Interactive Visualization of Oil Reservoir Data

Sang Yun Lee, Kwang-Wu Lee, Ulrich Neumann
University of Southern California

INTRODUCTION

Computer-based information systems have evolved for decades and are very popular in today's market. They are well equipped with intuitive and powerful features in terms of data retrieval and analysis. But, still, many users do not feel comfortable using them. Many empirical studies show evidence suggesting that it is not easy for inexperienced users or occasional users to get data of interest using computer-based information systems [8]. In our experience in an oil company project, we have observed similar problems at our customer's sites. Sharing existing information across organizations might be a significant burden since different domains use different tools and formats. Also, experts of a particular domain feel comfortable with their specialized tools, but general-purpose users such as mid/higher-level managers, financial/management personnel, and those who work in an operation control center in the organization feel uncomfortable manipulating these tools to extract the desired information.

We propose a visualization framework named Reservoir Model Information System (REMIS) as our first prototype based on to the following requirements from our project:

- Integrate oil reservoir related data sets with geographical information.
- Offer easy user interface/easy to understand system.
- Provide interactive visualization and data manipulation.
- Ultimately, provide a variety of visualization techniques so that a user can find correlation between data sets such as patterns, trends, and exceptions.

REMIS is intended for non-IT experts such as geologists, reservoir engineers, staffs, and mid/high-level decision makers in this field so that they can understand data and communicate between them in an easier way. To meet the requirements above, we designed REMIS into two parts: (1) the data access point module and (2) the data visualization module.

The data access point module is responsible for querying and acquiring data from data sources and visualizing data. This module is extended from our previous work, Phrase-Driven Grammar System (PDGS) [7], the thick-lined box in the middle of Figure 1. PDGS works as a middleware bridging between data sources and 3rd party visualization applications such as MS-Excel and Spotfire. By interfacing other visualization tools, a user can bring up a data visualization without knowing a specific usage of a visualization tool. For example, if a user issues a 2D plot visualization description, he can send that description over the connected 3rd party visualization tools such as MS-Excel and Spotfire and get the 2D plot visualization without knowing how to use MS-Excel and Spotfire. In addition, to support usability, expressibility, and dynamic interactivity of data visualization, PDGS adopts structured English-like grammar named Phrase-Driven Grammar (PDG). A data visualization description is expressed with relatively short and simple structured sentences. Also, each stage of constructing a sentence is guided by the GUI of the system step by step. A completed data visualization description is readable and understandable and can also be sent over to peers and be executed just like a scripting language in computer programming. Recipients can run and update the received sentences by themselves as well.

For the data visualization module, we implemented a new tool, REMVR, the dotted box in Figure 1. REMVR is not only an interactive 3D volume visualization tool but also a 3D geo-spatial information system, which means that all rendered objects are selectable data objects. If a user picks a rendered object in REMVR, then relevant data is queried and REMVR displays the result. REMVR is implemented on top of Simian [3], which is a volume rendering tool for scientific visualization and developed at the University of Utah in collaboration with the Advanced Computing Laboratory at LANL. It should be noted that REMVR is useful only with geo-location data. If the data does not come with geo-location specific information, then a user needs to add them by manual or in a pre-configured way.

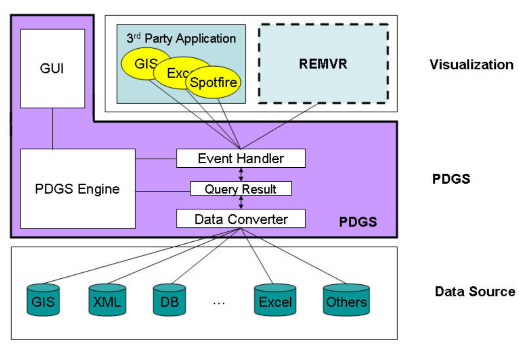

Figure 1. Architecture of REMIS

By integrating the two modules, PDGS and REMVR, REMIS provides a unified and powerful oil reservoir model information system. The main contributions of REMIS are in easy and effective oil reservoir visualization as follows:

- It enables users to create or update a visualization, which also can specify data to be associated with the visual object using PDG.

- Each process in constructing a PDG visualization description is guided in a user-friendly way so that users begin to work with little training.

- Users can visualize a data set without knowing how to use specific visualization applications.

Let us take an example of the benefits of REMIS. A user wants to see a 3D volume of a geological property. When the user displays this data set on REMVR, he thinks that some parts of the data set are not correct due to some reasons. So, he wants to correct it and visualize it again. REMVR can help the user to do this process. The user selects area of interest in REMVR. Then, the data in the selected area is stored as a temporary object in REMVR and shared with PDGS. Now, the user calls and open up the whole 3D volume data set and finds the data of the selected area in REMVR. The user calls a text editor, makes updates for data of interest to the 3D volume data set, and pushes it again to REMVR through PDGS.

Another example is that a user wants to see a 2D plot of time vs. oil production for each well in a selected region with well name and associated rock properties at the same time to see the correlation between oil production and associated rock property for each well in the region. The user can use Trellis visualization through PDGS to see a 2D plot visualization for each well of the region in one chart and, in the mean time, he brings up REMVR to display 3D rock properties for wells in the selected region. Now, the user can see 2D plot of time vs. oil for each well and its associated rock properties using two different data visualization tools. The main advantage here is that the user does not know what Trellis visualization is and how to use 3D volume viewer specifically, but by just triggering commands in PDGS, he can get visualization of interest.

Our challenge here is to achieve an optimal balance between usability and expressive power in visualization. Our policy is to make most frequently used ones and common cases work in our tool and leave the complicated query or visualization to domain expert systems. For this, we have been closely communicating with our customer, geoscientists and petroleum engineers, and have reflected their inputs iteratively. In this paper, we focus on showing how to easily and effectively explore major oil reservoir data sets (focus on usability, flexibility, extensibility, and interactivity).

DATA ACCESS POINT MODULE: PDGS

We add a set of resource types, visualization types, and instructions to PDGS from our previous work [7], so that users can create a variety of visualization in REMVR.

DATA VISUALIZATION MODULE: REMVR

In this section, we introduce the data visualization module, REMVR. REMVR supports two existing rendering techniques: direct volume rendering and OpenGL primitive-based rendering. Currently, the direct volume rendering method is only used for rock properties and general reservoir sub-surface properties such as oil saturation and water saturation. These properties are regular grid volume data sets and each consists of a set of floats or integers in a 3D array. We call this type of data, reservoir property data. All other visual objects, in general, are rendered as OpenGL primitives unless they are specified as 3D volume property data sets beforehand.

We focus on data visualization related to reservoir property data and how we approached it. In general, if a user displays reservoir property data, then there is not enough information for the user to learn from it. They do not contain specific boundaries or shapes inside. For example, a bunch of points or blurred shapes are displayed all over the volume. To solve this problem, we use several techniques to give users possible shapes and boundaries inside a geological volume data: (1) voxelization, (2) gradient segmentation and k-mean clustering techniques, and (3) radial basis functions (RBFs) to visualize geological objects out of sparse sampling geometry data by taking an example of geological layers [1, 2]. Also, we support a transfer function as a widget, with which a user can assign colors to boundaries and shapes interactively by specifying data ranges with colors [3, 4]. Its width and height mean the normalized scalar data range from 0 to 255 and the opacity of the color to be used for color coding, respectively.

DATA VISUALIZATION AND EXPLORATION

In this section, we show several examples of what REMIS can do in terms of data exploration and visualization. We assume that "Reservoir_LH" is a reservoir data object, which has all data sets including data objects and visual objects related to it. REMVR supports visualization of a 3D volume box to display oil reservoir property data using the visualization type called, "3D Volume". "Geometry" is used to specify whether a given volume data needs to be voxelized with geometry information or not by REMVR. The complete example of the PDG sentence is:

*For Reservoir_LH with **Geometry**, show **3D Volume** of Oil Saturation.*

The PDG sentence above displays oil saturation data in Reservoir_LH as a 3D volume, but since we do not assign color, REMVR uses its default color scheme. A user can assign colors to a specified data range of the volume data interactively. REMVR has the default color coding type called Rainbow Spectrum and it offers two data range modes: the full data range mode called "full data range" and the valid data range mode called "valid data range". The term "full data range" refers to the full normalized range of all data values (e.g., 0 ~ 255) while the term "valid data range" means the range which only has non-zero values. Unless it is specified, the valid data range mode is applied automatically. Figure 2 shows the result of oil saturation visualization with Rainbow_Spectrum with the valid data range and the relevant PDG sentence is as follows:

Show Vis {

For Reservoir_LH with Geometry,

show 3D Volume of Oil Saturation.

For the Previous Result,

show Rainbow_Spectrum for valid data range.

}

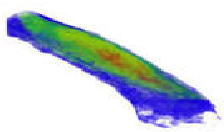

Figure 2. Visualization of a Voxelized Oil Saturation.

Several visualizations can be merged into one visualization using a special type of visualization. This visualization type is called, “Visualization” or “Vis”. A user can create a visualization of an oil saturation data as a 3D volume with well objects, their pipes, and the amount of water injection rate by issuing the following PDG sentence:

Show Vis {

For Reservoir_LH with Geometry,

show 3D Volume of Oil Saturation.

Show Cube of Well.

Show Line of Well Pipe.

Show Cylinder of Water Injection.

}

The PDG example above shows how multiple sentences can merge into one sentence. There are four sentences in parenthesis. The last three sentences do not have Data Source part, which starts with “For”. When Data Source part is abbreviated, it implies that it would use the result of the previous sentence as a data source. Figure 3 shows the result of the PDG statement.

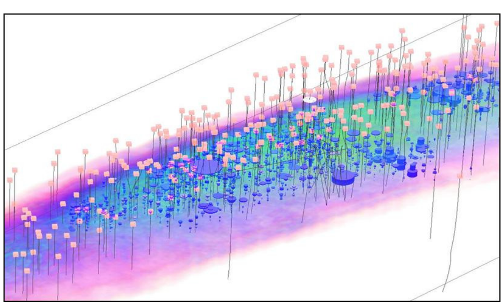

Figure 3. Visualization of a Oil Saturation with Well and its Water Injection rate.

The next examples below describe how to visualize a specific region of a reservoir model with geological and topological constraints. Figure 4 can be described in PDG as follows:

Show Vis {

For Section_21 and Layer_2 in Reservoir_LH,

show 3D Volume of Oil Saturation.

Show Cube of Well.

Show Line of Well Pipe.

Show Cylinder of Water Injection.

}

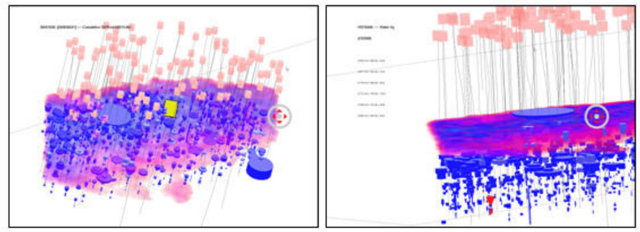

Figure 4. Visualization of Section and Layer

We visualize the oil saturation as a volume in the picture on the left image of Figure 5. But, boundaries of the volume are not clear enough to distinguish, especially in the voxel level. Now, the user wants clearer boundaries and shapes. We use a general spatial clustering approach. For example, to cluster the oil saturation volume in Section 32 and Layer 2 into 3 clusters, the following statement can be used and the clustered result is the picture on the right of Figure 5:

Show Vis {

For Section_21 and Layer_2 in Reservoir_LH,

show 3D Volume of Oil Saturation.

For the Previous Result,

show 3-Clustered of Oil Saturation.

}

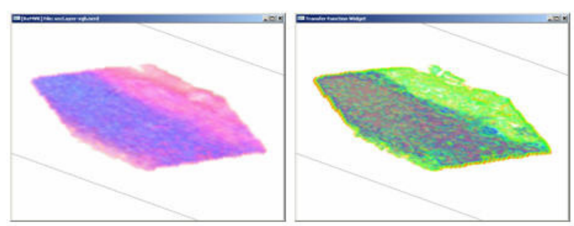

Figure 5. Direct Oil Saturation Clustering (right).

Figure 6 shows an example of a customized new visualization named “Well IO Web”. The Well IO Web is designed to represents the relationship between one input and multiple outputs in terms of oil production in one well pattern. One input well affects output wells in a well pattern. In our case, “input” means “water injection” and “output” means oil production” and the term “well pattern “ here is our own definition to divide a reservoir into smaller regions. Using a modified Delaunay triangulation method, we constructed a polygon mesh, which maximize the number of polygon, which contains one input well near the center of a polygon and multiple output wells around the input well and then for each well pattern, we pull out all input and output data of each well.

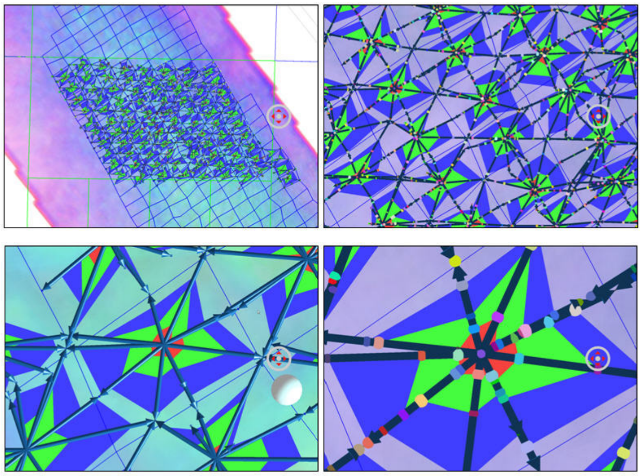

Figure 6. Visualization of Well IO Web.

What we wanted to visualize are: (1) which output well is affected first after water injection in a certain well pattern, (2) which output well is affected most, and (3) whether there is any anomaly. We provide the following PDG grammar for Well IO Web visualization:

For a given reservoir data source,

show Well IO-Web of ***[input well]*** *vs.* ***[output wells]***

with ***[threshold], [time interval]*** *over* ***[time period].***

Each bracketed element in this PDG grammar is a user input parameter. For each well pattern, we draw arrows from the input well in the center to multiple output wells and based on all input and output data over the specified time period, REMVR constructs a normalize time scale which is smaller than the shortest arrow in the well pattern. Based on user parameters, REMVR calculates time points, at which oil production value crosses its given threshold in a given time interval and a period of time for each output well. And, then, it connects corresponding time points of each well and generates an area with the same color. For each time points in the same colored area, it displays relative production changes using the bubble and its size.

In short, each arrow represents a normalized relative time starting from the input to output well. Since it is normalized, a user can perceive which well is affected first by checking the length of the color coded area. The same color area means the same time period for all well. The size of bubbles in the outside boundary of the red colored area means relative oil production to show which oil production well is affected most by a given water injection well. The biggest one in the same color region means the most affected one in that period of time. By changing user parameters interactively and repeatedly, REMIS can help users to explore data in a reservoir model and to find out a relationship between water injection and oil production.

CONCLUSION

REMIS is a geo-spatial information system for oil reservoir data. It provides an easy and flexible user interface for inexperienced and occasional users to get, manipulate, visualize, and explore data interactively and intuitively. It is designed based on the requirements from oil-field engineers and geo-scientists in the project. The teams have used scripting languages to gather information from other data sources and visualized in their own ways, but, in general, since many team members do not know about programming languages and database structure concepts, they depend on IT experts' helps in many cases. REMIS enables team members to access and visualize data without any intensive training. Thus, inter-domain communication problems due to different types of software tools are alleviated by use of unifying structured-language-like interface and flexible framework, which bridges external visualization tools into one consistent control point. Also, it provides ways of making customized visualizations by combing primitive visual objects together.

ACKNWLEGEMENT

This work is sponsored by CiSoft/Chevron.

REFERENCES

1. Buhmann J. M., Fellner D. W., Held M., Ketterer J., Puzicha J. Dithered color quantization, In Proc. of Eurographics'98, vol. 17(3), pages 219-231, 1988.
2. Bookstein F. L. Principal warps: Thin-plate splines and the decomposition of deformations, IEEE Trans. Pattern Analysis Machine Intelligence, vol. 11(6), pages 567-585, 1989.
3. Kniss J., Kindlmann G., Hansen C. Interactive Volume Rendering using Multi-dimensional transfer functions and direct manipulation widgets, Visualization, pages 255-562, 2001.
4. Levoy M. Display of Surfaces from Volume Data, IEEE Computer Graphics and Applications, vol. 8(3), pages 29-37, 1988.
5. Pat Hanrahan. Tableau Software White Paper - Visual Thinking for Business Intelligence, Tableau Software, Seattle, WA, 2003.
6. Pouliot J., Bedard K., Kirkwood D., Lachance B. Reasoning about geological space: Coupling 3D GeoModels and topological queries as an aid to spatial data selection. Comput. Geosci., vol. 34(5), pages 529-541, 2008.
7. S Y. Lee, U. Neumann. A Phrase-Driven Grammar System for Interactive Data Visualization, Visualization and Data Analysis, Proc. SPIE Int. Soc. Opt. Eng. Vol. 6809, pp. 68090K, Jan 2008.
8. Thiebaux M., Tangmunarunkit H., Czajkwski K., Kesselman C. Scalable Grid-based Visualization Framework, Technical report ISI-TR-2004-592, USC/Information Science Institute, June 2004.