RESEARCH ON KINEMATICS SIMULATION OF PARAMETERIZED MECHANISM BASED ON VISUALIZATION IN SCIENTIFIC COMPUTING

Gang Zhao, Liangxi Xie*

College of Mechanical Automation, Wuhan University of Science and Technology, Wuhan, Hubei, 430081, P.R. China

* *Corresponding Author, Address: P.O. Box 242, College of Mechanical Automation, Wuhan University of Science and Technology, Wuhan Hubei, 430081, P.R. China, Tel:+86-27-62423171, Email: lx_tse@163.com*

Abstract: It is fundamental for the kinematics simulation and mechanism design to apply the visualization in scientific computing on the simulation research of 3D mechanism. It is also the basis of independent development for 3D CAE software without any copyright limit. This paper offered a parameterized kinematics simulation template for 3D mechanism based on the visualization in scientific computing, in which the open graphics library is applied by MFC compiler. This paper illuminates the fundamental principle of the simulation development applying the OpenGL technology. The principle of OpenGL, the procedure structure, the working flow of OpenGL, the concept of program designing and the code implementation based on VC++ is demonstrated and an application approach for mechanism simulation is developed under the MFC environment.

Keywords: visualization in scientific computing, MFC compiler, OpenGL, parameterized, kinematics simulation for 3D mechanism

1. VISUALIZATION IN SCIENTIFIC COMPUTING APPLIED IN KINEMATICS SIMULATION

Visualization in Scientific Computing was first present in a report (Computer Graphics, 1987, Vol. 6) submitted to US National Science Foundation in 1987 by B. H. McCormick. It is goal to show the evidence between the data content and the graphics by an organic synthesis, which is formed by a mass of data obtained by digital computation or experiment according to its physical background. It is advantageous to totally grasp the designing evolution, discover the internal physical principle, enrich the scientific research and shorten the research period for Visualization in Scientific Computing. In an application of kinematics simulation by the visual technology, the problem is how to construct a physical model based on a data set, how to set up a visual model and implement a concrete graphics technology (Pu, 2004).

The visualization technology first transforms the abstract description of data to a description of graphic element and forms a visual physical model constituted by graphic elements such as vertex, curve, surface and solid. And then plots the visual model by the plot technology based on computer graphics, which extract the common properties of geometric elements to implement the data visualization, graphic parameterization, graphic property and property parameterization. Graphic parameterization includes the parameter control of geometric element, lighting model, perspective spot and projection type. Graphic property includes color, transparence, optical property of material and mapping texture. The property parameter and graphic parameter form the inter-visual computing model. In the plot technology, surface plot is basic to form the graphic element by vertex, curve and geometric contour based on them. Compared with this plot technology, solid plot distinguish itself from surface plot technology, which directly generate a solid contour by solid element, not an intermediate surface.

Under the principle of tomography, the visualization technology decreases the model dimensions and factually simulates the data field of solid by correct method offered by computer graphics. Solid data field exits in 3D space of research system with the time elapse. Visualization of solid data field is a mapping from data set to geometric property. It extracts solid data about space time according to the need. And that data with variable types is extracted when data is displayed.

2. IMPLEMENTATION OF MECHANISM KINEMATICS SIMULATION BY MFC COMPILER

The setup of parameterized model is a need for graphic intercourse on mechanism analysis and kinematics research. By several control parameter settings and the analysis for 3D parameterized graphic model, the best effect on geometric entity, the pertinence analysis of parameters, the parameter dependence on objects and the harmony between objects is obtained. All these parameters are the man-machine interface. It is hard to set up a visual parameterized model for kinematics simulation by that professional software such as ProE, Solidworks, ADAMS and so on, which is limited by the compiled code or the copyright. Therefore by the implementation of open graphic library in MFC compiling environment, it is relatively easy to develop a system of kinematics simulation, all copyrights reserved.

Generation of 3D graphic for solution is the object-oriented basic for the simulated objects by OpenGL. The question is transformed to the object and scene processed by OpenGL. Scene is constructed by geometric and graphic elements such as vertex, curve and surface, and complex scenes need some better algorithms for surface generation. By a set of image process for objects and its internal elements, such as material definition, lighting, polygon antialiasing, fairing, transparence and image synthesis, texture mapping, atomization etc, anticipated realistic effect is obtained after the geometric solid is generated in scene.

Because it is a software interface for special graphic device, a program language compatible with C++ grammar for visualization technology, OpenGL is advantageous to visualization in scientific computing. As a software interface for graphic device, OpenGL is composed of hundreds of commands and functions. These commands allow the users to instruct the 2D or 3D geometric object and operate object for frame rendering. OpenGL is a command set to operate the graphic hardware and retain a mass of geometric information. Due to the network transparence, OpenGL can transfer the graphic information to remote client servers or display devices, even the shared operation with other systems. OpenGL offers an enhanced capability of plot to ensure the reliability of popular graphic application, which is applied in the coming generation fields of medical imaging, geography information, prospect for petroleum, climate simulation and amusement animation, etc (He, 1994).

It is a technology question to choose, describe and plot the graphic elements when 3D mechanism is constructed. Fundamental graphic element is the most minimum figure unit in a plotting operation. An ideal description of fundamental graphic element shall be expressed by a flexible data structure, a correct data operation and mechanism recognition. Consequently, the ideal graphic element according with the need is the various components and the kinematic pairs. Considering the complexity of component contour and the diversity of kinematic pair, the fundamental graphic element RRR , RRP and RPR group is applied to construct the complicated mechanism of second grade group (Wang, 2001) (Cheng, 2003).

3. STRUCTURE OF OPENGL

As the software interface about graphic hardware, the principle function of OpenGL is to convert the 2D or 3D objects data to image frame buffer. Those solids are composed of a series of vertexes describing the geometric attributes of objects and the pixels describing the images. OpenGL convert that solid data to pixel data and form a final display in frame buffer of image. The fundamental functions of OpenGL include the core function of OpenGL, implement library function, auxiliary library function, windows professional function and Win32 API function. The fundamental structure of OpenGL includes GLYPHMETRICSFLOAT, LAYERPLANEDESCRIPTOR, PIXELFORMATDESCRIPTOR, POINTFLOAT.

Flow chart of OpenGL includes the operation and command of graphic element, graphic control, interpreted language of OpenGL command, fundamental operation of OpenGL, as the following Fig.1.

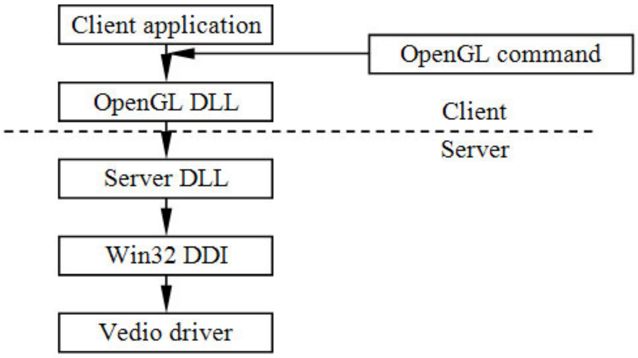


Fig.1 Flow chart of OpenGL on Windows

Interpreted mechanism of OpenGL command is based on client /server mode, in which the request from client application is interpreted by kernel server of OpenGL. Based on the client /server mode, it is convenient for various users on various computers to share the service offered by other servers. That’s to say the structure of OpenGL is transparent to network.

Library function of OpenGL is packed in the dynamic link library file opengl32.dll. OpenGL function access published by client application is first operated by file opengl32.dll. After transferred to server, it is processed by file winsrv.dll and then transferred to DDI, finally to video display driver.

User’s command is divided into two parts since it entered OpenGL. One part plots certain geometric solid and the other instructs the program how to operate solids at various stages. Many commands may be listed in display list as queuing messages waiting for operation. An effective method for curve and surface approximation is offered by the polynomial function, whose input value is offered by valuator. Then the geometric element described by vertex is operated. At this stage, vertex is transformed and rendered, graphic element is clipped to perspective space and the raster display is ready. Raster generates the frame buffer address for a series of images and the 2D expression of graphic element. The generated result is called substrate. Each substrate is fit to operate the single substrate before the frame buffer is changed. These operations include the buffer update according to the field depth, various tests and image synthesis, shield operation for substrate color and saved color, the logical operation and fade effect for substrate (Yan, 1995).

Operation of pixel data makes the pixel, displacement and image to be raster displayed after pixel operation. Because the graphic element of OpenGL is expressed by vertexes, it is convenient to operate data vertex by vertex, assemble the fundamental units by them, and then form substrate by raster display. As for the pixel data, the results are saved in texture memory in order to access pixel information from texture memory for raster display.

Display list of OpenGL is factually a function group, which is saved for OpenGL operation in future. The valuator of OpenGL is factually composed of many special functions, which generates the coordinate of vertex, normal, texture, color and transfers operational result to execute module, by a polynomial mapping about one or two variables (Ouyang, 2004). Valuator interface offers a template to generate curves and surfaces based on OpenGL. Compared with complex NURBS interface, the valuator interface in OpenGL is more advantageous when polynomial valuator is efficiently applied to express the non-NURBS curves and surfaces and special surface attributes without the NURBS transform (Yu, 2001).

Finally, the raster includes two part, geometric and physical mapping. Geometric operation turns graphic element to 2D image, physical operation computes the color and depth information of each vertex. Therefore, a raster of graphic element is completed by two steps. Firstly, the pixel occupied by graphic element in an integral raster of window coordinate is determined. Secondarily, the value of color and depth is computed for each pixel. The computed results are conferred to next step of OpenGL for update of correct units in frame buffer.

To operate graphic by OpenGL and finally plot 3D scene on screen, the main steps include:

(1) Format pixel. Format important OpenGL information such as plot style, color mode, color bit, depth bit, etc.

(2) Set up geometric model. Set up and demonstrate the 3D model by mathematics and fundamental graphic elements.

(3) Set scene. Determine the location of object in 3D space and set the perspective correctly.

(4) Effects setting. Set material of object and lighting condition.

(5) Raster display. Convert the information of scene and color into the pixel information displayed on screen.

4. VC++ IMPLEMENTATION OF KINEMATICS SIMULATION FOR 3D PARAMETERIZED MECHANISM

4.1 Concept of Modularized Program Designing

System frame of real-time simulation for mechanism is implemented by OpenGL as the following Fig.2.

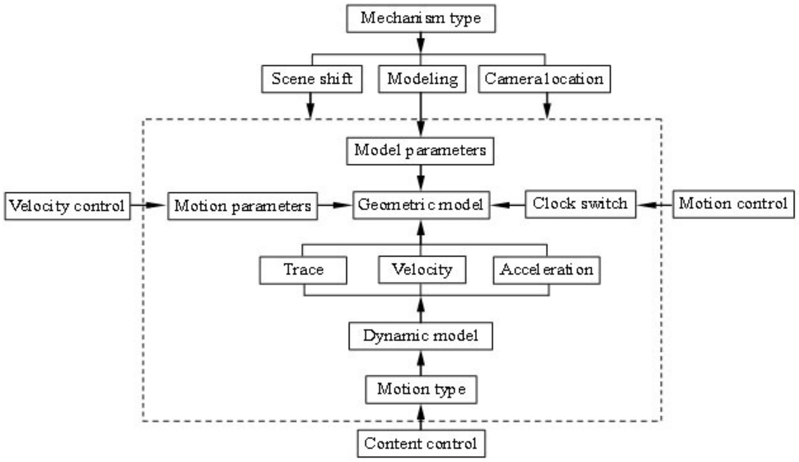


Fig.2 OpenGL implementation of kinematics simulation for 3D mechanism

Considering the concept of modularized program designing, mechanism type is determined by writing a simple conditional sentence in the window drawing function void CSimulationView::OnDraw(CDC* pDC). With the increase of mechanism types, program adds the corresponding head files and case condition in function OnDraw(). It is no need to modify the main frame of the program. Object-oriented Concept of program designing is implemented overall. The application codes for choice of mechanism type in function OnDraw() are listed as follows,

```
void CSimulationView::OnDraw(CDC* pDC)
{ CSimulationDoc* pDoc = GetDocument();
  ASSERT_VALID(pDoc);
  switch (mechanism)
  {  case Planer:
```

```
            DrawScene();
            break;
        … …
        default:
            OnFileNew();
            break;
    }
}
```

4.2 Generation of Geometric Solid

OpenGL offers a number of drawing functions for basic geometry solid. Under the VC++ compiling environment, a head file to draw the basic solid for the corresponding mechanism type is added into project. This head file cites the file glaux.h from the open graphic library. Consequently, the plot function can be directly loaded from library file. For the kinematics simulation of crank rocker mechanism, the head file GeoPlaner.h loads the library file gluCylinder() to draw rods, whose application codes are listed as following,

```
void DrawBar(UINT nLists, GLfloat m_radius, GLfloat m_length)
{ GLUquadricObj* quadObj;
  glNewList(nLists, GL_COMPILE);
  quadObj=gluNewQuadric();
  gluQuadricDrawStyle(quadObj, GLU_FILL);
  gluQuadricOrientation(quadObj, GLU_INSIDE);
  gluQuadricNormals(quadObj, GLU_SMOOTH);
  switch (nLists)
  {  case 1: case 5:
     gluCylinder(quadObj, m_radius, m_radius, m_length, 10, 1);
     break;
  … …
  }
  glEndList();
  gluDeleteQuadric(quadObj);
}
```

Only the plot function for basic solids is offered by OpenGL, so it is relatively difficult to construct a complex model. In order to construct various complex solids efficiently, 3DMAX is introduced as the professional 3D graphic software (Zhang, 2003). It is an efficient approach to construct the complex geometric solid by 3DMAX and control or convert these solids by OpenGL. At first, construct a complex solid by 3DMAX and then extract the space coordinate for vertexes of 3D solid and the material information to form a C++ binary source file by model transform utility named 3D Exploration. Domination program reads the corresponding binary file by file interface to generate a display list dominated by OpenGL when the certain 3D object needs to be displayed. The file interface of 3D model includes a read interface function and a generation function of display list. Users substitute specified material for model material and implement an intercourse control of object materials by generation of display list interface function (Zhu, 2000).

4.3 Implementation of Motion Simulation

Motion parameters are used to locate the motion objects, which is based on the object coordinate and local coordinate in OpenGL. The number of motion parameters is equal to freedom. The motion parameters control the motion speed by man-machine interface.

Absolute motion is implemented by a rational mathematics model based on a specified geometric model. Motion object in 3D space is located by motion parameters in real-time simulation. OpenGL implementation of absolute motion is demonstrated as the Fig.3.

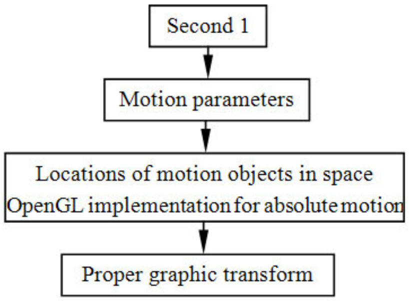

Second 1
Motion parameters
Locations of motion objects in space
OpenGL implementation for absolute motion
Proper graphic transform

Fig.3 OpenGL implementation of absolute motion for mechanism

OpenGL convert the absolute motion into relative motion of objects about the dynamic reference frame by its certain expression. There are two ways to get it. One is to take advantage of a relative coordinate directly. The other is to demonstrate a relative motion based on an existed absolute motion. Obviously, the second way is more efficient for a complicated motion mechanism. OpenGL implements a relative motion as the following Fig.4. When dynamic reference frame restores to the second $t = 0$, the graphic transform corresponding to the control parameters at this second locates all motion objects in virtual space from top source. Consequently, various complex relative motions are simulated easily.

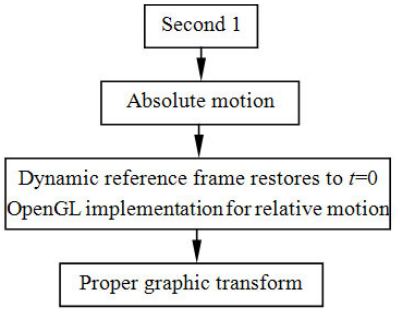

Second 1
Absolute motion
Dynamic reference frame restores to *t*=0
OpenGL implementation for relative motion
Proper graphic transform

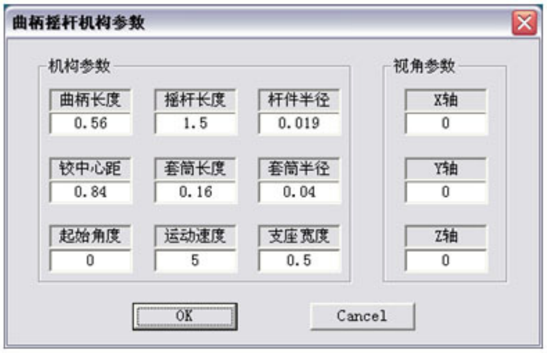
Fig.4 OpenGL implementation of relative motion for mechanism

Whatever absolute motion or relative motion, time engine for mechanism motion is implemented by the following codes,

```
void CSimulationView::OnTimer(UINT nIDEvent)
{ m_rotateAngle+=m_speedFactor;
  if (m_rotateAngle>360.0f)
  {   m_rotateAngle=0.0f;
  }
  Invalidate(FALSE);
  CView::OnTimer(nIDEvent);
}
```

4.4 Performance of Man-machine Interaction

Based on OpenGL, man-machine interaction of kinematics simulation embodies some aspects such as model setup, motion control, content control, perspective control, camera location, scene shift, etc. Dialog box in simulation of crank rocker mechanism is shown as the Fig.5. In order to understand the motion of a certain object or a local detail, it is necessary to observe the absolute motion, relative motion, motion trace, velocity and acceleration, etc. in various perspectives. Perspective parameters are determined by variables m_perspectiveX, m_perspectiveY and m_perspectiveZ, which can be given in dialog box or real-time controlled by cursor message series.



曲柄摇杆机构参数

机构参数

曲柄长度	摇杆长度	杆件半径
0.56	1.5	0.019
铰中心距	套筒长度	套筒半径
0.84	0.16	0.04
起始角度	运动速度	支座宽度
0	5	0.5

视角参数

X轴
0
Y轴
0
Z轴
0

Fig.5 Dialog box of simulation parameter input

4.5 Effect of Interface and Professional Splash

It is not enough just to simulate the motion of mechanism. In order to obtain the better effect closer to the real, it is necessary to render the interface image by functions such as lighting effect function EnableLighting(), background effect function DrawBackground(), coordinate display function DrawBackground(), velocity control function OnSpeed(), image size control function OnSize(UINT nType, int cx, int cy), function of sound effect and motion control OnControlMove(), etc. Splash effect is completed by class CSplashWnd to make the simulation software more professional. It shows the running interface as the Fig.6.

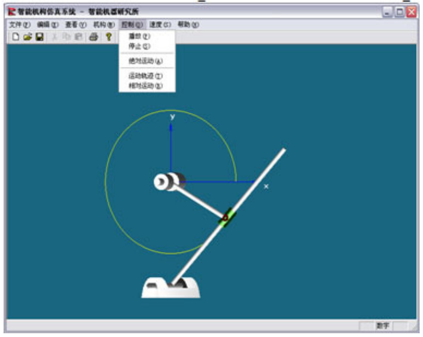


Fig.6 Interface of parameterized kinematics simulation of 3D mechanism

5. CONCLUSIONS

When the simulation software is completed, some professional tools such as Installshield are used to archive and encrypt the program files. The obtained installation package is up to the trade standard. Now, the simulation software has had the fundamental elements belonging to the professional software. It can efficiently set up the mechanism, simulate the motion, plot the trace of any point on mechanism, output any kinematic variables for analysis adjust the angle of view and zoom the field of view, observe the details about the motion properties. The flexible manipulation, the better vision effect, the real and reserved copyrights and so on, are the advantages of the simulation software.

A kinematics simulation template for parameterized 3D mechanism is constructed by the open graphic library and the open MFC compiler based on the visualization in scientific computing. Many attempts are benefit to the independent development for software of kinematics simulation and mechanism construct in this paper. Due to being the fundament of virtual reality and manufacturing information technology, visualization in scientific computing is efficiently applied in the mechanism simulation, the mechanism construct, the kinematics analysis and the dynamics field.

REFERENCES

Pu Zhixin, Gu Yanfeng, Li Ying, Realization of NC lathe-turning simulation software based on OpenGL [J], Machine Tool & Electric Apparatus, 2004; (2): 14-16

He Xiaoli, Du Ying, Visual C++ MFC class library design & user manual [M], the first edition, Beijing: Xueyuan Press, 1994; 223-229

Wang Chengzhi, Huang Kaixuan, Chen Feiyan, Research on mechanism drawing and animation system based on computer visualization [J], Machinary Design & Manufacture, 2001; (1): 24-25

Cheng Chonggong, Hu Guanyi, Dai Juan, The research of programme method for kinematics analysis based linkage group [J], Journal of Changsha University, 2003; 17 (4): 27-30

Yan Wen, C++ graphic program design — C++ interface and graphic program instance [M], the first edition, Beijing: Science Press, 1995; 177-192

Ouyang Zhen, Di Ruikun, Qin Feng, Research and exploitation of NC simulation control system based on OpenGL [J], Machine Tool & Hydraulics, 2004; (5): 66-68

Yu Bin, Liu Rongzhong, Research on simulation system for NC manufacturing based on OpenGL [J], Journal of Sichuan University (Engineering Science Edition), 2001; 33(5): 16-19

Zhang Wei, Li Yibing, Hu Yuanzhi, Construction of interactive virtual 3D accident scene based on OpenGL [J], Journal of Highway and Transportation Research and Development, 2003; 20 (2): 108-111

Zhu Qidan, Su Peng, Zhang Weiming, Application of the visual modeling method for 3D graph based on OpenGL [J], Techniques of Automation and Applications, 2000; 19 (1): 27-29