Application of Trackball in Visualization of Ionospheric Parameters

Cun Shen& Jing Fan & Zhe Wang & Qingjin Ma
School of Electronic and Information, Wuhan University, Wuhan, Hubei, China

Chunhua Jiang & Chen Zhou
Ionosphere Laboratory, School of Electronic and Information, Wuhan University, Wuhan, Hubei, China

ABSTRACT: In this paper, we designed and developed a visualization software of ionospheric parameters, which is based on International Reference Ionosphere model and OpenGL package. We implemented the free rotation of three-dimensional model by trackball, and wrote a program to implement three-dimensional visualization model of ionospheric parameters using the data obtained from IRI model. This program allows the user to interact with the software by mouse, so that they can observe and understand regional and temporal variation of ionospheric parameters intuitively and comprehensively.
KEYWORD: Trackball; Free Rotation; Visualization; OpenGL

1 GENERAL INSTRUCTIONS

Ionosphere is an important layers of earth atmosphere. It is the inner limit of earth magnetosphere, which is the ionized portion of atmosphere by solar rays. Transmission of radio wave is significantly impacted by ionosphere. As a result, study on the parameters of ionosphere is important. Some typical parameters, such as electron density, electron temperature, and f_0F_2, could be displayed by various colors in a visualization model based on measured results. The change of these parameters can be visually observed in this model.

In order to develop such a visualization model, a software which is able to display the typical parameters of ionosphere is developed. The software architecture is shown in Figure 1. Object-Oriented approach is used to solve the problems in applying OpenGL in MFC framework. A spherical model like Google Earth is then built. Another plane model is also developed to study the change of the parameters with height. The raw data from IRI are displayed in various colors. A nonlinear mapping scheme is used to show the change of these parameters using bitmap. The bitmap is then loaded in the model to implement visualization. Man-computer interaction can be implemented by input from mouse and keyboard in this OpenGL based system. Clear images can be obtained from this model. It is easy for the readers to understand the information in these images.

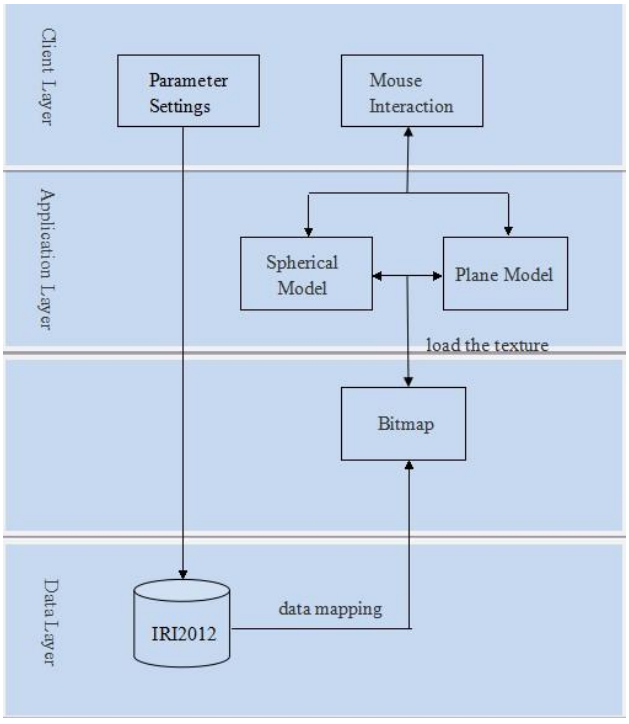

Figure1. Software architecture

Software interface is shown in Figure 2. Through the interactive graphic interface, we can input the parameters of ionosphere, date, time and height information. The results can be displayed in either spherical view or plane view. Although the 3D view under the OpenGL framework is able to show the data, only the front side of the object can be seen. In order to check the whole globe, we propose a method to rotate this object using mouse, which is similar to Google Earth. This method greatly facilitates the users of this software to study the

globe from any perspectives. Therefore, the rotation of the model with a fixed point controlled by mouse is completed in this study

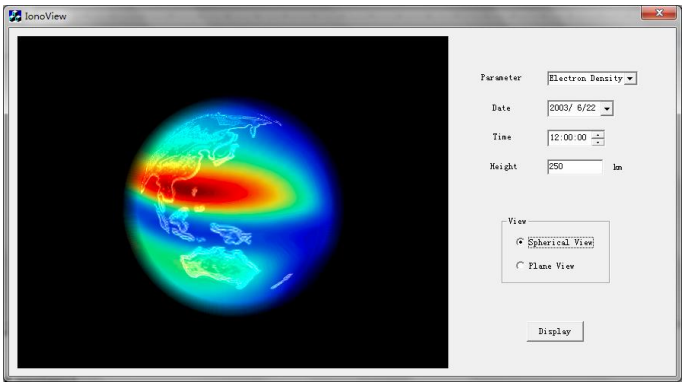

Figure 2.Software interface

2 THE TRACKBALL THEORY

What trackball method implements is to map the screen content on the three-dimensional trackball mouse surface, and to complete a rotation transformation. To facilitate the calculations, we first normalize the mouse coordinates and mapped the range of coordinates to [-1, 1]

MousePt.X = ((MousePt.X/((Width-1)/2))-1); (1)

MousePt.Y = -((MousePt.Y/((Height-1)/2))-1); (2)

We then consider the location of trackball and mapping of mouse position. The coordinates of the mouse from the origin to the point first calculate was calculated first. If it is greater than the trackball borders, its z-coordinate is 0; If the mouse coordinates within the distance from the origin to the boundary, the z-coordinate of the point is mapped to the z-axis coordinate value of the sphere. The equation is as follows

$$z=\begin{cases}\sqrt{1-x^2-y^2}, x^2+y^2\le 1\\ 0 \quad ,other\end{cases} \tag{3}$$

For any point (x,y) on the plane mouse, the surface on the trackball has a corresponding point (x,y,z). The position and location of the mouse trackball and the mapping is reversible. Therefore, with the mouse position and trackball position, three-dimensional information of the trackball can be calculated and the trackball position can be followed. Assume there are two positions in the hemispherical surface p_1 and p_2. Two vectors from the origin determine a plane (Figure 3). Normal vector of the plane is given by the cross product of these two vectors

$$n = p_1 \times p_2 \tag{4}$$

The motion of Trackball from position to position p_1 to p_2 can be implemented by a rotation around n. Rotation angle equals to the angle between the vector p_1 and p_2, which can be obtained by cross product norm of two vectors. Because the length of p_1 and p_2 is 1, respectively, $|\sin\theta| = |n|$.

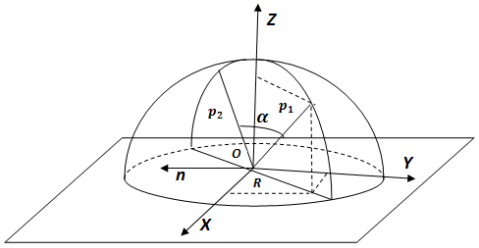

Figure 3.Plane of rotation

Homogeneous coordinate transformation matrix and the three-dimensional model for a variety of geometric transformation has been widely used. As we need to rotate the model around center of the sphere, which means the fixed point is at the origin. The rotation axis will pass through the origin to simplify the problem. In the rotation of a three-dimensional graphics, any rotation can be synthesized by three rotations around three axes to synthesis. Therefore, n and z-axis can coincide through two rotations, and then rotate around the z-axis by angle θ. Thereafter the effect of two rotations which make n and z-axis coincide offsets. The rotation matrix is expressed in the following form

$$R = R_x(-\theta_x)R_y(-\theta_y)R_z(\theta)R_y(\theta_y)R_x(\theta_x) \tag{5}$$

We start with n components because n is a unit vector, thus

$$\alpha_x^2+\alpha_y^2+\alpha_z^2=1 \tag{6}$$

Where α_x, α_y, α_z are component vectors of n on the x-axis, y-axis and z-axis. Three perpendicular linesare drawn through the point $(\alpha_x,\alpha_y,\alpha_z)$ for the (Figure 4). The angles of n and the three axis, Φ_x,Φ_y,Φ_z are the direction angle.

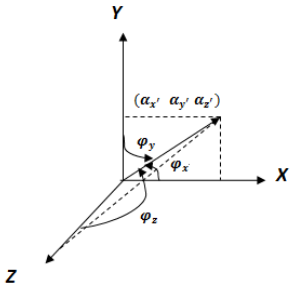

Figure 4.Direction angle

The purpose of rotating around x-axis by angle θ_x is to make vector n located in plane y=0 (Figure 5). Based on the geometric relationship, we can find $\cos\theta_x = \frac{\alpha_z}{\sqrt{\alpha_y^2+\alpha_z^2}}, \sin\theta_x = \frac{\alpha_y}{\sqrt{\alpha_y^2+\alpha_z^2}}$. Thus, the transformation matrix of rotating around x-axis by angle clockwise is

$$R_x(\theta_x) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{\alpha_z}{\sqrt{\alpha_y^2+\alpha_z^2}} & -\frac{\alpha_y}{\sqrt{\alpha_y^2+\alpha_z^2}} & 0 \\ 0 & \frac{\alpha_y}{\sqrt{\alpha_y^2+\alpha_z^2}} & \frac{\alpha_z}{\sqrt{\alpha_y^2+\alpha_z^2}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

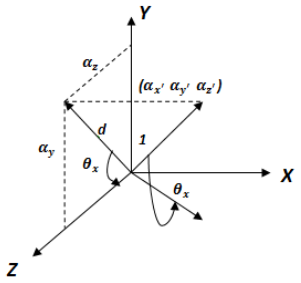

Figure 5.Calculation of

Similarly, the transformation matrix of rotating around y-axis by angle is

$$R_x(\theta_x) = \begin{bmatrix} \sqrt{\alpha_y^2+\alpha_z^2} & 0 & -\alpha_x & 0 \\ 0 & 1 & 0 & 0 \\ \alpha_x & 0 & \sqrt{\alpha_y^2+\alpha_z^2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

And the transformation matrix of rotating around z-axis by angle i

$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9)$$

From above we can get the rotation matrix as shown in Formula 5. A rotation whose fixed point is origin can be described not only by matrix, but also by quaternion. Quaternion method requires fewer number of operations, and its speed is much more faster.

Suppose origin is fixed point, and we rotate around z-axis by angle θ, unit vector $n=(0,0,1)$, quaternion r is

$$r = \cos\frac{\theta}{2} + \sin\frac{\theta}{2}(0,0,1) \quad (10)$$

After rotation, the corresponding quaternion of any point $p=(x,y,z)$ is

$$p' = rpr^{-1} = r(0,p)r^{-1} = (0,p') \quad (11)$$

Where

$$p' = (x\cos\theta - y\sin\theta, x\sin\theta + y\cos\theta, z) \quad (12)$$

The rotation matrix shown in Formula 5 can be described by arithmetic product of quaternion. Result is shown below:

$$R = \begin{bmatrix} 1-2\sin^2\frac{\theta}{2}(\alpha_y^2+\alpha_z^2) & 2\sin^2\frac{\theta}{2}\alpha_x\alpha_y - 2\cos\frac{\theta}{2}\sin\frac{\theta}{2}\alpha_z & 2\sin^2\frac{\theta}{2}\alpha_x\alpha_z + 2\cos\frac{\theta}{2}\sin\frac{\theta}{2}\alpha_y & 0 \\ 2\sin^2\frac{\theta}{2}\alpha_x\alpha_y + 2\cos\frac{\theta}{2}\sin\frac{\theta}{2}\alpha_z & 1-2\sin^2\frac{\theta}{2}(\alpha_x^2+\alpha_z^2) & 2\sin^2\frac{\theta}{2}\alpha_y\alpha_z - 2\cos\frac{\theta}{2}\sin\frac{\theta}{2}\alpha_x & 0 \\ 2\sin^2\frac{\theta}{2}\alpha_x\alpha_z - 2\cos\frac{\theta}{2}\sin\frac{\theta}{2}\alpha_y & 2\sin^2\frac{\theta}{2}\alpha_y\alpha_z + 2\cos\frac{\theta}{2}\sin\frac{\theta}{2}\alpha_x & 1-2\sin^2\frac{\theta}{2}(\alpha_x^2+\alpha_y^2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (13)$$

3 THE TRACKBALL PROGRAMMING

In the Update function, Transform is a homogeneous matrix, which is the final form of the transformation matrix, Last Rot is the transformation matrix obtained by dragging earth model last time, while This Rot is the transformation matrix obtained by dragging earth model this time. When the user clicks the mouse, to create a unit rotation matrix, while dragging the mouse, the matrix is capable of following mouse variation, we can work out the corresponding rotation information from each piece of starting point and ending point position in the trajectory. Mouse Pt is the current mouse coordinates, is Clicked and is Dragging are boolean variables which can judge whether the mouse is clicked or dragged independently. TrackBall is the object of the trackball class.

When TrackBall object is initialized, it can call the constructor to set boundaries and set StVec, EnVec to zero. In the Update function, we first check is Dragging, and if it is false, which indicates that the mouse is not dragged, afterwards, we will check is Checked state, and if it is true, which indicates that the left mouse button is pressed in the condition of not dragging, when setting is Dragging to true, and being ready to drag. And add This Rot to Last Rot, then call the click function of TrackBall to

map the coordinates of the mouse point to the 3D trajectory ball surface. When mouse is dragged, is Click will be checked, in the other words, the mouse is whether being pressed, and if the left mouse button is not pressed, is Dragging will be set to falsity. If the left mouse button has not been pressed until the mouse dragging was ending, the drag calling function will end the coordinates of the mouse point's mapped onto the 3D trajectory ball surface in the end point. And then it calculates the corresponding quarternion according to the axis of rotation vector and rotation angle, and the transformation matrix can be obtained by quarternion, the latter transformation matrix will be multiplied to the former, the result is the final form of transform matrix. Finally, glPushMatrix, glMultMatrixf and glPopMatrixfunctions will complete the transformation of operation model.

It is important to note that when clicking mouse, the initial transformation matrix is an identity matrix. When the mouse is dragged, Update renews constantly. That's why we only calculate the rotary information of the initial point of each short track trajectory to the end point in the trajectory. The rotation and transformation are applied to the 3D model expressed by ionospheric data, which will implement the rotation around a fixed point. The previous results will be accumulated to a rotation matrix to implement dynamic display of earth model. Otherwise, when the mouse is dragged again, the model display rotation from the initial state when the program is run, so that we can't implement dragging the mouse continuously.

4 RESULTS AND DISCUSSION

In the software, we set the date on June 23, 2003, the standard time, and set the time at 12:00 UT, height at 250km.We choose electron density as the parameter to be displayed, and choose spherical view to implement visualization. Then we texture map the surface of earth model with the image which is drawn with the data we obtained. The result is shown in Figure 6. Drag the mouse along the equator in this model to test trackball method, we can find that the rotation of earth model is smooth. Once the mouse position changes, Update function will update the position of trackball and rotation matrix, and Draw function will redraw the sphere model and reload textures. Through rotation we intuitively find that at the height of 250km, it shows an increasing trend in electron density from the poles to the equator on the whole, but electron density reaches peak level at latitude 20 degrees, not at equator. This phenomenon is called "equatorial anomaly".

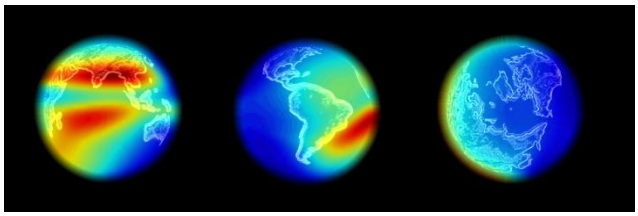

Figure 6.Rotation test on spherical model

To test the planar model, we set the date on March 22, 2003, the standard time, and set the time at 12:00 LT. Height varies from 250km to 350km. Electron density is the parameter to be displayed. Also, we texture map the surface of earth model with the image which is drawn with the data we obtained. The result is shown in Figure 7. Drag the model with mouse, we find that mouse coordinates can be mapped onto the trackball correctly to implement rotation even when it is outside the range of model. From the result we can find a peak of electron density in F_2 layer in terms of height. The peak value is about 300km.

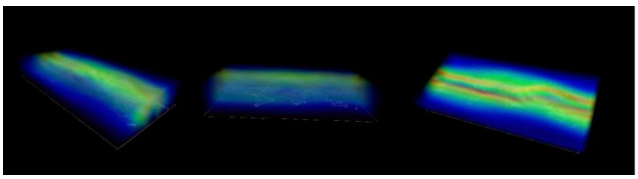

Figure 7.Rotation test on planar model

5 CONCLUSION

The visualization of the ionosphere parameters directly reveals information about characteristic parameters, density and temperature. The trackball approach described in this study improves the efficiency of the man-computer interaction and facilities operation of the users. In addition, with excellent portability and practicality, this approach can also be applied to other cases where rotation of 3D model with a fixed point is required.

REFERENCES

[1] RichardS. Wright.Jr 2012. OpenGL Super Bible (Fifth Edition). Beijing: People's Posts and Telecommunications Press

[2] DavaShreiner 2008. OpenGL Programming Guide (Seventh Edition). Beijing: Machinery Industry Press

[3] PangMingyong & Lu Zhangping2004.Mouse Tracking Ball Technology Implementation of Simple Class. Computer Engineering 30(17):182-183