The Fundamental Theory of True 3D Virtual Reality and The Development of Scene Structure Research

Hongli Liu [1, a *], Doudou You [2,b], Juan Cai [3,c] and Caixia Liu [4,d]

[1,2,3,4] College of resource environment and tourism, Capital Normal University, Beijing, China

[a*] hongli_liu2008@sina.com, [b]452818220@qq.com,
[c]1048111868@qq.com,[d]1261078957@qq.com

Keywords: true 3D virtual reality, scenarios theory, modeling theory, scene construction development

Abstract. This paper describes the fundamental theory of true 3D virtual reality which involves the theory of scene structure, modeling and scene construction. It presents the true 3D virtual reality development process, construction basic graphics library of 3D and simplified programming.

Scene Structure Theory

Virtual reality is a reality of virtual, which can be divided into several objects and should be classified to data processing. Computer system is composed of different software and hardware, calling for the object category of real world should be entered into the corresponding computer system. In order to make calculation results presenting in the display system and realize man-machine communication, display system and computer must be matched completely. Finally, people should have an integrated control of the above system, that is to say the conversation between the user and the virtual scene. This process can be expressed with mathematical formula as follows:

Setting World as virtual reality world, this can be divided into different sub object W (W_1, W_2, W_3...... (Wn); M is a computer system M (M_1, M_2, M_3...... Mn); Q is the display system Q (Q_1, Q_2, Q_3...... Qn).

Defining the virtual scene which needs to be built: Watch: World →W (W_1, W_2, W_3...... Wn)

Introducing the classification of virtual worlds into the computer system and defining the input function as f (x): f(x): W → f(M)

Projecting the computer processing results onto the display system and defining the performance function as S (y): S (y): F (M) → S (Q)

As for conservation between human and virtual scene dialog, defining the control function as A (Z):

A (z): S (Q) * M + Z, （Z : human factor）

[World, W, M, Q, Watch, f (x), S (y), A (z)] constitute the theoretical framework of virtual reality. In the essential process of F(x) function operation, the important content is modeling the virtual scene, which refers to geometric modeling, motion modeling and physical modeling.

Modeling Theory

Geometric modeling

The following mainly discusses the basic theory of modeling, such as geometric modeling, building the "skeleton" (polygon, triangle, point, line and region of arbitrary shape) and "skin"(surface texture, surface light intensity of illumination and color). No matter how complex the virtual scene is, it can be spliced by polygonal or simpler triangle, which can constitute a variety of objects in the virtual scene together, forming the mesh "skeleton" (Fig.1).

These polygons and triangles intertwined, having many common vertexes, radiating to different region from one point, which can improve the speed in the drawing process, as well as data transfer speed. It is better to deduct a function (higher-order function) to express the scene (or shape) with regard to extremely uneven scenes calling for high accuracy. For example, the functions used to express the surface shape (local spline) are as follows:

$$\begin{aligned} x(u) &= a_x \cdot u^3 + b_x \cdot u^2 + c_x \cdot u + d_x \\ y(u) &= a_y \cdot u^3 + b_y \cdot u^2 + c_y \cdot u + d_y \\ z(u) &= a_z \cdot u^3 + b_z \cdot u^2 + c_z \cdot u + d_z \end{aligned} \quad (1)$$

Among them, u is variable-value between 0 and 1, $a_{x,y,z}$,$b_{x,y,z}$,$c_{x,y,z}$,$d_{x,y,z}$ are constant. The shape expressed by function is smooth and continuous, but it is very difficult to change it partly. (Foley, etc. 1994)

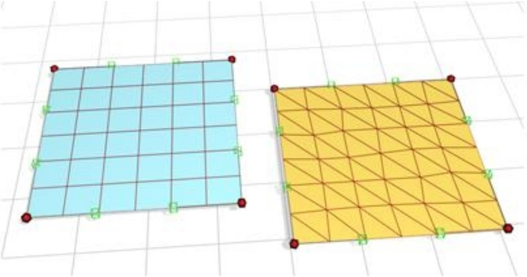


Fig.1. Polygons and triangle interweaving

Motion modeling

Motion modeling is mostly used to determine the location of 3D objects in the scene coordinate system and their motion in the virtual environment. The object motion is restricted by hierarchical relationships of father and son, moreover the parent object's movement can affect sub object. The other aspect of motion modeling is to set up the way of observing the world namely virtual camera motion. Finally, the camera image needs to be projection transformed into a two-dimensional display window to provide users with visual feedback.

When modeling the object surface, vertex coordinates are using object coordinate (x, y, z) which is tied on the object and usually located in the center of gravity, whose direction along the symmetry axis direction of the object. As the object is moving in the virtual world, its coordinate system is moving too. No matter how the object's position is changed in the scene, the position and direction of the object's vertex coordinates in the object coordinate system can remain the same.

Fig.2.(a) shows the virtual object's coordinate transformation (Burdea, 1993), which is called object 1, bound in Cartesian coordinates (i_1, j_1, k_1), with origin is located in the center of the object, covering three prism axis of symmetry. The absolute position of object 1 is relative to a fixed coordinate system, which is called world coordinate system (i_w, j_w, k_w). The transformation formula between the coordinate system and the world coordinate system of object 1 is as follows:

$$T_{W\leftarrow 1} = \begin{bmatrix} i_{W\leftarrow 1} & j_{W\leftarrow 1} & k_{W\leftarrow 1} & P_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$i_{W\leftarrow 1}$, $j_{W\leftarrow 1}$ and $k_{W\leftarrow 1}$ are 3×1vectors describing the components in the world coordinate system. P_1 is the position vector changing from O_w to O_1.

If the virtual prism is static, the matrix described in the formula (2) is fixed. Whereas, if the object 1 is motorial, it is transformed into a function of time $T_{W\leftarrow 1}(t)$:

$$T_{W\leftarrow 1}(t) = \begin{bmatrix} i_{W\leftarrow 1}(t) & j_{W\leftarrow 1}(t) & k_{W\leftarrow 1}(t) & P_1(t) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

The position of any vertex V_i in a coordinate system (i_1, j_1, k_1) can be obtained from the 3D model file of the object 1. When the virtual prism is moving, the vertex's changing position $V_i^{(w)}(t)$ in the world coordinate system can be expressed as: $V_i^{(w)}(t) = T_{w\leftarrow 1}(t) V_i^{(1)}$ (4) $i = 1, \mathbf{L}, n$ (n is the total number of vertices in the object file).

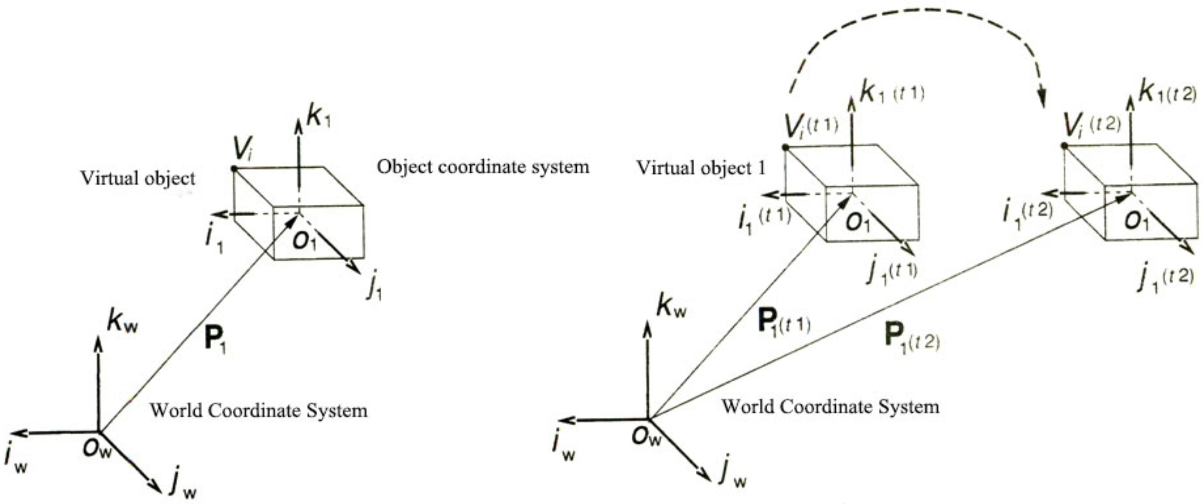


Fig.2.(a) Fig.2.(b)

Fig.2.(b) shows the case when object 1 is doing translational motion and the only changing in $T_{w\leftarrow 1}$ is the position vector $P_1(t)$, so $V_i^{(w)}(t)$ can be simplified as:

$$V_i^{(w)}(t) = \begin{bmatrix} 1 & 0 & 0 & p_{1i}(t) \\ 0 & 1 & 0 & p_{1j}(t) \\ 0 & 0 & 1 & p_{1k}(t) \\ 0 & 0 & 0 & 1 \end{bmatrix} v_i^{(1)} \qquad (5)$$

The object coordinate system is aligned with the world coordinate system in this case, so the rotation sub-matrix $R_{3\times3}$ is a unit matrix. By repeating the same transformation to all the vertices, the object is moved to a different position at the moment t_2. Multiplying the inverse of homogeneous transformation matrix and formula (5) can shift the object to the origin.

When an object is placed in a virtual world, it is necessary to zoom it sometimes. Scaling is actually completed by three operations: (I) translating the object to the origin of the world coordinate system; (II) scaling object; (III) translating the object back to the original position. Translation is achieved by multiplying the homogeneous transformation matrix given in the formula. Scaling is given by the formula as follows:

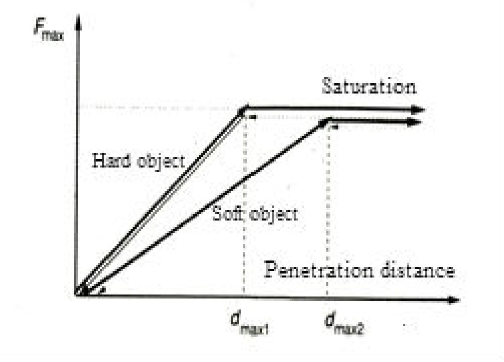
$$T_{scaling} = \begin{bmatrix} s_i & 0 & 0 & 0 \\ 0 & s_j & 0 & 0 \\ 0 & 0 & s_k & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (6)$$

s_i , s_j and s_k are scaling factors about the three axis of the world coordinate system. If scaling is uniform, the three factors are equal.

Physical modeling

Physical modeling reflects the physical properties of objects (elastic, reflectivity, hardness, etc.), which makes the objects in the virtual scene more realistic. Any object has flexibility with different sizes, it is necessary to carry out the stress calculation when the user interacts with the object particularly, needing to produce a sense of force. Proportional to the size of the virtual flexible object deformation degree and contact pressure, after the loss of contact pressure, elastic virtual object can restore to the previous geometry, the interaction model is according to Hooke's Law: (Burdea,etc. 1992)

$$F_{constraint} = \begin{cases} K_{object} \cdot d & 0 \le d \le d_{\max} \\ F_{\max} & d_{\max} < d \end{cases} \quad (7)$$



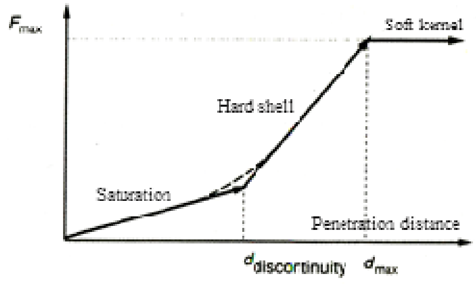


Fig.3. Elastic virtual objects

Fig.4. Hardness uneven objects

K_{object} is the hardness constant of the object, D is the deformation degree of contact point surface (Fig.3).

If the hardness of object (hard inside，soft outside etc.) is not uniform, the relationship between the force and deformation can be approximated expressed as piecewise linear relationship (Dinsmore et al. 1997) (Fig.4).

$$F_{constraint} = \begin{cases} K_1 \cdot d & 0 \le d \le d_{disc} \\ K_1 \cdot d_{disc} + K_2 \cdot (d - d_{disc}) & d_{disc} \le d_{\max} \\ F_{\max} & d_{\max} < d \end{cases} \quad (8)$$

d_{disc} is the surface deformation degree when feeling the hard part inside.

Plastic virtual objects have almost no force to bounce back when deforming, because their surface will remain in the shape. The following formula can be described as:

$$F_{initial} = \begin{cases} K_{elastic} \cdot d & 0 \le d \le m \\ 0 & \text{In relaxation process} \end{cases} \quad (9)$$

After that, $F_{subsequent}$ linear growth is growing to the maximum value of the force within the scope of the contact point. The contact pressure model is given by the following formula:

$$F_{subsequent} = \begin{cases} K_{plastic} \cdot d \cdot u_m & 0 \le d \le d_{\max} \\ F_{\max} & d_{\max} < d \\ 0 & \text{In relaxation process} \end{cases} \quad (10)$$

Rigid body (wall, hard metal, etc.) has great anti elastic in the compression of force. When virtual object and wall are colliding, it will produce large recoil and part vibration, even shock the whole scene. In order to absorb the vibration to the effect of the gradual recovery, damper B is introduced. The proportional model of the elastic object and the plastic object change to the derivative proportional model:

$$F_{constraint} = \begin{cases} K_{wall} \cdot v + B \cdot v & v < 0 \\ K_{wall} \cdot v & v \ge 0 \end{cases} \quad (11)$$

v is speed, deforming from d. $v < 0$ is the object's impact rigid body and $v \ge 0$ is the object's far from rigid body (Fig.5).

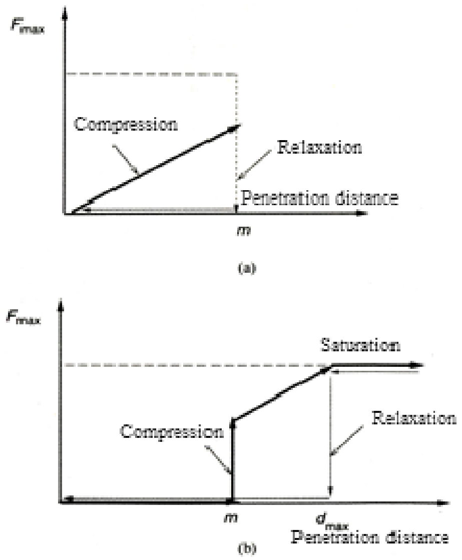


Fig. 5. Plastic deformation

Fig. 6. Scene illumination

Research on The Construction and Development of Virtual Scene

After the completion of the virtual scene modeling (skeleton), designing the light (illuminant) and the scene is illuminated, this is the key of virtual reality simulation. In the virtual scene， some objects are illuminated directly, some are illuminated obliquely, but also some objects are reflected light to illuminate, which important is that many objects are in the shadow of some other objects.

If the surface is water, the surface reflective point is moving with observer's perspective. The light source is more than one when the scene is illuminated by light. There also exists mapping between different light sources (Fig.6).

The light and shade of the scene can be dealt with constructing the polygon mesh of light intensity of each vertex, this normal vector can be calculated by adjacent polygon method. Vertex normal vector is as follows:

$$\vec{N}_V = \sum_{i=1}^{n} \frac{\vec{N}_{\mathrm{i}}}{\sum_{i=1}^{n} \vec{N}_i} \qquad (12)$$

$\vec{N}_V$ is the average normal vector of vertex, $\vec{N}_{\mathrm{i}}$ is the normal vector for each polygon.

Vertex light intensity can be figured out on the basis of the vertex normal vector, the light intensity of the edge can be calculated by vertex intensity .According to the interpolation of the side of the light intensity can get any line of light intensity in the scene scanning (Foley, etc.1990).

$$I_a = I_1 - (I_1 - I_2)\frac{y_1 - y_s}{y_1 - y_2} \qquad (13)$$

$$I_b = I_1 - (I_1 - I_3)\frac{y_1 - y_s}{y_1 - y_3} \qquad (14)$$

$$I_c = I_b - (I_{b1} - I_a)\frac{x_b - x_c}{x_b - y_a} \qquad (15)$$

I is the intensity, and a, b, c are two sides of polygon and in the middle of the point (in a horizontal line s), x, y are Cartesian coordinate and 1, 2, 3 are three scan lines along the Y axis.

Modeling and illuminate the scene are the basis of virtual programming, in order to simplify the programming, it is essential to construct based 3D graphics library and develop three-dimensional graphics engine for constructing visual development platform. This process is presented in Fig.7:

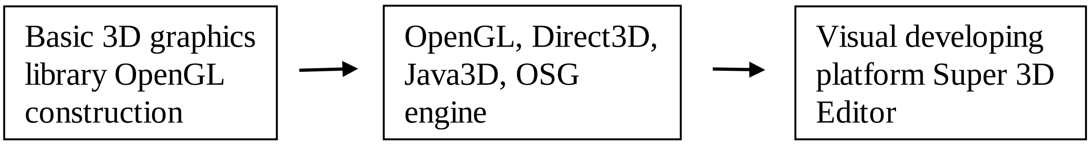

Basic 3D graphics library OpenGL construction
OpenGL, Direct3D, Java3D, OSG engine
Visual developing platform Super 3D Editor

Fig.7. Three levels of 3D virtual reality development

This is the development of true 3D virtual reality.

References

[1] Burdea, G., and P. Coiffet, La Realite Virtuelle, Hermes, Paris，1993.

[2] Burdea, G., E. Roskos, D. Silver, F. Thibaud, and R. Wolpov,“A Distributed Virtual Environment with Dextrous Force Feedback,” in Proceedings of Interface to Real and Virtual World Conference, Montpellier, France, 1992, P255-256

[3] Dinsmore, M., N. Langrana, G. Burdea, and J. Ladeji, “Virtual Reality Training for Palpation of Subsurface Tumors”in Proceeding of IEEE International Symposium on Virtual Reality and Applications (VRAIS ’97), Albuquerque, NM, 1997,P54-60.

[4] Foley,J.,A.van Dam,S.Feiner,and J.Hughes, Computer Graphics: Priciples and Practices,Addison-Wesley,Reading,MA. 1990.

[5] Foley,J.,A.van Dam,S.Feiner,and J.Hughes,and R.Phillips, Introduction to Computer Graphics, Addison-Wesley,Reading,MA. 1994.