The astronaut’s flight virtual simulation outboard of International Space Station

Hongwei Gao\(^1,\,2,\,a\), Dong Li\(^1,\,b\) and Jinguo Liu\(^2,\,c\)

\(^1\)School of Information Science and Engineering, Shenyang Ligong University, Shenyang, 110159, China
\(^2\)State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China

\(^a\)ghw1978@sohu.com, \(^b\)lidong19870622@126.com, \(^c\)liujinguo@sia.cn

Key words: Model of astronaut; VC++; OpenGL; 3-dimensional simulation

Abstract. This paper studied at the problem of visual simulation of astronaut fling outside in the cabin. Firstly, according to morphological characteristics of the human’s body, we used SolidWorks to construct the simplified geometric model of astronaut. Then we made the model of astronaut carried out the 3-dimensional real-time flight animation based on OpenGL by view transformation and model transformation in VC++ environment.

Introduction

With the launch for series of Chang’e satellite and the following lunar probe project, the research related to aerospace engineering has been caught attention widely. In these aerospace engineerings, the extravehicular activity for astronaut is a very important mission. As we all know, the aerospace engineering is often high risk, high cost and high challenge, so the 3D visual simulation based on virtual reality is very necessary in order to guarantee the safety of extravehicular activity for astronaut and the successful completion of scheduled tasks.

The completion of motion simulation is researched by use of OpenGL technique in Windows environment through designing and analyzing on flight simulation system for astronaut in outboard. The space station and human model are difficult to be drawn by OpenGL directly due to their complexity. In this paper, the space station and human model are modeled by SolidWorks software, and complete simulation with Visual C++ as a programming language. Visual C++ owns perfect MFC and AppWizard, so it is easy to call OpenGL, and it also has helpful to realize flight path planning algorithm for astronaut on the other hand.

Astronaut’s Geometry Model

Geometry modeling describes the shape (polygonal, triangle and summits) of objects and their surface (texture, surface reflection coefficient and color) of virtual object. The accuracy of modeling will influence that if the real model could complete operating duty and its result. OpenGL could combine with Visual C++ software which is OOP (Object–Oriented Programming), the perfect MFC and API of could be used for developing program. But it also has limitation: it only provides modeling function of basic geometric element (point, line, triangle and polygon), so the modeling for complex model is difficult more or less\(^{[1]}\). SolidWorks is a powerful modeling tool, and it can provide complex and accurate 3D geometry physical model. But the tool still has many inadequacies in render and program portability. For the above reasons, SolidWorks is used to build 3D model for astronaut and his motion environment, and these models are provided for OpenGL to render graph so that the difficulty of modeling is decreased and the generality of system is increased, however, it also realized vivid astronaut’s 3D simulation model. The structure diagram of system is shown as figure 1.
### SolidWorks Modeling and Data Transformation.

Firstly, the independent motion parts (head, torso, arms and legs) of astronaut are modeled accurately by use of SolidWorks: Each part has its own geometry model coordinate and geometry motion coordinate, and at the same time the relation between coordinates are recorded. Then, the SLDASM file, generated by SolidWorks, is opened by Deep Exploration which is 3D transformation software, and the file is transformed into data structure file who record the geometry information and physical information of the model such as CPP file could be embedded in VC directly to be operated, therefore the method increased reading speed largely compared with reading SLDASM and 3ds file in the program[2].

### Set Display Parameter.

In the process of using OpenGL to draw graphics, the display parameters need to be set which includes: the method of projection, time control, the method of texture mapping, the method of using buffer zone, the size of window, the position of illuminant, the reflection parameter of material, the method of processing light and shade, and the color of background and so on.

### Draw Model.

At first, the information of model is read; Then, the display list is set for each independent part (All the data are stored whatever it is graphic or pixel), triangular mesh are drew to approximate model’s curve in the list.

Firstly, the display list is set by use of glNewList (POLYGON_LIST, GL_COMPLE), and triangular mesh is drew by using glBegin (GL_TRIANGLES). If \( j=0 \); \( j<3 \); \( j++ \), we define the three points array respectively as follows:

\[
vi=face\_indicies[i][j]
\]

\[
ni=face\_indicies[i][j+3]
\]

\[
ti=face\_indicies[i][j+6]
\]

and then use glNormal3f (normals[ni][0], normals[ni][1], normals[ni][2]); glVertex3f (vertices[vi][0], vertices[vi][1], vertices[vi][2]) to represent geometry information of the model.

In order to avoid repeated drawing, we put the graphic of object model into display list, and the display list for every part is set so that the whole list of object model is completed. OpenGL function and glCallList (POLYGON_LIST) are called when the graphic is drawn; therefore, the whole 3D physical model is drawn finally.
The View Transformation of Scene

In order to simulate astronaut activity from multi-view, transformation of scene view is necessary. All of key points about computer graphic are to set 2D image of 3D object (it must be 2D, because it will display on plane screen). However, we must consider it in the way of 3D when we decide how to draw on screen. The model should be build from 3D space in inside a computer, and the pixels to be drew are decided by computer[3].

To transform 3D coordinate into pixel coordinate on screen need complete following steps:

Transformation: It is represented by matrix multiplication which includes: model, view and projection operation. These operations also include: rotation, moving, scale, reflect, orthographic projection and perspective projection. In general, several transformations should be used in combination.

Cut: Because the scene is rendered within a rectangular window, the object beyond the window must be cut. In 3D computer graphic, the cut is just to discard the object beyond cut plane.

Finally, the pixels in transformed coordinate and screen will establish the corresponding relationship.

The procedure is called the view transformation of scene view, it usually include: the view transformation, the model transformation, perspective transformation, view port transformation.

The View Transformation.
The view transformation is equivalent to fix the camera on tripod and let camera aim at scene. Before the view transformation, we need use glLoadIdentity () function to set the current matrix as unit matrix. If the current matrix has not been cleared by loading unit matrix, the processed matrix is the result of combination current matrix with last transformation. In general, the view transformation is appointed by gluLookAt () after the matrix being initialized. The parameters of the function represent the position, aim and upturned direction of camera (or eye).

The Projective Transformation.
The appointed projective transformation is just like choosing lens for camera. The transformation could be seemed as confirming view so as to ensure which objects locate in the view and their visibility. For example, the operation is equivalent to choose wide-angle lens, normal lens or telephoto lens for camera. Besides considering the view, the projective transformation has also confirmed how the object projects on the screen. OpenGL provides two basic projection methods: One kind of method is perspective projection, it is similar to the scene that we all see in everyday life; the other method is called orthographic projection, it makes the object project on the screen directly however it would not influence their relative size. Just pay attention, the current projection matrix must be initialized by glLoadIdentity () function. Thus, just appointed projection transformation could do the deed.

The Model Transformation.
The purpose of model transformation is to set the position and direction of the model. For example, the model could be rotated, translated and scaled as well combined all these operation. Such as the model transformation function for glScalef (1.0, 2.0, 1.0), its parameters have appointed how the object scaled in the three axes. If a corner of one cube locates in (2.0, 2.0, 2.0), so it will locate in (2.0, 4.0, 2.0) after being scaled. And if all of parameters are set as 1.0, the function don’t do any deed. The rotation function of glRotate () is a result that the current matrix multiplied seemed rotation object (or object’s local coordinate system), and rotate around the line from origin to (x, y, z) in counterclockwise.

It is supposed that the astronaut rotate $\alpha$, $\beta$ and $\gamma$ degree around X, Y, Z axis respectively. It is also saying that the initial coordinate B is coincident with coordinate A, then B rotate $\gamma$, $\beta$ and $\alpha$ degree around $X_A$, $Y_A$ and $Z_A$ respectively. Each of the three rotation is referenced to the fixed coordinate A, so we can get corresponding choosing matrix as follows according to the principle of “from right to left”: 
\[
\begin{align*}
\mathbf{\hat{b}} & R_{xyz}(g,b,a) = R(Z,a)R(Y,b)R(X,g) \\
\mathbf{\hat{b}} & R_{xyz}(\gamma,\beta,\alpha) = 
\begin{bmatrix}
\cos \alpha & -\sin \alpha & 0 \\
\sin \alpha & \cos \alpha & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
\cos \beta & 0 & \sin \beta \\
0 & 1 & 0 \\
-\sin \beta & 0 & \cos \beta
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 \\
0 & \cos \gamma & -\sin \gamma \\
0 & \sin \gamma & \cos \gamma
\end{bmatrix}
\end{align*}
\] (1)

Multiply them:

\[
\begin{align*}
\mathbf{\hat{b}} & R_{xyz}(\gamma,\beta,\alpha) = 
\begin{bmatrix}
\cos \alpha \cos \beta - \sin \alpha \sin \beta \sin \gamma - \sin \alpha \cos \gamma & \cos \alpha \sin \beta \cos \gamma + \sin \alpha \sin \gamma \\
\sin \alpha \cos \beta & \sin \alpha \sin \beta \sin \gamma + \cos \alpha \cos \gamma & \sin \alpha \sin \beta \cos \gamma - \cos \alpha \sin \gamma \\
-\sin \beta & \cos \beta \sin \gamma & \cos \beta \cos \gamma
\end{bmatrix}
\end{align*}
\] (2)

**Simulation Results**

Animation is an important component of computer graphic, and the motion animation is got by shooting a lot of images, then the images are projected by 24 frames a second to achieve the whole motion animation. Every frame will move to one position behind lens and the shutter is opened soon so that the frame has been displayed. The moment that the lens is switched to the next frame, the shutter is closed and then the next frame is opened, and the process is taken as a whole. Although the pictures we have seen consist of 24 frame a second, the brain will mix them into an smooth animation\[^4\].

Within OpenGL, the 2D and 3D object being described as frame in buffer is most important. These objects are comprised of a series of peaks which describe the objects’ geometry properties, or some pixels that used to describe pictures. OpenGL could execute a series of operation so that the data will be transformed into pixels data and formed in frame buffer finally. The frame buffer of OpenGL includes: color buffer, depth buffer, accumulation buffer and stencil buffer. Most of OpenGL provide double buffering which offer two whole color buffers. If one of the buffers is displayed, the other is being drawn. And when a certain frame has been drawn over, the two buffers will exchange each other. Thus, the buffer, just be used to display, is used to draw graphic at present, and vice versa.

The structure of real time animation program has little difference with above we discussed, which is achieved through redraw and exchange. In general, each frame is judged which part need to be redraw in buffer, however redrawing the whole buffer is more easy. In most of animation, the object is redraw in scene according simple and difference transformation such as the viewpoint of moving and rotation from astronaut.

**Summary**

In the paper, 3D real time simulation system for extravehicular astronaut is designed. The system is developed by using VC++ and calling OpenGL graphic library in windows environment. The 3D animation of real time simulation is generated through a series of switching scene for the model after it model being imported. OpenGL could realize vivid simulation result which makes it a powerful tool in robot simulation, and its application might contribute to the solution the problem of simulation for aerospace engineering.
Acknowledgement

This work is supported by China Liaoning Province Educational Office fund (No.20080611).

References


Information Technology for Manufacturing Systems II
10.4028/www.scientific.net/AMM.58-60

The Astronaut’s Flight Virtual Simulation Outboard of International Space Station
10.4028/www.scientific.net/AMM.58-60.938