Research on the Interactive Technology of Reservoir 3D Display

Zhang Yan[1, a], Nie Yongdan[1,b], Hou junlong[1,c], He Wenjie[1,d]

[1] School of Computer and Information Technology, North East Petroleum University, Daqing,China

[a]zhangyuanyan_309@126.com

Keywords: Open Inventor, DialogViz, MeshViz, model slice, reservoir.

Abstract. In the process of reservoir 3D modeling and display, when the data amount of the model was large, the visual model would become huge. While the oil exploration and development experts might usually pay more attention to the distribution of the cells within some layers, some areas, or some regions with specific attribute range. This article used the MeshViz and DialogViz extended module of the Open Inventor to achieve the interactive display technology including extracting slices of the 3-dimensional model and the filtering valid cells based on the specific range of the attribute value. The proposed method could extract specific slices and display the ROI region interactively, and could help the user to analyze the characteristics of three-dimensional reservoir model more efficiently.

Introduction

Currently the OpenGL and Open Inventor were the most widely used three-dimensional graphics package. The feature of OpenGL was too powerful and complex to master for learners in a short period of time. Development based on the OpenGL usually needed too many code to make the tedious settings in three dimensional scenes such as graphic projection, color, texture, light, and so on. While the Open Inventor was the most widely used object-oriented graphic and images software development interface, it allowed the user to build a complex three-dimensional scene through the building blocks like approach, so that the users only needed to spend a little period time to construct a complex three-dimensional scene. With the Open Inventor 3D visualization technology, users could quickly and easily generate 3D stratum model display, So studying the fine display of numerical reservoir simulation data model in depth based on Open Inventor, could not load the extra geological objects and data, but also provide a more intuitive understanding of the geological conditions of the target block and reserves information, thereby providing better decision support for oil exploration and development [1].

Reservoir 3D Display

According to the particular needs of different fields, the Open Inventor included a number of extended modules. The MeshViz XLM extended module provided powerful functions of model grids drawing in the field of applied science, model grids extraction and data matching, and contained advanced data visualization design components, complex surfaces and 3D charts components, and other visual objects. The DialogViz extended module provided the programming interface that could interact with the 3D scene to control and manipulate the objects in the scene, and the usage of DialogViz node was very like the other the Open Inventor nodes [2]. Because of the characteristics of irregular hexahedral cells in the reservoir model, the MeshViz XLM module was used to achieve the three-dimensional display. The topology relationship of pillar model was organized by the MbHexahedronTopologyExplicitIjk type object, and the geometry objects and information were stored and managed by the MiGeometryI type object. The reservoir model were stored by the MoMesh type object, and displayed in the scene, including the expression of the formation data

Foundation item: Scientific Research Fund of Heilongjiang Provincial Education Department (No.12521050), National Science and Technology Major Project of the Ministry of Science and Technology of China (No.2011ZX05021),CNPC Innovation Foundation (No 2012D-5006-0609)

attribute information, while the model attribute value were stored in the MoScalarSetIJK types of objects, and then the material binding method specified by MoLinearColorMapping type object was taken to render by the MoLinearColorMapping type object.

There were two ways of Open Inventor material binding, PER_CELL and PER_NODE. The former method was chosen in this paper to bind the material rendering to the cell, which expressed the different attribute information. The property categories of strata model to display was optional, and the user could view the property values of any cell in real time [3]. The reservoir 3D display provided an intuitive analysis of the strata and reserves information for geological researchers and decision-makers, combined with a variety of interactive and customized tools. The Figure 1 showed the Water saturation attribute parameter display of a certain reservoir model.

Interactive node

The SoClipPlaneManip manipulator was developed to allow easy interactive manipulation of the clipping plane. The SoClipPlaneManip utilized a SoJackDragger to provide a 3D interface for moving the clip plane. This manipulator derived from SoClipPlane could be maintained like all other manipulators, consistency between the fields of the dragger and its own fields. The SoClipPlaneManip node was used in this paper to achieve the function of the slice of rotation at any angles along the coordinate axis. The key codes are follows:

```
SoClipPlaneManip initiation:
m_ClipPlane->draggerPosition.setValue(m_meshCenter);
m_ClipPlane->plane.setValue(SbPlane(SbVec3f(0,1,0),m_meshCenter));
SoJackDragger* dragger = (SoJackDragger *) m_ClipPlane->getDragger();
dragger->scaleFactor.setValue(bbSize/10);
Scene settings:
mainScene->addChild( m_switchPlaneSlice );
{
  SoSeparator* app = new SoSeparator;
  m_switchPlaneSlice->addChild( app );
  app->addChild (m_moMesh);
  app->addChild(m_switchPlaneSliceWireFrame);
  {
    m_switchPlaneSliceWireFrame->addChild(m_style);
    m_switchPlaneSliceWireFrame->addChild(m_material);
  }
  app->addChild(m_meshPlaneSlice );
  m_switchPlaneSlice->addChild(m_ClipPlane);
}
```

Interaction

Controls definition

The DialogViz module included many nodes such as SoTopLevelDialog, SoMenuBar, SoMenuPopup, and some nodes derived from the SoDialogGroup, like SoColumnDialog SoRowDialog, and SoTabDialog. DialogViz initialization supported in two ways, it not only could directly initiate from the program by using the code, but also could read from the already definition files [5].

The key codes of using SoDialogCheckBox to achieve SoClipPlaneManip interactive were as follows:

```
SoDialogCheckBox* sliceCheck =
(SoDialogCheckBox *)m_topLevelDialog->searchForAuditorId("planeslicecheck");
if (sliceCheck != NULL)
```

```
sliceCheck->addAuditor(new uditorCheckSwitch(m_sceneGraph->getSwitchPlaneSlice()));
 // plane slice WireFrame check
 SoDialogCheckBox* planeSliceWireFrameCheck =
(SoDialogCheckBox *)m_topLevelDialog->searchForAuditorId("planeslicewireframe");
 if (planeSliceWireFrameCheck != NULL)
planeSliceWireFrameCheck->addAuditor(new
AuditorCheckSwitch(m_sceneGraph->getSwitchPlaneSliceWireFrame()));
 // plane slice clipping
SoDialogCheckBox* sliceClip =
(SoDialogCheckBox *)m_topLevelDialog->searchForAuditorId("planesliceclipping");
if (sliceClip != NULL && m_sceneGraph->getClipPlane()!=NULL)
m_sceneGraph->getClipPlane()->on.connectFrom(&sliceClip->state);
```

The Figure 2 showed the slice along the Z-axis direction, and rotated by 160 degrees.

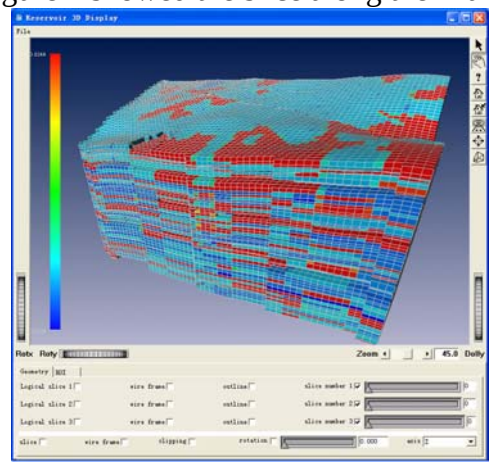

Fig.1 the reservoir model 3d display

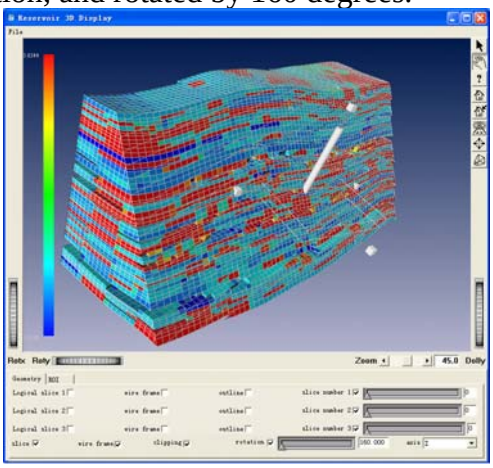

Fig.2 plane slices display of the reservoir model

Cell filtering

The MoCellFilter node stored a pointer to the object implementing the cell filter interface. It must be set in the scene graph before any data mapping node worked. The same node was used for storing all types of cell filter. This was done by providing one setCellFilter method for each type of mesh. The cells filtering and ROI setting could use the MoCellFilter object to set condition. Before using with the MoCellFilter object, the setCellFilter function must be called to specify the MiCellFilterIjk type parameters of object, which must achieve the acceptCell interface to set the filter conditions. Those cells not meeting the filter conditions would be set to the invalid cells, and meant the properties were not concerned, and would not display in the scene.

ROI setting

After eliminating all invalid cells in the scene, the remaining areas were the interested parts, the ROI setting key code were as follows:

```
inline bool ROIDataRangeFilter :: acceptCell (size_t i, size_t j, size_t k) const
{
  if (((int) i> = m_imin) && ((int) i <= m_imax) && ((int) j> = m_jmin) && ((int) j <= m_jmax)
&& ((int) k> = m_kmin ) && ((int) k <= m_kmax))
  {
    return (m_dataMin <= m_Mesh-> getDataSet () -> get (i, j, k) &&
            m_Mesh-> getDataSet () -> get (i, j, k) <= m_dataMax);
  }
  else
    return false;
}
```

The Figure 3 showed the region of interest of water saturation attribute parameter display of a certain reservoir model, the scope of the region were that 23-76 layers along the x-axis direction, 12-47 layers along the y-axis direction, 4-37 layer along the z-axis direction, and the proper cells were also selected whose data range were between 0.2 and 0.685.

Logical slice

Using the acceptCell interface to set the filters of x, y, z coordinates, when the cells were not in the target block, they meant be set as invalid. The mechanism could be employed to achieve the slice function along the three directions.The Figure 4 showed the slices of water saturation attribute parameter display in three directions, the scope of the region were layer 35 along the X-axis direction, layer 30 along the y-axis direction, and the layer 25 along the z-axis direction.

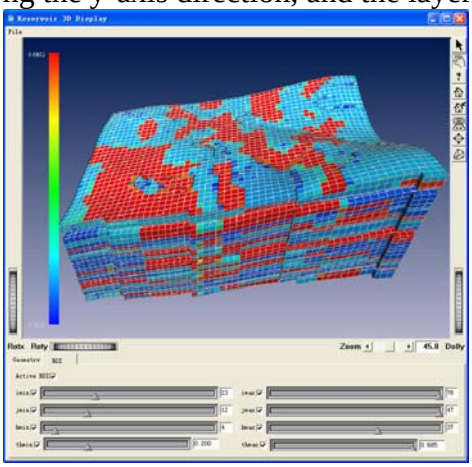

Fig. 3 ROI setting

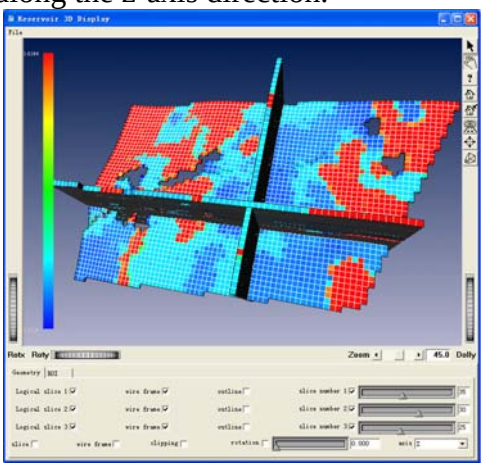

Fig.4 logical slices along the three directions

Summary

The interactive technology of reservoir three-dimensional display was studied based on the MeshViz and DialogViz extended module of Open Inventor. The proposed method could solve the problems of slices and cells filter in the process of reservoir three-dimensional display, and it would be the basis of an effective solution of the three-dimensional mesh data interaction.

References

[1] TENG Yi-jian,CHAI Shan,JING Xu,JIAO Xue-jian.Virtual driving simulation system based on Open Inventor[J].journal of Computer Applications,2009,29(Z1):323-325.

[2]YAN Feng-xin,HOU Zeng-xuan,ZHANG Ding-hua, etc. Open Inventor programming from entry to the master [M]. Beijing: Tsinghua University Press, 2007.

[3]. Open Inventor6 User’s Guide. Mercury Computer System Inc.2006.

[4]. Alan Ezust,Paul Ezust.An Introduction to Design Patterns in C++ with Qt4[M]. Prentice Hall,2006.

[5[. Josie Wernecke.The Inventor Mentor: Programming Object-Oriented 3D Graphics with Open Inventor,Release 2 [M],Addison-Wesley Publishing Company,1994.