

A Rendering Engine Development for Civil Aviation Simulation

Jing Mu

ABSTRACT

If the application software systems want to process terrain by high frame, terrain visualization is a important and spinous problem, such as the simulation of flight and synchronous sensor about ground flight test, the visualization project priority researched in this article. As far as the technology we have mastered is concerned, the main problem is the space model and coordinate transformation of visualization, which needs sufficient efficiency arithmetic and procedure to come true. Therefore, this article provides a arithmetic model and implementation to establish space construction and realize coordination transformation, which makes sure that the error rate of frame transmission is under control, and the transmission efficiency is improved.¹

KEY WORDS

Rendering Engine, Civil Aviation Simulation, Opengl

INTRODUCTION

Currently, graphic processing field focuses on 3D rendering technology injecting the technology of photo-gram metric and laser scanning. Under the support of 3D engine, most developed into an even higher sky so that airports in most medium and small cities except first tier cities set foot in digital terrain scheme. Because of that, the demand for 3D effect becomes higher and higher, therefore, our digital scheme

¹Jing Mu, Chengdu Technician College

of airport terminal not only need multitude data regarding as theoretical pillars, but also need special handling for detail.

SUMMARY ABOUT OPENGL

OpenGL

OpenGL(open graphic library) is essentially a graphic API procedure connector, which not only has cross-language and cross-platform character, but also is a powerful as well as easily invoking underlying graphics library, consequently, it prevails in 3D project(2D as well) and application area.

The ES Coordinate System of OpenGL

The ES coordinate system of OpenGL consist of three orthogonal vectors, that is X, Y, Z, and one original point intersected by the three axes. Due to the physics location relativity, the original coordination system α can be replaced by another one β . Therefore, we can use a top coordination system called world coordination system easily which can replace any others, and its original point, X, Y, Z axis are defined as (0,0,0), (1,0,0), (0,1,0), (0,0,1).

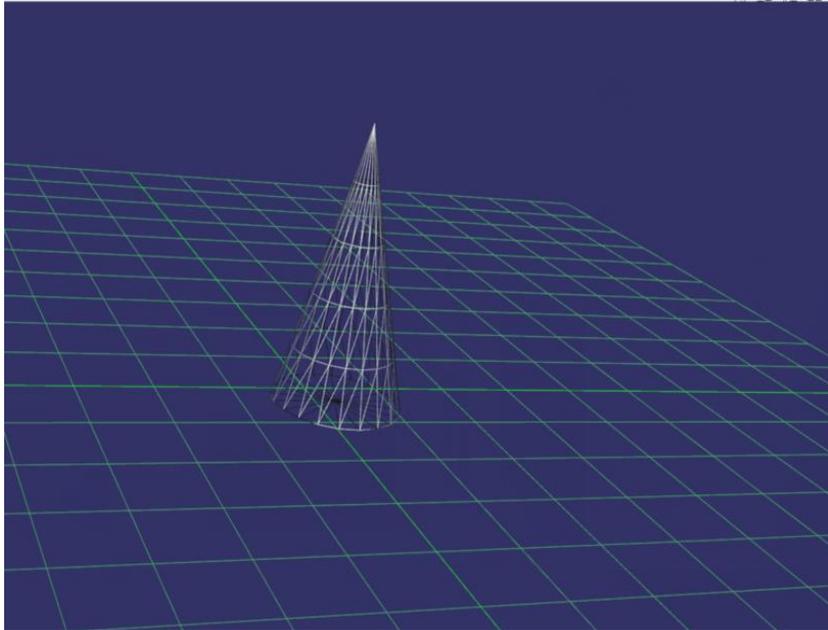
PyOpenGL

PyOpenGL is a API of muti-platform OpenGL implemented by Python. In order to study Python and PyOpenGL, this article will use Python not net version IronPython. There are two libraries connecting with PyOpenGLlibrary, that is Nump and Py Imaging Library(PIL). The key code of PyOpenGL rendering are as following:

```
#3D.py
import math
from OpenGL.GL import *
fromOpenGL.arrays import vbo
fromOpenGL.GLU import *
fromOpenGL.GLUT import *
#import OpenGL.GLUT as glut
importnumpy as ny
#Python Imaging Library (PIL)
class common:
    bCreate = False

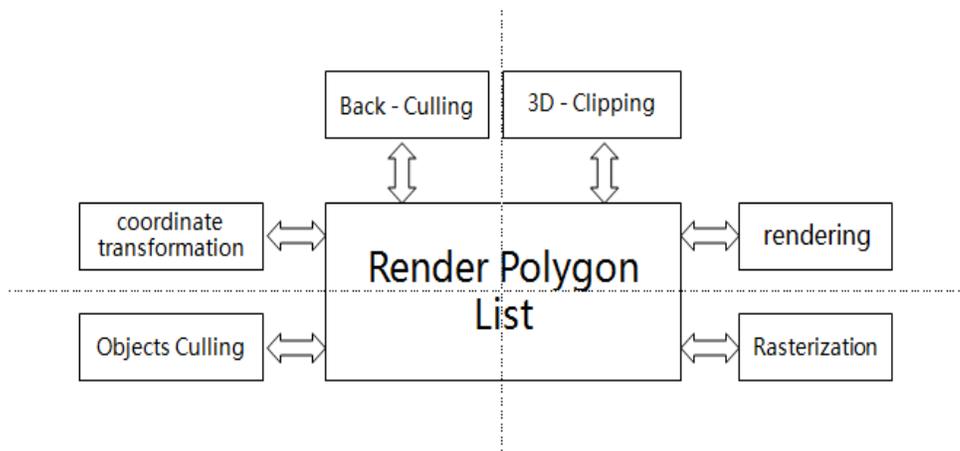
#3D implement
classCone(common):
    def __init__(this,radius,segments, rigns):
```

```
this.rigns = rigns
this.segments = segments
this.radius = radius
defcreateVAO(this):
vdata = []
Effect picture of code:
```



DEAIGNATION OF RENDER ENGINE

In this article, the 3D coordination transformation and implementation model of arithmetic are the key to render engine. By the project, we can show the geographic graphic in vision termination, and concrete framework as follows:



- 1) Coordination transformation model
- 2) Object and hiding-side eliminating
- 3) 3D clipping
- 4) Render and rasterization

IMPLEMENTATION FOR RENDER ENGINE

Implementation for Coordination

1. Transformation from local coordinates to global coordinates

There are two function regarded as the key point to this procedure:

- 1) The function of translation matrix structure.
- 2) The function of transformation from local coordination to global coordination.

We can let all of the diagonal points in the handled objects to multiply the structured translation matrix in the first step in order to operate coordination. After the first step and the second step, the global coordination which we need can be implemented by original local one. At last, the result can be reserved in physical device, and just in case, we should backup the data.

2. Transformation from global coordination to camera coordination

By translation and rotation, the transformation between two coordination can be implemented and concrete operation as follows:

1) We assume the camera's global coordination as $cam_pos(cam_x, cam_y, cam_z)$, then using inverse matrix to translate all the objects, vertexes and polygons in the global.

2) In the Euler model, there are three angles used to show the orientation of camera. And each of them represents degree to which the axis is rotated clockwise or counterclockwise. And their translation matrix is R_x, R_y, R_z .

3. Translation from camera coordination to perspective coordination.

An important goal of perspective transformation is to map the vertex point of the object to the interface of the image. Assuming that the distance between the two is D , then we can obtain the position of junction between any point and the visual point of the operation object by geometrical operation.

4. Translation from perspective coordination to interface coordination.

Here we need to use a standardized library function to complete the transformation from ordinary coordination to global video terminals.

Object Elimination

There is a principle that needs to be introduced here called the guard ball test. In the theory design, we envision for all objects in the three-dimensional world will be wrapped by a sphere, then its center coordination transformation can be implemented in the world coordinates and relative coordinates conversion method. At this point, we have obtained the basis of interpretation, and through this conclusion, the package that is not in the scope can be discarded, while in the scope, it can be reserved for further testing.

Hiding-Side Elimination

Operating principle of hiding-side test: we can use consistent methods, including clockwise and anticlockwise way, to mark all image objects, then the face vector can be obtained by calculation. If the angle between N and V vector meet the demand, then this image object is satisfying.

3D Object Space Cut

When we finish all the previous woke, the final step is to cut the spatial pattern of the image object. Here we will use the clip()function of render engine to eliminate the image object's redundant pats, remarkably, which may be not enough but significant. And then, the result can be reserved in RAM or solid state disk.

Render and Rasterization

There are many various points in one polygon, which are the basic points we need to render and rasterize. The concrete operation is to write the basic pints' pixel number into corresponding RAM in the visual terminal. At the same time, before rasterization, the basic points should be illuminated, color matching and texture mapping. And the datum will be reserved in render table tentatively when all of works are finished.

CONCLUSIONS

The aim of this article is to study the core of airport terminal model render. Although the method provided is not precise, it's very fast. And using this method, you can use the image rendering software model very smoothly on the current PC. If the method can be used, the image quality of airport terminal digital rendering model will be raised substantially.

REFERENCES

1. OpenGL Programming Guide. Dave Shreiner. 2010.
2. The Design and Implementation of 3DImage. University of Electronic Science and Technology of China ,Academic Dissertation of Hao Liuin 2007.
3. Digital Image Processing [M]. Electronic Industry Press, Ruanqiuqi, 2001.
4. Research on Satellite Remote Sensing Image Processing System Based on OPENGL and GDAL [D]. XingyuChen. South China University of Technology 2013.
5. OpenCL Language and Compilation Technology Analysis [J]. Shenli, Wangshanshan. Information System Engineering. 2012(09).
6. Research on Several Key Algorithms in Image Processing [D]. Kang Mu. Xi'an University of Electronic Technology 2009.