



2012 International Conference on Applied Physics and Industrial Engineering

Research on Dam Simulation System Based on OpenGL

Kairong Li, Juanni Wang, Kui Zuo, Jian Yun

*Information Engineering School,
Yangzhou University,
Yangzhou, China*

Abstract

Dam, not only has great ability to alleviate water shortages especially during the dry season which improve hydropower capacity, but also plays an important rule in the river ecosystem. Dam has significant effect in improving our water environment and protecting ecological system. This paper copes with the three-dimensional model of dams and discusses OpenGL modeling, lighting, material, model transformation, perspective transformation technologies to achieve visualization of the dam, and eventually we visualize the dam through concrete examples.

© 2011 Published by Elsevier B.V. Selection and/or peer-review under responsibility of ICAPIE Organization Committee.

Open access under [CC BY-NC-ND license](https://creativecommons.org/licenses/by-nc-nd/4.0/).

Keywords: OpenGL; Dam Modeling; Material; Geometric Primitives; Perspective Transform

1. Introduction

It has been thousands of years for human to conserve water resources by dam. By the end of 20 century, there have been 240 countries in which about 75 countries control flood mainly rely on dams, and 90% of the electricity comes from hydropower. No doubt, dam construction, hydroelectric power have become a significant part of social civilization. Dam construction could not play a single role, except for flood control, irrigation, shipping, tourism, aquaculture, etc., the most important respects are the water supply and power generation. It is one of the important roles for modern dams to control flood and then turn it into available water resources. In particular, the contradiction of water resource in modern society is very conspicuous, therefore dam construction is quite critical in solving water problems. For instance in China, fresh water resources are considerably sparse. Ecological environment has been devastating as a result of natural disasters. On one hand, flood takes place frequently, on the other hand, drought is also severely. Therefore, the fundamental solution of these problems are inseparable from the construction of a number of dams to increase the regulatory capacity of natural water resources^[1]. For this reason, the research on a fast three-dimensional visual simulation of the dam is considered fairly significant.

In recent years, computer systems, the continuously development of the devices of graphics input and output and computer graphics software, which create a condition for the computer-aided design. Open Graphics Library (OpenGL) is a multi-platform, high-performance 3D graphics software development system, provides a strong design capability in two and three-dimensional graphics. There are many factors in the OpenGL for the drawing quadratic curves and surfaces, such as NURBS Curve. It is in this paper that we can visualize the dam by the technology of OpenGL.

2. Theories

2.1 The description of dam's data structure

The cross-section of the dam is one of the important parameter in geometric modeling which mainly contains arched, rectangular, trapezoidal, and inclined trapezoidal shape (Figure 1) [2]. Measuring the cross-section feature points, we will take arch-shaped cross-section as an example: the upper part of the cross-section is semi-circular, the lower part is rectangular.

The error should be controlled in the measurement during the processing, in order to make sure that the 4 points approximately in the same vertical plane, and also make sure the cross-section similarly perpendicular to the centerline of the dam on the ground.

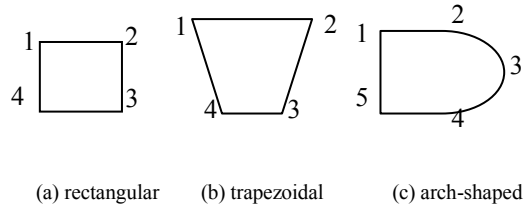


Figure 1 dam cross-section

In order to study well, we will focus on the situation of circle (Figure 2).

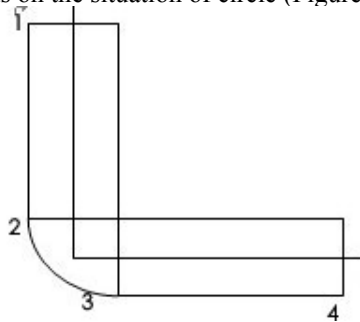


Figure 2. dam surface

2.2 OpenGL Modeling Technology

OpenGL is a graphics hardware with software interface. It is composed of a series of OpenGL commands, which achieves most of the functionality and create dynamic, interactive three-dimensional color graphics program. There are about 10 categories of basic geometric pels shown in Figure 3.

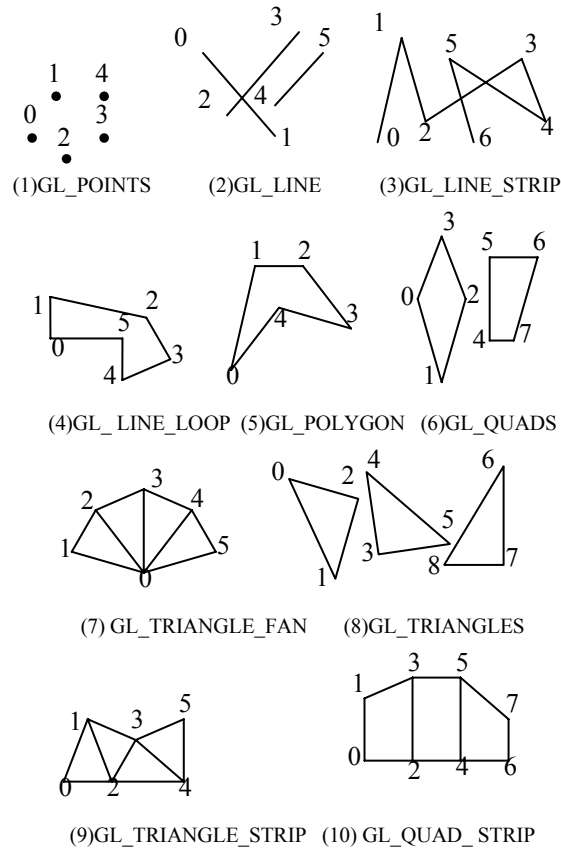


Figure 3. Geometric primitives

In the OpenGL,all the graphics are indeed composed by certain basic primitives,namely point, line, polygon which is also used mostly by triangular and rectangular, even in some complex graphics^[3]. All the geometric objects are ultimately described as a set of ordered vertices by the function of *glVertex3f()*.

2.3 Light and Material

Three-dimensional display of different objects, affected obviously by the light position, light color, material objects. Lighting effect is the result of both light resource and the physical surface.In the OpenGL,light is determined by the combination of three primary colors called red, green and blue(RGB),while the physical surface is determined by the reflected light. As the result of the lighting effects, we can calculate by segistration of the four separate parts, including the emitted light, ambient light, diffuse reflected light, specular light.

3. Application and Experiment

3.1 Structure of Dam

According to the data and graphical features of the dam, we have finished the establishment of related data structures as follows:

```
typedef struct Vertex {
    GLfloat v[3];
    GLfloat c[3];
    GLfloat u[3];
};
struct Sector {
    SectorPoint POINT1; //the points in the section
    SectorPoint POINT2;
    SectorPoint POINT3;
    SectorPoint POINT4;
};
```

3.2 Modeling of the dam

Considering the shape characteristics of dam, we have to discuss 4 kinds of submodels in the system respectively, such as rectangular at the bottom of the rectangular plane model, both sides of the rectangular; fan at the bottom of the fan-shaped plane model; cylindrical surface and rectangular model contains circular wall of the intersection of the dam, half-dome dam wall, the dome of cylindrical surface and fan model. Schematic diagram of various sub-models of the dam are all geometry united by *GL_QUAD_STRIP* whose rectangular element size should be approximately equal, and we will mix material and light in it. We must make sure each of the normal direction in sub-model should be calculated carefully, that on a circle should toward the center of the circle, and that on the plane should have the same direction. We can add normal direction through the function of *glNormal3fv()*, and normalized by function of *glEnable(GL_NORMALIZE)*, otherwise the same light and materials shown different intensity.

3.3 Application of light and material

Although the OpenGL lighting model is an approximation, it is very useful and computing speedy.

The color of material is realized by the function of *glMaterialfv()*. Its reflected light in the red, green and blue color the percentage of decisions, in the approximation that OpenGL through the use of function implementation. How light is reflected is determined by the material properties of objects in the scene, and thus shows that what material it is. Corresponds with light, material has the independently of environmental reflection, diffuse and specular color components respectively, which determines the reflection capability.

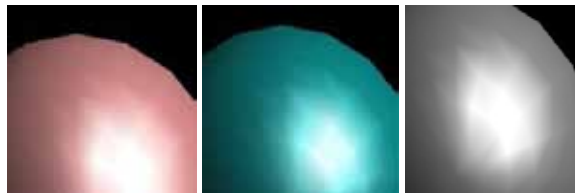


Figure 4. Reflection of light

Add lighting to the scene as the following steps:

1. The definition of the normal vector direction of vertex in all objects as the normal vector determines the orientation of light and physical surface.
2. Establish, select and place one or more of the illuminant.
3. To establish a lighting model, which determines the degree of the global environment illumination as well as the valid location view point (for lighting calculations).
4. The definition of material objects in the scene.

The code of light and material is as follow:

```
GLfloat light_position[] = {0.0,1.0,5.0,1.0};
//light resource position
GLfloat no_mat[] = {0.0,0.0,1.0,0.0};
GLfloat mat_grey_ambient[] = {0.5,0.5,0.5,0.5};
GLfloat mat_red_ambient[] = {0.0,0.0,1.0,1.0};
GLfloat mat_diffuse[] = {0.5,0.5,0.5,1.0};
GLfloat mat_specular[] = {1.0,1.0,1.0,1.0};
GLfloat no_shininess[] = {0.0};
GLfloat low_shininess[] = {4.0};
GLfloat high_shininess[] = {100.0}; //mirror index
GLfloat mat_emission[] = {0.3,0.3,0.3,0.3};
// radiation material color
glLightfv(GL_LIGHT2, GL_POSITION, light_position); glEnable(GL_DEPTH_TEST);
glDepthFunc(GL_LESS);
glEnable(GL_LIGHTING);
glEnable(GL_LIGHT0);
glShadeModel(GL_SMOOTH);
```

With the light-related function call will be in function of *myinit()*, *glEnable()* used for opening the light source, *glLight()* specify the location and direction of it. With the assuming of light parameters (RL, GL, BL), material parameters (RM, GM, BM) and without considering other factors, then the observed light parameters is $(RL \times RM, GL \times GM, BL \times BM)$ [5].

In the imitate of dam, we will color all the material in order to make it true. RGB components are respectively (R1, G1, B1) and (R2, B2, G2), then the OpenGL will be the sum of the corresponding components obtained (R1 + R2, G1 + G2, B1 + B2). If any sum is greater than 1 (indicating that the light intensity is a display device can not be achieved), was considered to be one.

3.4 Quick View of dams

Implementation of this feature would be realized by model transformation or perspective transformation.

Model transformation was carried out in the world coordinate system with the coordinate origin as a default [6]. In this coordinate system, there are three commands of OpenGL such as the model translation by the function *glTranslate (fd) (TYPE x, TYPE y, TYPE z)*; model rotation by *glRotate (fd) (TYPE angle, TYPE x, TYPE y, TYPE z)*; model scaling by *glScale (fd) (TYPE x, TYPE y, TYPE z)*.

Perspective transformation of objects determines the location and direction of viewpoint in the scene. Thus, combined the perspective transformation with model transformation is fairly meaningful.

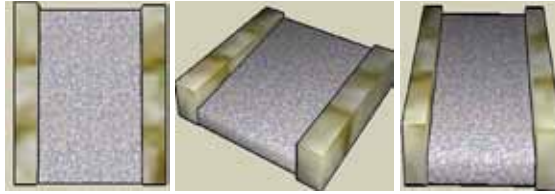


Figure 5. dam full view

This system will achieve browsing of all the processing through this function, upx, upy, upz refer to the direction of upward vector; eyex, eyey, eyez refer to the viewpoint position; centerx, centery and centerz refer to the location of reference point ; in the software will assume the eyex, eyey, eyez, centerx, centery and centerz as parameter variables in the program. By using the function of *OnTimer (UINT nIDEvent)*, we could obtain automatically the message of viewpoint change, so as to realize the dam quick browsing. The results as shown in Figure 5.

4. Conclusions and future work

OpenGL provides a good condition to achieve realistic 3D graphics rendering and display on the computer. The dam screen simulation by use of OpenGL modeling, lighting, material, blanking and other technologies obtains good results of three-dimensional realistic graphics and wholly browsing, which has an important theoretical and practical significance not only help the staff observe the basic situation easier, but also better our project management and construction services.

The dam simulation in this paper plays an important role in hydraulic, but it is just limited to the cross-section. Therefore in the future work, we hope to expand the simulation scope based on the current work to simulate the entire dam structure. In the simulation processing, if we refer to the various factors, such as wind, time, displacement and destruction and other factors, to make the research results more realistic, then there is a better research significance.

References

- [1] Implementing OpenGL ES on OpenGL Consumer Electronics, 2009, ISCE'09, IEEE 13th International Symposium..
- [2] Modern OpenGL: its design and evolution International Conference on Computer Graphics and Interactive Techniques 3 ACM SIGGRAPH ASIA 2008 courses.
- [3] 3D visualization simulation system based on OpenGL of low altitude chase/interception
- [4] Research of Creating and Fetching 3D Models of Virtual Reality Based on OpenGL Mechatronics and Automation, Proceedings of the 2006 IEEE International Conference on
- [5] Solving the embedded OpenGL puzzle - making standards, tools, and APIs work together in highly embedded and safety critical environments Digital Avionics Systems Conference, 2005. DASC 2005. The 24th
- [6] An introduction to shader-based OpenGL programming. International Conference on Computer Graphics and Interactive Techniques ACM SIGGRAPH 2009 Courses