

THE APPLICATION OF THE TECHNOLOGY OF 3D SATELLITE CLOUD IMAGING IN VIRTUAL REALITY SIMULATION

Xiao-fang Xie^{1*}, Yong-jun Qiao², and Tao Sun³

¹Department of Ordnance Science and Technology, Naval Aeronautical Engineering Institute, Yantai 264001, China

Email: xiexf106@yahoo.com.cn

²Graduate Students' Brigade of Naval Aeronautical Engineering Institute, Yantai 264001, China

Email: qyjdatabase@163.com

³Graduate Students' Brigade of Naval Aeronautical Engineering Institute, Yantai 264001, China

Email: big_st19800227@yahoo.com.cn

ABSTRACT

Using satellite cloud images to simulate clouds is one of the new visual simulation technologies in Virtual Reality (VR). Taking the original data of satellite cloud images as the source, this paper depicts specifically the technology of 3D satellite cloud imaging through the transforming of coordinates and projection, creating a DEM (Digital Elevation Model) of cloud imaging and 3D simulation. A Mercator projection was introduced to create a cloud image DEM, while solutions for geodetic problems were introduced to calculate distances, and the outer-trajectory science of rockets was introduced to obtain the elevation of clouds. For demonstration, we report on a computer program to simulate the 3D satellite cloud images.

Keywords: Virtual reality, Meteorological data, Satellite cloud image, Projection transformation, Cloud image DEM, 3D simulation

1 INTRODUCTION

As Xie Xiao-fang (2002) has shown, to exploit the technology of Synthetic Environment (SE) based on the real natural environment now widely employed in VR, more detailed environment data are required for visual simulation, especially for military simulation. In visual simulation nowadays, most models can be created based on real data, such as terrain, building, armament, and creatures. As far as the weather environment is concerned, most meteorological elements used in these models are unreal, a fact that can be seen in popular simulation software such as MultiGen Vega, OpenGVS, and OpenSceneGraph. The main reason for this is that the available meteorological data are too rare. In the last several years, because of CODATA and global meteorological data sharing, just with the use of modern satellite receiving equipment, we can get almost worldwide real-time meteorological data that yields real data to build a virtual natural environment. The weather elements relevant to simulation include clouds, rain, snow, wind, fog, visibility, etc. Among these, clouds are the most difficult to simulate. As the most convenient and available data source, satellite cloud images are the best for cloud simulation. Figure 1 shows the usual flow of data in the application of satellite cloud images.

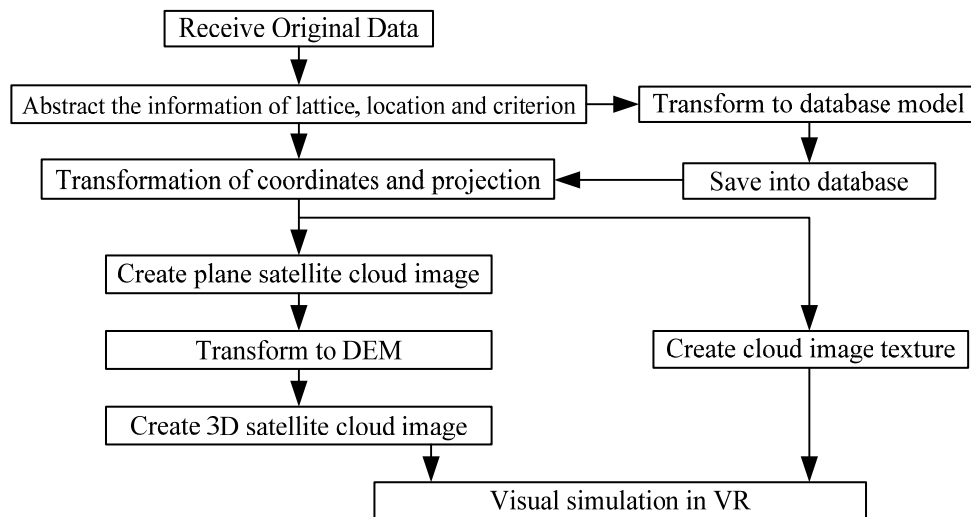


Figure 1. The usual flow of the application of satellite cloud image

2 PROJECTION TRANSFORMATION OF SATELLITE CLOUD IMAGES

The cloud image abstracted from original data is an image of the global disk, 2048×2048 in dimension. Because the disk image is an inextensible spherical surface, and the scale in different area is different, it must be projected into a square image by transformation. Projection transformation establishes the functional relationship between geographical coordinates and Cartesian coordinates. It can be formulated as

$$x = f_1(L, B) \quad y = f_2(L, B) \quad (1)$$

where (x, y) are the Cartesian coordinates and (L, B) are the geographical coordinates. Equation (1) is the general equation of projection transformation. For different projections, the expressions of f_1 and f_2 differ accordingly.

2.1 Calculating the geographic coordinates

Other than the common projection, the geographic coordinates of a point in the disk image are unknown, and we must get its geographic coordinates before the projection. Using the information of location, we can get the number of the first scan line and the pixel number of the first scan column, together with the numbers of scan lines and the numbers of pixels in 25 by 25 lattices. By virtue of these data, we can calculate the geographic coordinates and pixel coordinates of every point within a lattice. Every point that does not belong to a lattice must be in a quadrangle whose vertices are four points in the lattice. Thus, the geographic coordinates of this point can be expressed by the coordinates of these four points. Because the space between two grids is small, the longitude or latitude can be regarded as straight. In this quadrangle, take the geographic coordinates of $A_i (B_i, L_i)$, where A_i is one of clockwise four vertexes. Its pixel coordinates are (x_i, y_i) . Then the geographic coordinates (B_p, L_p) of a point in this quadrangle can be calculated from Equation (2).

$$\begin{aligned} B_p &= B_1 \times i + B_2 \times (1 - i) \\ L_p &= L_1 \times j + L_4 \times (1 - j) \end{aligned} \quad (2)$$

where i and j are the roots of the Equation (3).

$$\begin{aligned} \frac{(y - y_2) + i \times (y_2 - y_1)}{(x - x_2) + i \times (x_2 - x_1)} &= \frac{(y_3 - y_2) + i \times (y_4 - y_1 + y_2 - y_3)}{(x_3 - x_2) + i \times (x_4 - x_1 + x_2 - x_3)} \\ \frac{(y - y_4) + j \times (y_4 - y_1)}{(x - x_4) + j \times (x_4 - x_1)} &= \frac{(y_2 - y_1) + j \times (y_2 - y_1 + y_4 - y_3)}{(x_2 - x_1) + j \times (x_2 - x_1 + x_4 - x_3)} \end{aligned} \quad (3)$$

This method is faster and more precise than the method of line and row scanning.

2.2 Projection transformation

In the transformation between Cartesian coordinates and geographic coordinates, projection plays an important role. Hua Yi-xin & Wu Sheng. (2001) have shown that the common projections of satellite cloud images include Mercator cylindrical projection, Lambert conic projection, Stereographic projection, Gauss-Krüger projection, etc. In these projections, the Mercator cylindrical projection has valuable characteristics in that the grid composed of longitudes or latitudes is a rectangle after projection, which is suitable for creating cloud image DEM. Moreover, because the image projected is a rectangle, it is easy to join several cloud images into a larger composite image.

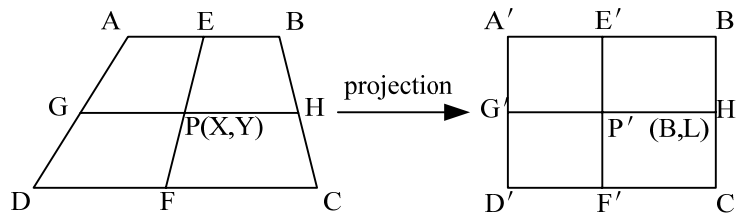


Figure 2. The principle of Mercator projection of satellite cloud image

Figure 2 illustrates the Mercator projection of satellite cloud images. After the projection, the points A, B, C, D, and P were mapped into the points A', B', C', D', and P', where A and B, C and D are in the same latitude respectively. Also, A and D, B and C are in the same longitude respectively. According to the principle of projection, the corresponding points will still lie on the same longitudes and latitudes. The relationship of point P and point P' is given by Equation (4).

$$\begin{aligned} \frac{AE}{EB} = \frac{DF}{FC} = \frac{i}{1-i} = \frac{A'E'}{E'B'} = \frac{D'F'}{F'C'} \\ \frac{AG}{GD} = \frac{BH}{HC} = \frac{j}{1-j} = \frac{A'G'}{G'D'} = \frac{B'H'}{H'C'} \end{aligned} \quad (4)$$

Projection transformation maps P to P'; the inverse transformation maps P' to P. Here, we use reverse transformation, for it assures the surjection, so we need not interpolate after the projection. Meanwhile, we can calculate variables i and j conveniently by the interpolation.

Under Visual C++, assume the variable grid is a CPoint pointer, pointing to the pixel coordinates array of a 24 by 24 lattice. Then the mapping between P(X, Y) and P' (B, L) can be expressed as follows:

```
i = 0.2 * (5 - B + (int)B / 5 * 5);
if (L > 0) {
    j = 0.2 * (L - (int)L / 5 * 5);
    if (j < 1.0E6) j = 1;
}
else {
```

```

    j = 0.2 * (5 + L - (int)L / 5 * 5);
}
X = i * j * grid[gy * 25 + gx].x + (1 - i) * j * grid[gy * 25 + gx + 1].x + i * (1 - j) * grid[gy * 25 + gx + 25].x + (1 - i - j + i * j) * grid[gy * 25 + gx + 26].x;
Y = i * j * grid[gy * 25 + gx].y + (1 - i) * j * grid[gy * 25 + gx + 1].y + i * (1 - j) * grid[gy * 25 + gx + 25].y + (1 - i - j + i * j) * grid[gy * 25 + gx + 26].y;

```

where g_x and g_y are the grid coordinates of point $P'(B, L)$. Figure 3 is one part of the satellite cloud image after a Mercator projection used with the above method.

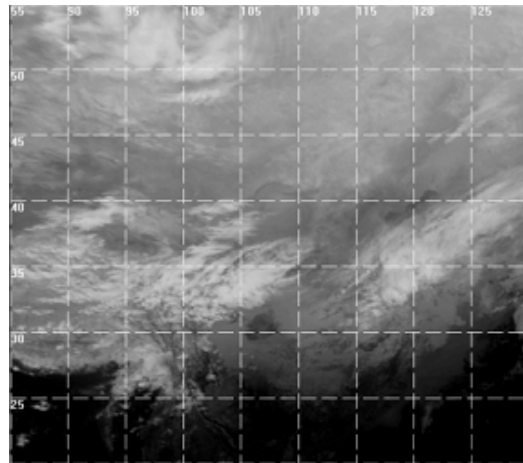


Figure 3. The satellite cloud image after a Mercator projection

3 CREATION OF A SATELLITE CLOUD IMAGE DEM

As one kind of simulation model, the Digital Elevation Model (DEM) is widely used in Geographical Information Systems (GIS) and VR. After the Mercator projection and data sampling with the sample space in conformity to resolution, we get the data grid, together with the elevation. Finally we are able to obtain the DEM of a satellite cloud image.

3.1 Distance solution of satellite cloud imaging

Because the cloud image DEM satisfies the standards of USGS DEM, (Standards for Digital Elevation Models, 1992), first we must calculate the distance of two points $A(B_1, L_1)$ and $B(B_2, L_2)$ in the satellite cloud image, which is in agreement with results obtained later. The distance can be solved by the method of solutions of geodetic problems (Kong Xiang-yuan & Mei Shi-yi, 1998).

To begin the distance solution, we should convert the parameters $B_1, L_1, B_2,$ and L_2 into radians.

$$b = B_2 - B_1$$

$$P = \sin b \cdot \cos L_2$$

$$Q = \cos L_1 \cdot \sin L_2 - \sin L_1 \cdot \cos L_2 \cdot \cos b$$

$$M = \sin L_1 \cdot \sin L_2 + \cos L_1 \cdot \cos L_2 \cdot \cos b$$

$$A_0 = \text{atan}(P / Q)$$

$$A_1 = \text{atan}(\cos L_2 \cdot \sin A_0 / \sqrt{1 - \cos L_2 \cdot \sin A_0 \cdot \cos L_2 \cdot \sin A_0})$$

$$A = \text{atan}((P \cdot \sin A_0 + Q \cdot \cos A_0) / M)$$

$$S = 6356863.020 + (10708.949 - 13.474\cos A1)\cos A1$$

$$L = A \cdot S$$

L is the distance required, and all other variables such as b and P are temporary variables. The constants are the elements of the Krasovsky ellipsoid.

3.2 Calculation of elevation

Wang Wen-jun (2002) has shown that the elevation of a cloud can be calculated from infrared and visible light information. These parameters are the bright temperature and albedo. Taking the bright temperature, for example, the satellite cloud image is composed of 256 grey levels. Through the transformation of bright temperature, every grey level can be converted into a corresponding temperature. If the location being simulated is above the cloud, we can just select a camber datum. Thus according to the temperature variation law with altitude, we can calculate the relative altitude because it expresses the elevation of every piece of cloud as relative altitude. If the viewpoint is variable, we must calculate the absolute altitude. As a precondition, we must get the corresponding temperature of the ground surface and the altitude of the bottom of the cloud, which all can be obtained from live meteorological data. With these data, we can create not only the cambered model of the cloud but also the closed-curve surface model of the body of the cloud.

The law that temperature varies with altitude is different in different areas and even for the same area; it also changes with time (Wu Yun-long & Zhang Yun-meng, 1999). To achieve higher simulation accuracy, we use here the outer-trajectory science of rockets for reference to calculate the elevation of clouds (Xi Ming-you, 1989).

Under standardized atmosphere and standardized meteorological conditions, the standard law of virtual temperature varies with altitude is

$$\tau(Z) = \tau_{on} - G_1 Z \tag{5}$$

where $\tau_{on}=288.9K$, $G_1=6.328 \times 10^{-3}K/m$ and Z is the altitude.

The transition formula of virtual temperature $\tau(Z)$ and air temperature $t(Z)$ is

$$\tau(z) = \frac{273 + t(z)}{1 - \frac{3}{8} \times \frac{a(z)}{p(z)}} \tag{6}$$

Where $p(Z)$ is the overall pressure of humid air and $a(Z)$ is the component of vapor (i.e. absolute humidity).

The meteorological conditions are different in the troposphere and stratosphere, so we need to differentiate between them.

$$\begin{cases} 0m < Z \leq 9300m, \tau(Z) = \tau_{on} - GZ \\ 9300m < Z \leq 12000m, \tau(Z) = A - B(Z - 9300) + C(Z - 9300)^2 \\ 12000m < Z \leq 30000m, \tau(Z) = 221.5K \end{cases} \tag{7}$$

In Equation (7), the constants $G=B=6.328 \times 10^{-3} K/m$, $A=230.0K$, and $C=1.172 \times 10^{-6} K/m^2$.

Equation (7) is the form of standard meteorological conditions, where virtual temperature is the standard virtual temperature. In practical application, Equation 5 should be amended according to the real ground observation.

$$\tau(Z) = \tau_{on} - G_1 Z + \tau_d \tag{8}$$

Let $Z = 0$, then $\tau_d = \tau(0) - \tau_{on}$. With $p(0)$ and $a(0)$, we can calculate $\tau(0)$. Thus, after converting the temperature of the top of cloud to virtual temperature, the inverse calculation will result in the more precise altitude of the cloud.

4 SIMULATION OF A 3D SATELLITE CLOUD IMAGE

Nowadays, there are four main methods for simulating clouds in VR visual simulation. The first and simplest method is to fill the sky with blue. The second step is BOX, i.e. mapping texture on polygon. The third one is DOME, which assumes the sky is a half sphere and maps texture onto its surface. The last method introduces a sky model with seven layers: the top thin cloud layer, the cloud layer, the bottom cloud layer, and so on. For each of these layers, according to their height and brightness, we can map the corresponding texture on it (Zhao Qin-ping, 2002).

The common characteristics of the above methods are that they all use the technology of texture mapping. The cloud simulated is not the real cloud in real time, so they cannot simulate the real cloud with a 3D structure.

Here, we introduce a DEM method to simulate the real cloud. Because the satellite cloud image is represented with a color, color rendering is better than texture mapping. To create a 3D satellite cloud image, we can use many methods such as Bezier camber and NURBS camber based on OpenGL (Peace Dove Studio, 2003).

Figure 4 is the 3D satellite cloud image created in Visual C++. The first step is curve interpolation of DEM data; the second is triangular approximation, and the last is color rendering. Having DEM, we can also convert it into another 3D model in Open Flight format with the software Multigen Creator. Thus we can use it as an object in software Multigen Vega.

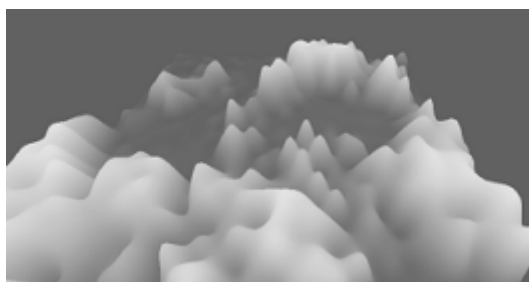


Figure 4. The 3D satellite cloud image created in Visual C++

The following C++ function is a procedure to create 3D satellite cloud imaging based on OpenGL. In this function, we put many 3-order Bezier cambers together to create the full image. Suppose the mesh after sampling is an M by N grid, and the horizontal and vertical widths of cells are $xStride$ and $yStride$ respectively, while the number of cells is $xNum$ and $yNum$. Variable order is the order of camber, and the pointer grid points to the data of the elevation of the image (Zhang Xiu-shan, 1999).

```
void Draw3DCloud()  
{  
    float x,y,z;  
    static float ctrlpoints[4][4][3];  
    for(int i=0; i<(M-1)/order; i++)  
        for(int j=0;j<(N-1)/order;j++){  
            for(int k=0;k<(N-1)/order;k++)
```

```
for(int l=0;l<=order;l++){
    x=(i*order+k)*xStride;
    z=(j*order+l)*yStride;
    y=grid[(j*oem+1)*M+i*order+k];
    ctrlpoints[k][l][0]=x;
    ctrlpoints[k][l][1]=y;
    ctrlpoints[k][l][2]=z;
}
glMap2f(GL_MAP2_VERTEX_3,0,1,3,order+1,0,1,12,order+1,&ctrlpoints[0][0][0]);
glEvalMesh2(GL_FILL,0,xNum,0,yNum);
}
}
```

5 CONCLUSION

The method introduced above has achieved good visual simulation effects, which is of important applicable value in military VR and has been applied in the pre-research project of the “Tenth Five-Year-Plan” of China. The data of satellite cloud imaging is very dense and timely, so it can be applied to improve simulation precision. Thus it can be a new data source for visual simulation and military application. A problem with the calculation of the elevation of cloud image DEM is the dependency on the statistical model.

6 ACKNOWLEDGEMENTS

This work was supported by the Military Grand Laboratory Foundation of China under Grant No.2110-11-201. We thank Prof. Liu Ai-dong and Dr. Wang Chang-jin for their comments and suggestions.

7 REFERENCES

- Hua Yi-xin & Wu Sheng. (2001) *The principle and technology of Geographical Information System*. Beijing: Liberation Army Publishing House.
- Kong Xiang-yuan & Mei Shi-yi. (1998) *Control Surveying*. Wuhan: Wuhan Technical University of Surveying & Mapping Press.
- Peace Dove Studio. (2003) *OpenGL high-level programming and the development of visual system*. Beijing: China WaterPower Press.
- Standards for Digital Elevation Models*. (1992) U.S. Department of the Interior U.S. Geological Survey National Mapping Division.
- Wang Wen-jun. (2002) The second developing of WT-8D Geostationary Meteorological Satellite Data Processing System. *Military meteorology* (1), pp 39.
- Wu Yun-long & Zhang Yun-meng. (1999) *The theory of accuracy parameter of self-propelled antiaircraft gun weapon system*. Beijing: The Publishing House of National Defence Industry.
- Xi Ming-you. (1989) *Outer- trajectory science of rocket*. Beijing: The Publishing House of Ordnance Industry.

Xie Xiao-fang. (2002) *The technology of virtual reality and its application*. Beijing: Haichao Publishing House.

Zhao Qin-ping. (2002) *DVENET distributed virtual environment*. Beijing: Science Press.

Zhang Xiu-shan. (1999) *The technology of virtual reality and the technique of programming*. Beijing: University of National Defence Technology Press.