

Application of Embedded Graphics Processing Design Based on DSP + FPGA Structure in 3D GIS

Yuanyuan He¹, Lijun Liu² and Yi Zhang^{*1}

¹Information Engineering Department, Shijiazhuang Institute of Railway Technology Shijiazhuang 050061, Hebei, China

²College of Economics and Trade, Shijiazhuang University of Economics, Shijiazhuang 050031, Hebei, China

Abstract: In this paper, the design method of OpenGL graphics processing module is analyzed, and a graphics processing method based on DSP + FPGA structure is proposed. And design a can realize the function interface standard of OpenGL graphics processing module, from the OpenGL based modeling, coordinate transformation, fire hidden technology, light illumination model and texture mapping are introduced the principle of OpenGL and 3D GIS modeling method and the three-dimensional modeling technique were to explore and research.

Keywords: 3D modeling, DSP, FPGA, GIS, OpenGL.

1. INTRODUCTION

In recent years, with the continuous development of the information age, the concept of digital earth has been deeply rooted in the hearts of the people, GIS in the digital earth's progress has played a key role in supporting technology. Spatial information visualization technology, virtual reality technology, 3D technology, image processing and artificial intelligence technology have been widely used in geographic information system. Virtual reality is a kind of real-time three-dimensional space, it for the user provides a realistic simulation of the operating environment, users seem to place oneself in the real world, through man-machine interactive operation in virtual reality 3D object, the user can query, browsing and analysis of virtual reality objects. Three dimensional GIS are very important in the fields of geology, mining, environment, ocean, city, transportation and so on. D GIS is the inevitable product of the development of the two-dimensional GIS to a certain stage. As shown in Figs. (1-3), the traditional two-dimensional GIS can only display the two-dimensional information of the plane, can't give people the true feelings of nature. Compared with the 2D GIS, 3D GIS can give a more realistic feeling to the objective world. It is through the three-dimensional modeling technology to the user to show the geographical space phenomenon, not only expressed the spatial object plane relationship, but also describes and expresses the relationship between the object. With the development of the application, the elevation information is becoming more and more important. People are also increasingly being asked to deal with the problem from the perspective of three-dimensional space [1]. Therefore, the 3D object modeling technology has become the research hotspot in recent years. To meet the real-time, effective and fast graphics processing

ability of GIS display processing system put forward higher requirements, design a meet the requirements of embedded graphics processing module becomes particularly necessary. At present, the design of embedded graphics processing module mainly has the following two ways:

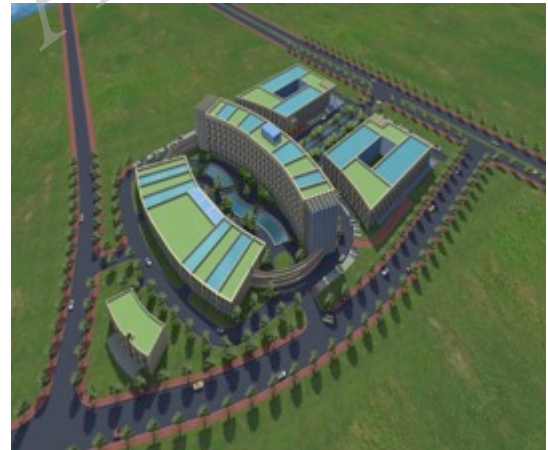


Fig. (1). Application of urban 3D model.

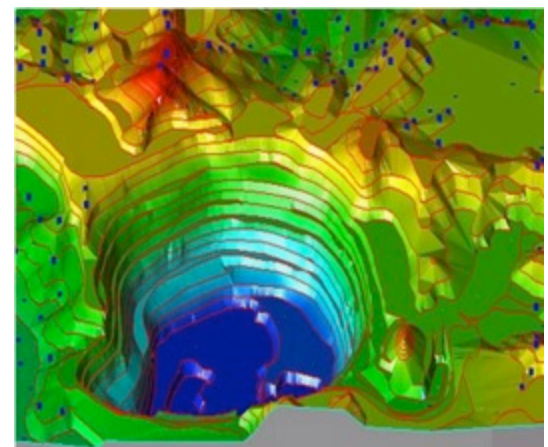


Fig. (2). Application of three dimensional model of elevation.

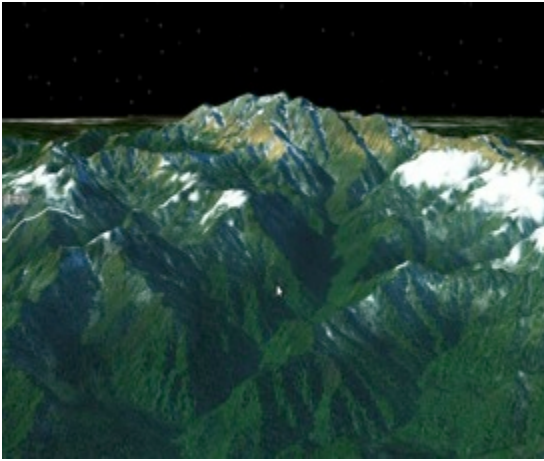


Fig. (3). Map 3D model.

1. using the special graphics processing chip to build a graphics processing platform. This is graphic design in the traditional way by control special graphics processor (GPU) to achieve ideal graphics output, this chip generally provide standard interface of OpenGL functions, in the design to reduce the development cycle. But the power consumption of the platform image, is not conducive to compact the use of low power embedded system, GPU chip internal pipeline rendering and rendering algorithm for.
2. using FPGA or DSP to build a graphics processing platform. DSP image processing platform using DSP graphical display algorithm is the advantage of easy to handle, but must choose to support graphic processing function of DSP chip, chip select a narrow range, drawing instruction parallel processing efficiency is low; using FPGA to build the platform using the features of FPGA programmable, instruction stream can be processed in parallel, but FPGA for data solution calculation and processing algorithms support is not high enough, intensive type matrix geometric processing algorithms require relatively high-end FPGA and takes high FPGA hardware resources, making graphics display processing performance can't be better to play. Not providing the OpenGL function interface standard also makes these two platforms in the embedded graphics processing system, the relatively narrow range of applications [2].

In this paper, we use the two methods and gives a kind of DSP + FPGA structure of airborne embedded graphics processing module design method, the method fully integrated with DSP and FPGA in the graphical display of the advantages, according to the needs of specific tasks, to function for heavy cutting configuration, design a specific function of the graphics processing platform, in order to reduce the processing load of the FPGA and DSP, reduce the power consumption of the platform, improve overall performance under the running environment of the graphics processing. But also has optional multi class DSP chip and the OpenGL graphics rendering pipeline and DSP + FPGA

structure combining the characteristics, making the platform not only conforms to the interface of OpenGL functions and applied more widely.

With the development of chip technology, with massively parallel processing ability of FPGA and multi DSP core, using DSP + FPGA architecture design conforms to the specific use of graphics processing module will be a trend in graphics processing.

2. THE CHARACTERISTICS AND BASIC PRINCIPLE OF OPENGL

2.1. OpenGL Features

1. High graphics quality, good performance. In the field of CAD/CAM/CAE, medical image processing, virtual reality, entertainment, advertising and so on, the developers can take advantage of the ability to use OpenGL to develop their own creativity [3].
2. Standardization-OpenGL is the only truly open, independent, cross platform graphics standard.
3. Stability-OpenGL has been used for more than 7 years on various platforms, it has a clear and well controlled specification, and has backward compatibility, so that existing applications will not fail. On any hardware compatible with OpenGL, no matter what operating system is used, the same application is shown to have the same results.
4. Scalability. Through the OpenGL extension mechanism, we can use API to expand the function.
5. Scalable. OpenGL based applications can be run on a variety of systems, ranging from home appliances to PC machines, from workstations to supercomputers. That is to say, OpenGL applications can adapt to the development of a variety of target platform.
6. Easy to use. OpenGL has good structure, intuitive design and logical command. Compared with other graphics packages, the code of OpenGL application is less. In addition, the OpenGL package of information on the basic hardware, so that developers do not need to specifically for the hard Design of a special piece.
7. Document rich. The information about the OpenGL is many, and there is a large number of available code, easy to develop the reference.

2.2 Basic Principles of OpenGL

2.2.1. Process Rather than Descriptive

OpenGL very directly specified transform matrix, light, anti-aliasing method parameters to draw 2D and 3D graphics. However, it does not provide a means for the description or modeling of complex geometric objects. Therefore, the publication of the OpenGL command is to specify how to generate a specific result, rather than the exact description of the results should be, that is, the OpenGL is a process rather than a descriptive.

2.2.2. Execution-Mode

The OpenGL command mode is the client / server mode, that is, the client (application) issued a command, and the command is interpreted and processed by the OpenGL server. The server may run on the same or different computers as the customer. Based on this, the OpenGL is transparent to the network [4].

2.2.3. Primitives and Commands

OpenGL be able to draw primitives, including points, lines and polygons. OpenGL can choose between these types of primitives, and set a pattern that does not affect other modes. OpenGL orders are always sequence processing, that is, the first definition of the elements must be finished before the execution of the following orders.

2.2.4. Rendering Method

OpenGL provides the following object in 3D rendering: wireframe rendering mode, depth priority line frame rendering and anti-aliasing wireframe rendering, flat shading mode, smooth shading mode, shading and texture, motion blur, atmospheric effects, in depth domain effect drawing, and so on [5].

2.3. OpenGL Workflow and Drawing Process

OpenGL is designed to be built on hardware, in a way that works, and the workflow is shown in Fig. (4). The input of OpenGL work flow can be image or geometric primitives, but the final result is the image of the grating. After the image is entered into the frame buffer, the hardware is displayed on the output device. All graphics objects (including geometric primitives and images) of OpenGL can be stored either in the display list (delay mode), or immediately (immediately).

For the image, OpenGL first by unpacking the pixel to the pixel format into OpenGL internal format, followed by pixel operations directly after the rasterized output or as other objects of the surface texture. For geometric primitives, all the geometric primitives in OpenGL are described by vertices. OpenGL first through the vertex of the unpacked different format of the vertex data into the internal standard format, and then operate the pair of vertices and related data (the coordinates, colors, texture coordinate, logo on the side, etc.), finally get visible image. OpenGL drawing process steps are as follows:

1. Set the pixel format - mainly including the establishment of OpenGL rendering style, color mode, color, depth, etc.;
2. Three-dimensional model is established to construct geometric elements (points, lines, polygons, images and bitmap), and a mathematical description of the object is created;
3. Stage set - an object that is placed in the appropriate position in the three-dimensional space, and the observation point of a favorable scene is selected;
4. The effect of the process of setting the material (color, optical performance and texture mapping, etc.) of the object, and adding the light and illumination conditions;
5. The mathematical description of the object and the color information of the object can be transformed into the pixel information that can be displayed on the computer.

OpenGL is designed to be built on hardware, in a way that works, and the workflow is shown in figure.

3. COMPUTER GRAPHICS PROCESSING STRUCTURE

The hardware structure of graphics system is composed of graphics processing, display control, and frame buffer as the core, and the memory circuit, interface circuit and signal output interface circuit, the schematic diagram of the structure is shown in Fig. (4).

Graphic processing and display unit is the core of the whole graphics processing hardware. The unit mainly through graphics rendering pipeline completed graphic display the desired geometry processing and raster processing operations, will be responsible for processing pixel data into the frame buffer, the final completion of ideal graphics display [6]. Graphics rendering pipeline in line with OpenGL standard as shown in Fig. (5)

4. OVERALL DESIGN OF GRAPHICS PROCESSING MODULE

4.1. Graphics Processing Module

The DSP + FPGA Architecture Design of graphics processing module hardware components as shown in Fig. (6), using PCI bus connection with external devices, support

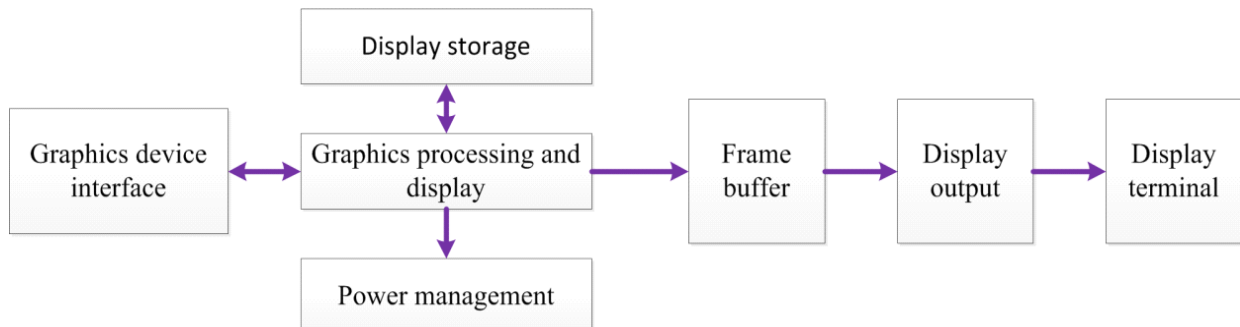


Fig. (4). Graphics system hardware structure.

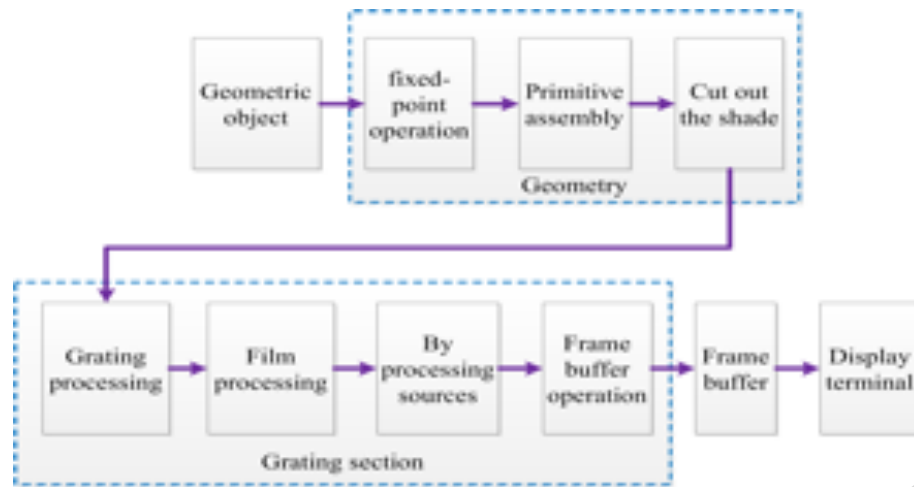


Fig. (5). Graphics system hardware structure.

2 3 line XGA video input, 1 line 3 XGA output, 1 5 lines XGA output and 1 LVDS outputs and between DSP and FPGA using EMIF bus connection, DSP and FPGA together to complete the graphic processing and OpenGL interface [7].

4.2. DSP and FPGA Functions

In the DSP + FPGA structure, the FPGA is fully implemented in parallel with the data stream and the unique advantages of DSP in the analysis of the algorithm are improved. Specific implementation, related to the geometric arithmetic steps are assigned to the DSP for processing, such

as graphics translation, scaling, primitive assembly, fire hidden judgment, matrix calculation and graph vertex position judgment *etc.*; relate to the grating and display of the step distribution to FPGA for processing, such as the basic graphic drawing, grating, the chip element operation and texture control.

5. KEY TECHNOLOGY

5.1. Realization of FPGA Logic Function

In FPGA is mainly to the completion of the PCI bus, peripheral coding and decoding chip, memory and VESA

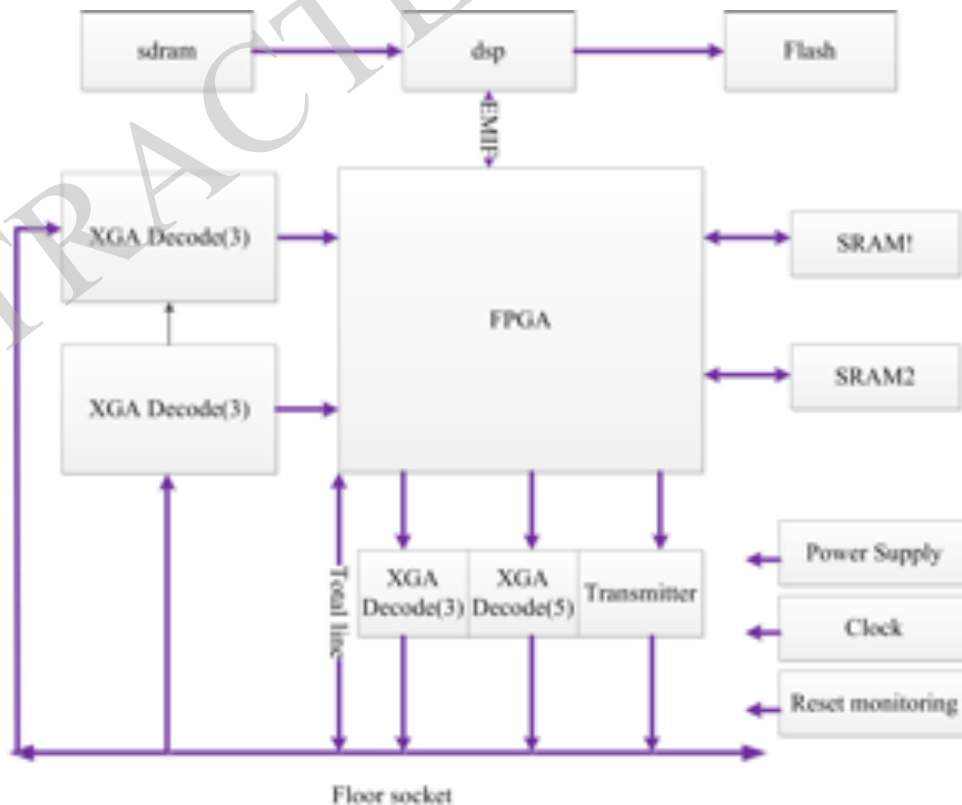


Fig. (6). Graphics processing module hardware structure diagram.

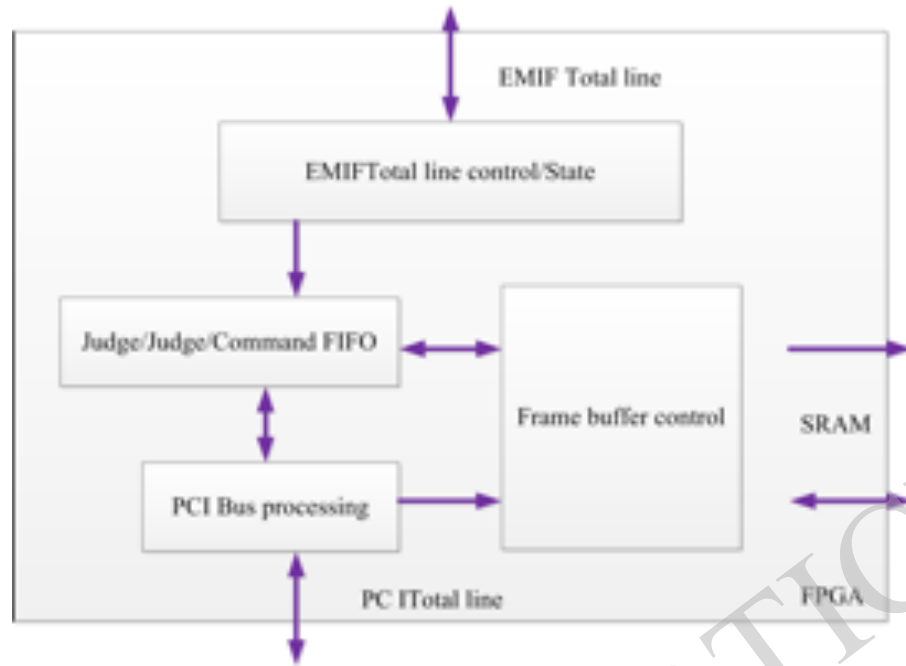


Fig. (7). FPGA internal functional structure diagram.

timing control, according to the standard of OpenGL interface of grating processing rendering requirements to complete the basic primitives such as points, lines and circles. In a field programmable gate array (FPGA) set up a shared memory area, the storage area by EMIF bus and PCI bus control, CPU stores over the data and drawing instructions and graphics processing module itself produce various kinds of status information. The schematic diagram of the structure is shown in Fig. (7).

PCI bus send in the drawing commands and other information through FIFO write shared memory area, and read the graphics processing module state information, such as drawing the end state and response state information, according to the state information generated a directive. Graphics processing module through the inquiry of the shared memory area corresponding to the location of data, analytical drawing instructions, according to the judgment criterion will need to perform in the FPGA instruction direct implementation, will be the final rendering graphics write memory, and control the frame-buffer graphics VESA timing in accordance with the standard quasi real time output.

5.2. OpenGL Library Function Implementation Method

OpenGL (Open Graphics Library) is a kind of software interface standard, which provides a very rich graphical interface function. The developer can use these functions to establish 2D/3D model and make a real time interaction. OpenGL's streamlined, independent of hardware interface features make it the industry's most widely accepted 2D/3D graphics API [8].

In this paper, Function of OpenGL core library of unique code, and settings are required in DSP for processing of the flag, the CPU end OpenGL drawing functions encapsulated into only coding in the transfer to the graphics processing module, graphics processing module receives the drawing

instructions to commands into analytical processing. In the DSP to complicated instruction do further split, and FPGA is called the basic graphic drawing and clear the screen operation command to complete the function of the whole function. For example gl-Begin function, the function with parameter mode, a total of 10 different data values, respectively, gl-Begin and gl-End between the 10 different forms of graphics through vertex rendering. Realization and the function of a unique code, set processing flag bit is 1, show that the function needs further split in DSP, graphics processing module from FPGA shared memory area read drawing instructions judgment handed by DSP is responsible to analyze the function of command, DSP in vertex operations and primitive assembly the two rendering steps were operates as follows: to pass over the vertex coordinates coordinate conversion, generates the appropriate screen coordinates of vertices, and according to the read gl-begin function parameters and the vertex of primitive assembly, generating parameters of primitive, and the primitives into a rendering step, by FPGA is responsible for drawing is completed. The instruction flow chart is shown in Fig. (8).

6. (3D) GIS MODELING TECHNOLOGY

After many years of development, 3D GIS modeling technology has been developed rapidly. Launched by the SGI open three-dimensional graphic software interface of OpenGL, to the high performance of the interactive 3D graphics modeling capability and easy programming development superiority, the favor of a large number of companies and recognition. OpenGL is independent of the window system and operating system, it is convenient to transplant between various platforms, and can be flexible and convenient to achieve two-dimensional and three-dimensional advanced graphics technology. It provides modeling, transformation, illumination processing, color

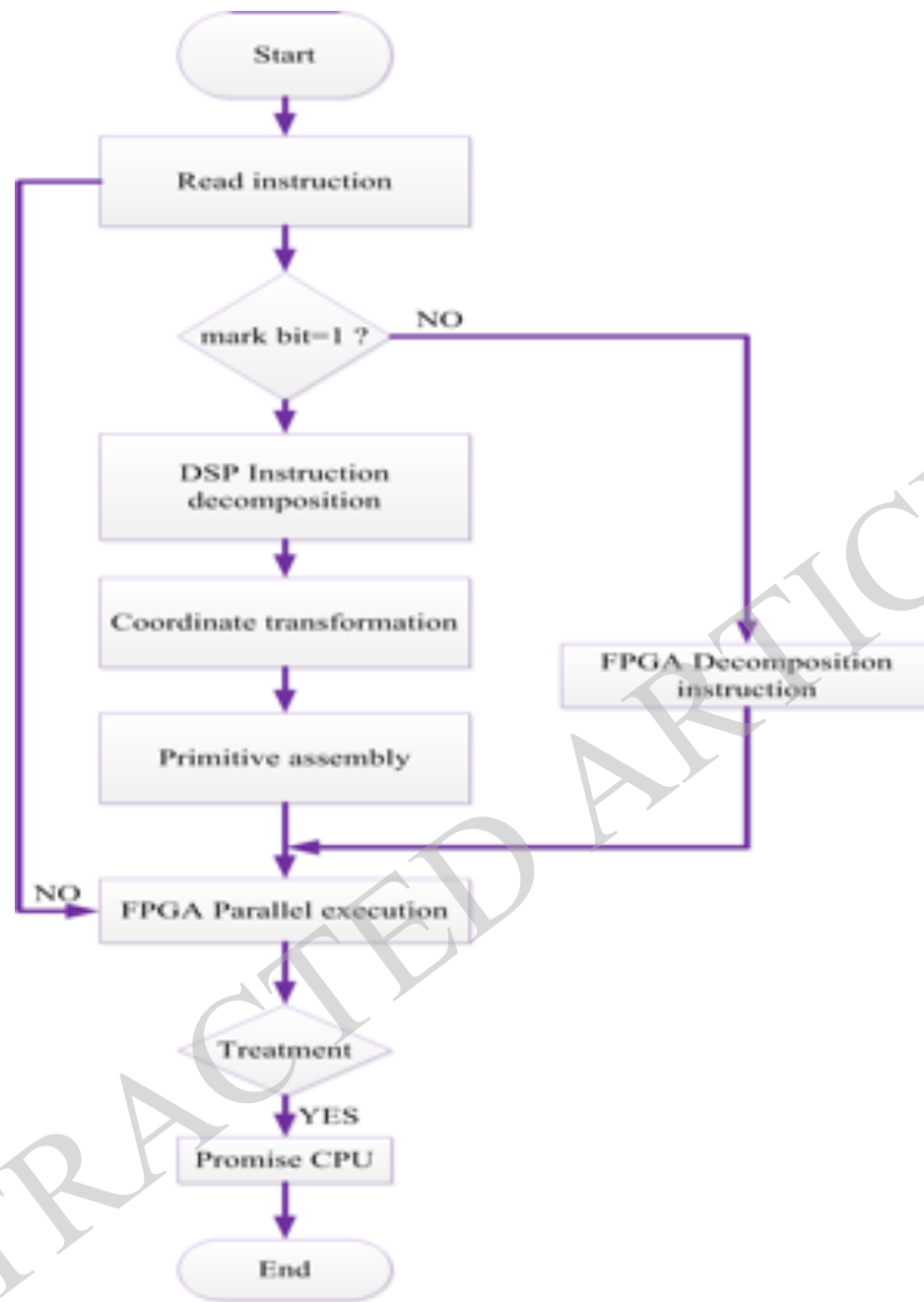


Fig. (8). Instruction execution flow chart.

processing, animation, and more advanced features, such as texture mapping, motion blur and atomization effect to achieve realistic 3D rendering and establish interactive 3D scene laid a good foundation. At present, OpenGL has become a standard of 3D graphics development, which is a necessary tool for the development of 3D graphics [9].

6.1. (3D) Modeling

In 3D GIS, 3D objects are generally represented by polygonal patch combination, in dealing with the vertices of the polygon information description and storage. According to the geometrical properties of different objects [10-12]:

1. For the surface of the object, a mathematical function can be formed. OpenGL provides a lot of functions to draw the two surfaces, such as `gluCylinder ()` for the drawing of a cylindrical table, `gluSphere ()` used to draw the ball, *etc.*
2. For irregular surfaces, a series of discrete data points can be used to define. OpenGL provides evaluation device based on Bezier curves and surfaces, and can directly use the function `gluNurbsSurface ()` by definition surface control points to draw the NURBS surface.
3. In a 3D scene still exist some fuzzy scenery, such as smoke, dust, oil and gas, *etc.*, they present greater

randomness and irregularity, the same is also composed an important part of the natural scenery. In OpenGL can be activated by calling the function `glEnable (GL_FOG)`, and can be used to select the color and density of the fog `glFog*`.

4. Into the third party the modeling files, for some very complex shape can use third-party software, such as Autodesk 3ds Max modeling software function is very powerful, can be very easy to generate the complex object model. Even more important, its graphics file format (*.3ds) has a very strong intelligibility and generality. Through the program can be 3DS format graphics file conversion for the C language to support the file format to obtain data object model, then import it into the construction of the 3D scene. Only in this way can greatly shorten the complex objects modeling time, improve the efficiency of modeling.

6.2. Coordinate Transformation

Will 3D objects in the 3D virtual scene on the computer screen displayed from the three-dimensional space to two-dimensional projection process, like taking pictures with the camera, usually through the following steps:

1. The camera is placed on a tripod, and the 3D object is to be shot, which is equivalent to the viewpoint transformation (Viewing Transformation) in the OpenGL, *i.e.*, the position of the adjusting viewpoint [13].
2. The three-dimensional object is placed in the appropriate position in the scene, which is equivalent to the model transformation (Modeling Transformation) in OpenGL, that is, rotation, translation and scaling of the model.
3. The camera lens and focusing, make the projection of a 3D object in the two-dimensional film, it is quite in OpenGL projective transformation, projection transformation, namely 3D model projection to two-dimensional screen process.
4. Camera photographs, to determine the size of 2D photos, it is equivalent to the OpenGL viewport transform (ViewportTransformation) and the provisions of the screen shows the location and size of the scene.
5. Through the above steps, a 3D object in a three-dimensional space can be represented by the corresponding two-dimensional objects, it can be displayed on the two-dimensional computer screen.

6.3. Blanking Technology

In 3D scene modeling, using polyhedral to simulate three-dimensional objects, but from a perspective of observation of a 3D object can see just objects on the surface part of point, line and plane, and the rest is the visible part of the occlusion. If you have multiple 3D objects in a 3D scene, the object may not be visible to some of the objects. Fire hidden technology is in determining the viewpoint, the

objects on the surface of the invisible plane eliminated, the drawing objects is more realistic [14].

It through the function `gl-DepthFunc (GLenum func)` to test the depth to develop different comparison functions, through comparing new pixel depth values and stored in the depth buffer depth value, on the basis of whether meet the comparison function to decide whether the pixel is drawn.

6.4. Illumination Model

In order to create a real sense of strong virtual objects to a realistic three-dimensional effect, but also for illumination, material, texture, atomization and anti-aliasing *etc.* the treatment effect.

OpenGL light illumination model will light subdivided into four independent components: radiation light Emitted environmental light AmbientLight, diffuse light, diffuse light and reflected light SpecularLight. OpenGL by function `void glColorMaterial (GLenum light, GLenum pname, GLenum param)` to specify the various properties of the light source. One of the second parameter `pname` specifies the light source, the parameters `GL_AMBIENT`, `GL_DIFFUSE`, `GL_SPECULAR`, `GL_POSITION` define the ambient light, diffuse light, reflected light and light position.

7. SYSTEM IMPLEMENTATION

With the development of three-dimensional virtual road system, the process of 3D modeling using OpenGL is described in detail [15].

7.1. System Framework

The system uses C++6.0 Visual and 3D graphics development interface OpenGL as software development platform. New build a MFCAppwizard (exe) project, through VC++ interface of the Project menu, select Add to project command, add the OpenGL three static library file: `gluax.lib`, `glu32.lib` "opengl32.lib", then in the view class of CPP files added to the header files in the static library: `gl\gl.h`, `gl\glu.h`", `gl\gluax.h`. So you can use the OpenGL function to draw three-dimensional graphics on the VC++ platform [16].

7.2. System Drawing

Using OpenGL to create three-dimensional scene, drawing process is the main package, include the following steps.

7.2.1. Pixel Format

Before you create a graphics operation table, you must first set the pixel format. Pixel format contains graphics device interface attributes. These attributes determine the graphics interface is RGBA mode or color table model, pixel buffer is with a single cache or double buffer, color number, depth buffer and stencil buffer number and some other attribute information.

In order to describe the pixel format, the `PIXELFORMATDESCRIPTOR` structure must be filled.

7.2.2. Set Pixel Format

After completion of the pixel format, the ChoosePixelFormat function is passed to the Win32. The function of the Windows system is supported by the actual pixel format and the device pixel format is supported by the device with a given pixel format for comparison, and then returns the best matching. Then use the SetPixelFormat function to set the pixel format, the specific implementation of the statement is as follows [17]:

```
CClientDC clientDC(this);
int PixelFormat=ChoosePixelFormat
(clientDC.m_hDC,&pf);
BOOL
result=SetPixelFormat(clientDC.m_hDC, PixelF
ormat,&pf);
```

7.2.3 Create a Drawing Description Table

The drawing description table is the port that is passed through all the OpenGL commands. The drawing description table has the same pixel format as the device associated with it. The glCreateContext function creates a drawing description table, the function returns the type HGLRC, whose value is an effective device description table handle.

7.2.4. OpenGL Drawing

In the system of the oncreate event, describes the required pixel format and the transfer function ChoosePixelFormat, the function returns and specify the pixel format that best matches the device descriptor table, pixel format, then the this format is transmitted to the SetPixelFormat established the right window format, and then wglCreateContext function create drawing description table to accept the OpenGL commands.

OpenGL through graph transformation to three-dimensional objects in the real world in the form of two-dimensional plane performance on the computer screen, using color, light, mixed, anti aliasing, atomization, bitmap and image and texture mapping technology to simulate real world and draw a perfect three-dimensional entity model.

7.3. Simulation of Natural Phenomena Such as Rain and Snow

In OpenGL, the viewport is window of the rectangular plot, any used to display two-dimensional or three-dimensional shapes are drawn in the viewport. OpenGL support build multiple viewports in the same window, and the use of each viewport different projection and view transformation to render the object. For scene roaming system moderate rain, snow simulation model redefines the viewport view-port size a and rendering of the scene the same, make the viewport in the drawing of the object is always in the viewer, and in the viewport in sight of the effective depth, from a certain height of the particle source downward spray the generation technology of rain and snow particle fusion.

7.4. Animation Roaming

In OpenGL, the user can operate the keyboard, mouse and other input devices, to achieve human-computer interaction, users can browse from different perspectives. In order to realize the 3D scene animation, the control is determined by setting the viewpoint of the viewpoint. In OpenGL concrete is the use of auxiliary library gluLookAt () function, by setting the appropriate parameters to achieve the effect of roaming. In this paper, the modeling technology of 3D GIS scene is described simply. A 3D GIS is a comprehensive technology. It relates to cartography, geography, mapping science, computer science and technology, planning and management, etc. and many other subjects. The research contents of 3D GIS is very complex. The design also includes data acquisition, data storage, data management, data model, visualization expression, spatial query and spatial analysis and so on. These have yet to be further and deeper study. As a developing standard of 3D modeling, OpenGL is a necessary tool for 3D modeling and development., extensibility, given its strong vitality and practical prospect, OpenGL has been widely used in CAD / CAM / CAE, medical image processing, virtual reality, entertainment, advertising and other areas, so the research and application of OpenGL has a great significance.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

This work is supported by the science foundation of educational department of HeBei province(Nos:20130209).

REFERENCES

- [1] B. Chen and H.H. Cheng, "Interpretive OpenGL for computer graphics", *Computers & Graphics*, vol. 29, no. 3, pp. 331-339, 2005.
- [2] J. Batlle, J. Marti, P. Rida and J. Amat, "A new FPGA/DSP-based parallel architecture for real-time image processing", *Real-Time Imaging*, vol. 8, no. 5, pp. 345-356, 2002.
- [3] A. Carrara, M. Cardinali, R. Detti, F. Guzzetti V. Pasqui and P. Reincenbach, "GIS techniques and statistical models in evaluating landslide hazard", *Earth Surface Processes and Landforms*, vol. 16, no. 5, pp. 427-445, 1991.
- [4] Q. Weng, "Land use change analysis in the Zhujiang Delta of China using satellite remote sensing, GIS and stochastic modelling", *Journal of Environmental Management*, vol. 64, no. 3, pp. 273-284, 2002.
- [5] K.C. Clarke, L.J. Gaydos, "Loose-coupling a cellular automaton model and GIS: long-term urban growth prediction for San Francisco and Washington/Baltimore", *International Journal of Geographical Information Science*, vol. 12, no. 7, pp. 699-714, 1998.
- [6] S. Aronoff, *Geographic information systems: a management perspective*, WDL Publications, Ottawa, Canada, 1989.
- [7] N.J. Obermeyer, "The evolution of public participation GIS", *Cartography and Geographic Information Systems*, vol. 25, no. 2, pp. 65-66, 1998.

- [8] J. Geoghegan, L.A. Wainger and N.E. Bockstael, "Spatial landscape indices in a hedonic framework: an ecological economics analysis using GIS", *Ecological Economics*, vol. 23, no. 3, pp. 251-264, 1997.
- [9] L. Ayalew and H. Yamagishi, "The application of GIS-based logistic regression for landslide susceptibility mapping in the Kakuda-Yahiko Mountains, Central Japan", *Geomorphology*, vol. 65, no. 1, pp. 15-31, 2005.
- [10] J. Malczewski, "GIS-based land-use suitability analysis: a critical overview", *Progress in Planning*, vol. 62, no. 1, pp. 3-65, 2004.
- [11] M. Šúri and J. Hofierka, "A new GIS - based solar radiation model and its application to photovoltaic assessments", *Transactions in GIS*, vol. 8, no. 2, pp. 175-190, 2004.
- [12] L. Yan, T. Zhang and S. Zhong, "A DSP/FPGA-based parallel architecture for real-time image processing", In: *The Sixth World Congress on IEEE Intelligent Control and Automation*, WCICA, vol. 2, 2006, pp. 10022-10025.
- [13] F.J. Rodriguez, S. Cobreces, E.J. Bueno, Á. Hernández, R. Mateos and F. Espinosa, "Control electronic platform based on floating-point DSP and FPGA for a NPC multilevel back-to-back converter", *Electric Power Systems Research*, vol. 78, no. 9, pp. 1597-1609, 2008.
- [14] Y. Lei, Z. Gang, R. Si-Heon, L. Choon-Young, L. Sang-Ryong, and K.-M. Bae, "The platform of image acquisition and processing system based on DSP and FPGA", In: *International Conference on IEEE Smart Manufacturing Application*, ICSMA, 2008 pp. 470-473, 2008.
- [15] Z.J.G.Q.H. Lu, "Solution of data communication based on DSP and FPGA", *Electronic Measurement Technology*, vol. 1, p. 031, 2009.
- [16] M.J. Kilgard, *The OpenGL Utility Toolkit (GLUT) Programming Interface*, 1996.
- [17] D. Bartz, M. Meißner, and T. Hüttner, "OpenGL-assisted occlusion culling for large polygonal models", *Computers & Graphics*, vol. 23, no. 5, pp. 667-679, 1999.

Received: September 16, 2014

Revised: December 23, 2014

Accepted: December 31, 2014

© He et al.; licensee Bentham Open.

This is an open access article licensed under the terms of the (<https://creativecommons.org/licenses/by/4.0/legalcode>), which permits unrestricted, non-commercial use, distribution and reproduction in any medium, provided the work is properly cited.

RETRACTED ARTICLE