study on the Generation Method of Virtual Reality Three-dimensional Scene Based on Java3D", Jilin University, (2011).
- [6] Z. Jiandong, "Virtual reality technology", Tsinghua university press, vol. 8, (2009), pp. 8-9.
- [7] D. Guoqing and L. Yuqing, "Computer Simulation", vol. 28, no. 11, (2011), pp. 26-30, 86.
- [8] G. Kunlin, "Dissemination of Mathematics", Chengdu: southwest Jiaotong University, 7, vol. 32, no. 2, (2001), pp. 13-15.

Author



Fang Li, lecture form Huangshi of Hubei Province, Work in Hubei Normal University.

Research interests: architecture art design.

