

Research Article

Cross-Platform Drilling 3D Visualization System Based on WebGL

Shanshan Liu ¹, Yueli Feng ^{1,2}, Xiaoqiu Wang ¹, and Pengyin Yan ¹

¹College of Petroleum Engineering, China University of Petroleum, Beijing 102249, China

²State Key Laboratory of Petroleum Resources and Prospecting, China University of Petroleum, Beijing, China

Correspondence should be addressed to Xiaoqiu Wang; 13466396559@126.com

Received 10 January 2021; Revised 23 February 2021; Accepted 23 April 2021; Published 5 May 2021

Academic Editor: Teen-Hang Meen

Copyright © 2021 Shanshan Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This study develops a novel drilling 3D visualization solution based on WebGL, termed as WebDrillingViz, and introduces the system architecture design and software programming implementation. The software is part of the Engineering Technology Internet of Things (IoT) System, interfacing with other software, and also capable of direct hardware interfacing for data retrieval and system control. It is fully web-based, used real time, and used in RTOC (Real-Time Operating Center) of IoT system, which is a software system for drilling process remote monitor and decision. WebDrillingViz uses the most frontier HTML5 technology to realize a brand-new drilling 3D visualization system. The front end is designed in single-page application (SPA) mode and adopts technologies such as angular, bootstrap, and WebGL. The front-end uses single page application (SPA) mode, Angular, Bootstrap, WebGL and other technologies are used. The back-end data services provide data interface support for front-end visualization applications based on HTTP protocol which uses NodeJS, a lightweight development platform suitable for cloud platform, and Restify to realize a REST JSON API. Both sides are using the same object-oriented development language—TypeScript. The front-end develops an easy-to-extend 3D visualization class library based on WebGL for drilling. It is encapsulated as Angular modularization to form an Angular component, which can be used standalone or integrated into other Angular applications. At the same time, the back-end microservice architecture combined with container and cloud technology is easy to maintain, deploy, and expand and has the advantages of being lightweight, cross-platform, flexible, and efficient. Using HTML5 standard and Bootstrap's responsive layout achieves cross-platform, which can support different operating systems and screen sizes. The system has better robustness and maintainability, thanks to the object-oriented and strong typing characteristics of TypeScript. Practical application shows that WebDrillingViz is efficient, capable of visualization of large drilling 3D scene, and compatible with mainstream devices, such as Windows, Linux, macOS, iOS, and Android. The use of open standards-based modern web technologies and data format enables a more lightweight and economical solution. WebGL, Angular, NodeJS, and TypeScript formed a powerful technology stack, which can be used as an excellent reference for other browser-based visualization development.

1. Introduction

In November 2005, the International Telecommunication Union (ITU) released a report entitled “the Internet of things,” which formally puts forward the Internet of things (IOT), which attracted the attention of governments and industries. Petroleum industry is an important industry of IOT application, which has been applied in some enterprises. The Internet of things system of engineering technology includes the automatic data acquisition, remote data transmission, data center, and remote operation support center (RTOC). Through the remote operation support center, technical experts can view the data and video

information of the well pad on the computer of the center for remote analysis and decision-making. RTOC aims at building data acquisition and application, a multiprofessional collaborative service platform to improve the integrated service capabilities, realize intelligent operation support, the integration of engineering and technology wellbore business, and information technology [1–3]. RTOC includes the following parts: automatic collection, integrated storage, and remote transmission of field data; real-time data detection, early warning and analysis optimization; remote technical support; and decision-making. It provides a powerful guarantee for making and transmitting drilling decision quickly and effectively, as shown in Figure 1.

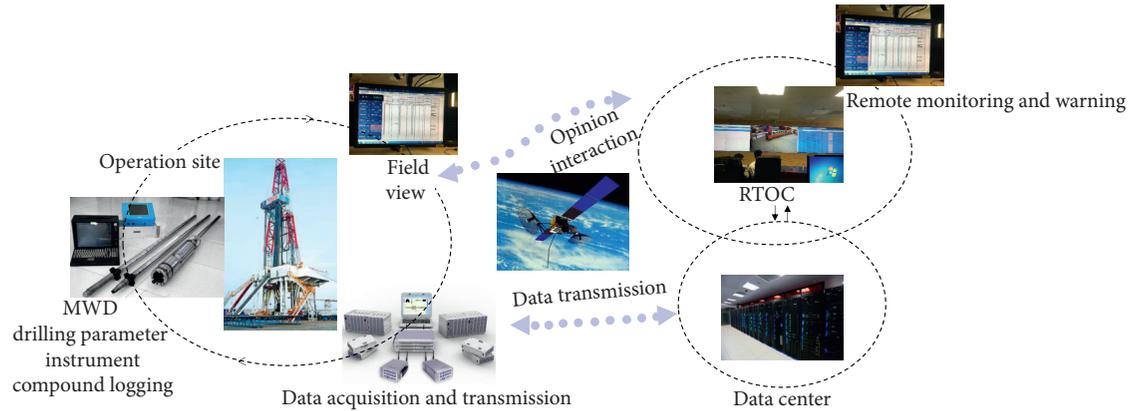


FIGURE 1: Remote operation center architecture.

Drilling 3D visualization system is an important task for building web applications in ROTC, which provides a highly visualized working environment to view real-time 3D drilling scene for drilling process remote monitor and decision. It can put borehole trajectories, LWD, MWD, mud logging, well-logging data, seismic slice, and formation interfaces, reservoir geological model all into a 3D scene, realizing the information sharing of logging, geology, and engineering; thus, experts can get realistic underground scenes. Cross-platform, lightweighted, and open are three basic rules when building drilling 3D visualization systems. Currently, some efforts have been made on drilling 3D visualization; however, most of them are traditional desktop software which cannot provide online service for ROTC. At present, some companies built such visualization products: 3D visualization software enables drilling engineers, geophysical engineers, geological engineers, and reservoir engineers to work with the same data volume environment, breaking professional and geographical constraints, and making multidisciplinary [4–9]. Combination really becomes a reality. Most of the existing 3D visualization technologies adopt the client-server (C/S) mode, such as DecisionSpace®, and WellViz3D are released as standalone software, which must be installed and used in a local operation system (OS). Discovery Web of Kongsberg adopts the B/S mode. However, based on ActiveX plug-in technology, special plug-ins need to be installed, which can only support IE browser on PC Windows and cannot support mobile devices (Android and IOS) [10]. Undoubtedly, a web-based drilling 3D visualization system has flexibility to perform real-time monitoring of drilling activities through a browser platform in ROTC. However, there is no 3D visualization technology of drilling based on standard browser at present. A cross-platform online drilling 3D visualization system to describe the underground geological environment and drilling state using WebGL (termed as WebDrillingViz) based on HTML5 and WebGL standard is developed in this paper, which can support all mainstream operating systems and browsers. By using standards-based technologies, this visualization can operate on a multitude of devices removing both the technical and logistical restraints on a remote collaboration. The online drilling 3D visualization system is

different from that on local drilling 3D visualization in the following aspects:

- (i) Lightweight: HTML5 and WebGL enable the browsers realize excellent 3D effects without any plugins, the REST server built with Node.js and Restify in microservice architecture, which can easily be deployed on cloud.
- (ii) Cross-platform: rich, interactive, browser-based, cross-platform visualization tools can serve both PCs and mobile devices that run different OSs, which support WebGL, such as Windows, Linux, IOS, and Android.
- (iii) Open: the software supports the data formats commonly used in oil and gas industry and can easily import geological and related data into the system's database.
- (iv) Economic: the software is all based on HTML5 and open-source technology stack, which is not bundled with expensive software packages and has no tedious licensing restrictions.

2. 3D Visualization Technology on the Web

The idea is to update the experts of ROTC with live data, i.e., detailed 3D real-time insights into the ongoing drilling operation. In the past few years, a great progress has been made in modern web browsers and network standards [11, 12]. Cross-browser and cross-platform standardization greatly simplifies the design and implementation of rich interactive Web applications. In the past, third-party plug-ins needed to realize complex interaction or graphics in browsers. Java applets, Flash, and Silverlight were used for browser-based rich and interactive applications. Many security issues, as well as poor support and performance on mobile devices, made these technologies gradually abandoned. Instead, a powerful set of Web standards (commonly known as HTML5) has evolved into a framework on which real browser-based and cross-platform visualization tools can be developed. Special interest in drilling and geological visualization is HTML5, HTML5 canvas, and WebGL.

2.1. HTML5. HTML5 is a markup language used for structuring and presenting content on the web pages. It is the fifth and current major version of the HTML standard, and it was developed by World Wide Web Consortium (W3C, a broad coalition of organizations).

Compared with the previous version, the 5th version has made a qualitative leap in processing graphics and images. On the premise of not relying on third-party plug-ins, a new standard for graphics and image applications is proposed, which supports dynamic display and interaction on various mobile platforms and further improves the flexibility and security of web applications. After discarding all kinds of plug-ins, it combined with JavaScript makes the expansion of website functions easier to achieve and the development of website more efficient and safer.

2.2. HTML5 Canvas. The <canvas/> HTML tag provides a container that programmatically draws graphical objects on the screen. In short, a canvas tag is similar to an image but provides the ability to draw raster-based dynamic graphics. The canvas tag has been existing for many years and was eventually integrated into the HTML5 specification. At present, all the major browsers and devices platforms have implemented canvas tag, which makes it a good and feasible way to draw dynamic drilling scenarios. Although the canvas is essentially a 2D space, 3D effects can be simulated in 3D games. The rendering engine calculates the perspective on the 3D object and maps it to the 2D view for display on a flat screen. No matter what the underlying technology is, this general method is suitable for graphic rendering of all 3D objects in the 2D environment. The canvas tag simply provides a standards-based and well-supported tool for rendering dynamic graphics in Web browsers.

2.3. WebGL. WebGL has been introduced as one of the powerful web features for developing 3D content, and it is based on OpenGL ES 2.0 and allows rendering 3D scene in the browser. WebGL provides a general interface for accessing the 3D graphics hardware of the underlying system in the browser and can realize the rendering of complex 3D scene efficiently. Currently, all major browsers provide support for WebGL, making it the ideal technology to render 3D drilling and geology scene across platform and devices.

3. System Architecture

The system was built on HTML5 standards and open-source frameworks.

3.1. Overall Framework. Under the environment of HTML5, the system adopts the B/S architecture building a network platform using the TypeScript development language. With the help of the language's full object-oriented and strong typing characteristics, the code has strong expansibility, reusability, security, robustness, and stability. The system uses an object-oriented design method, builds a scalable 3D

visualization class library, and encapsulates it as an Angular module, which has strong reusability and inheritance.

Through Angular's powerful compile ability, the compiled program can run efficiently in the browser. It accesses back-end RESTful data interface API to obtain user data and provides a friendly user interface by using the responsive layout, which can support both desktop and mobile systems.

The software realizes the unified management of visual objects, displays the coordinates of 3D scene, and performs the fundamental functions including zooming, rotating, and translating. Figure 2 illustrates software architecture including data layer, service layer, and view layer.

The data layer is Geological Engineering Warehouse, and it uses real-time and relational databases to receive and preserve static and dynamic drilling geological data. Static data include oilfield information, adjacent well data, drilling design data, formation tops, seismic slices, and other geological engineering data. Dynamic data include drilling, MWD, LWD, and mud logging data.

The service layer based on Node.js and Restify framework provides a RESTful API. It is the data interface between front-end application and the back-end data warehouse. All communications between the data warehouse and the visualization program will also be performed through the servers.

The view layer is a single page Web application (SPA), based on HTML5 standards and TypeScript language, using Angular front-end framework and Bootstrap UI library to realize responsive SPA. Through the WebGL interface of browser, 3D visualization components are implemented to realize web-based cross-platform interactive 3D visualization application for drilling and geological.

3.2. Technology Stack. This software uses a series of open-source technologies to build an economical and efficient technology stack.

3.2.1. TypeScript. TypeScript is an open-source programming language developed and maintained by Microsoft [13]. It is a strict syntactical superset of JavaScript and adds optional static typing to the language and offers a module system, classes, interfaces, and a rich gradual-type system. TypeScript developers sought a solution that would not break compatibility with the standard and its cross-platform support. TypeScript may be used to develop applications for both client-side and server-side execution. The client and server of this software are all developed with TypeScript.

3.2.2. Angular. Single Page Application (SPA) was built on expanding reach via the browser, reducing round-tripping, and enhancing User Experience (UX). Compared with traditional web applications, SPA application loads the html, CSS, and JavaScript programs of a single page at one time and dynamically updates the content when the user interacts with it. It enables Web application to realize complex dynamic interaction in one page. Advanced SPA application framework will compile, package, and compress all source

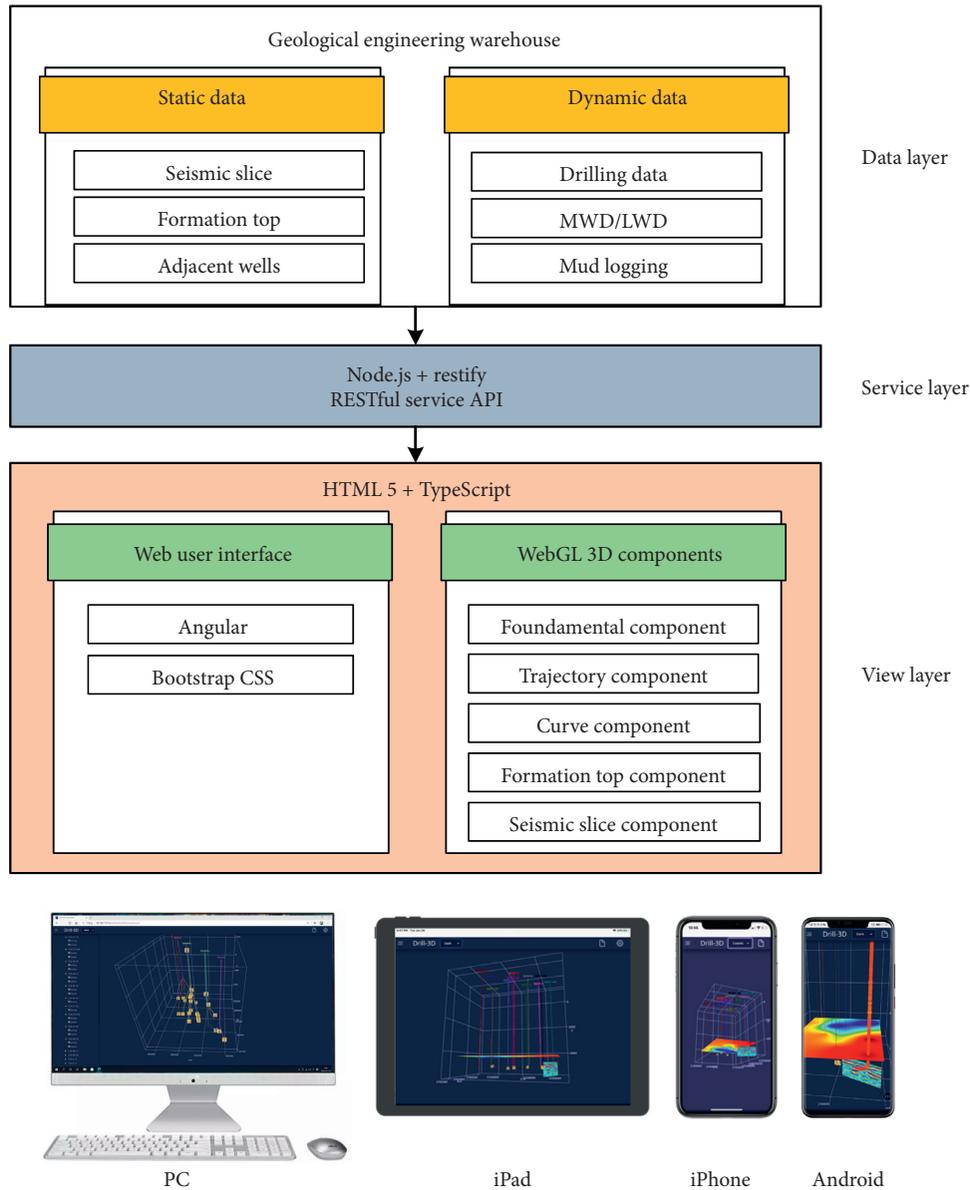


FIGURE 2: System architecture.

codes. The final application has only a small number of compiled, optimized, and compressed .js files. The execution efficiency is much higher than the traditional way through the browser one-time loading. The core of SPA technology is the combination of front-end technology and back-end REST data service. The interactive operation of the application is completed through JavaScript programs in the browser without the participation of the server. The server only provides the support of data services, so it can get closer to the interactive effect of desktop C/S applications.

A clear picture of typical SPA architecture contains a server and client, as shown in Figure 3. Angular has become an increasingly popular choice for rapid development of dynamic HTML pages [14]. The major front-end technologies are Angular, VUE, and React. Angular, an open-source TypeScript framework, has been developed to enable and

give an extreme freedom to client-side developers building powerful SPA. This software adopts the MVVM architecture and latest version 7.0 of Angular to develop.

3.2.3. Responsive Web Design. Although Angular implements SPA, it mainly solves the problems of data binding and page interaction. At the UI level, it needs to use additional framework. Responsive web design (RWD) is a method for web page construction to detect the user's screen size and orientation and dynamically change the layout accordingly for multiple devices [15], so the site produces the output, which is viewable and navigable with the devices and web software of the intended site users. It employs the use of flexible layouts (columns), scalable images, and CSS media queries. Thus, responsive web pages display their elements

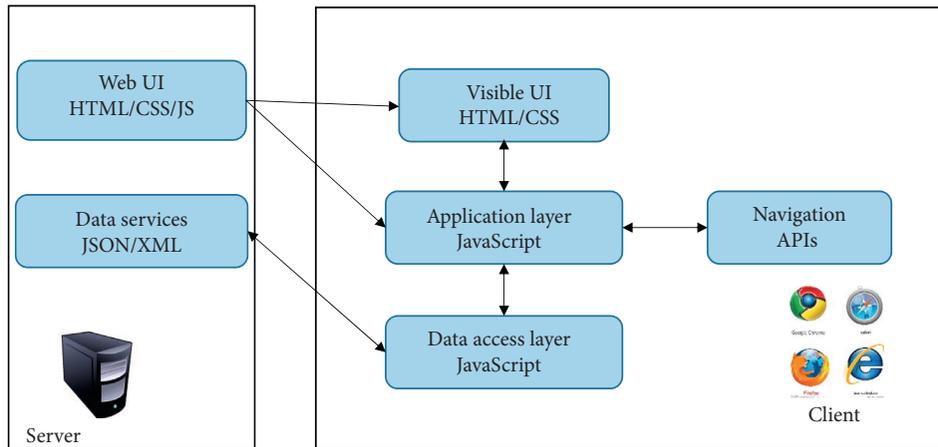


FIGURE 3: High-level architecture of SPA.

differently for different screen sizes. They change the sizes, shapes, and arrangements of their elements or make some elements invisible on small-sized screens using DOM APIs. An adaptive layout system for different platforms (devices) is needed in front-end page layout of the software, as is shown in Figure 4. Each of these devices has unique display dimensions that Web elements will adapt to, maintaining a consistent user experience.

3.2.4. Three.js. WebGL provides an API to create hardware-accelerate 3D graphics program, but we need to have an in-depth knowledge of how WebGL works internally to work with the API. Luckily, there are several JavaScript libraries available that hide the complexity of WebGL and provide an easy-to-use API to create 3D applications. Currently, the best of these libraries is Three.js, and it is a third-party open-source library of WebGL written in JavaScript, which provides a lightweight and easy-to-use 3D class library.

3.2.5. Restify. Restify is a middleware framework for developing REST-style APIs, which enable rapid development of robust REST service interfaces. Restify is used by some of the industry's most respected companies to power some of the largest deployments of Node.js on planet Earth.

3.3. 3D Visualization Components. According to the type of information to be visualized, corresponding components are developed to realize detailed display of well, borehole, real-time data and geological model, which makes visualization more useful for drilling experts, for example, the gamma ray curve related to formation lithology; the curve component can show the curve along the wellbore, and the change in formation lithology can be intuitively displayed, as shown in Figure 5.

The main 3D components include 3D visualization fundamental component, well-trajectory component, curve

component, formation tops component, seismic slice component, and reservoir model component.

3.3.1. Fundamental Component. The fundamental components are the most important part of the whole software. In addition to realizing the basic functions of 3D drawing, including zooming, rotation, illumination, object management, and coordinate display of graphics, an extensible 3D drawing framework is realized by using object-oriented technology. All other 3D drawing components are extended on this basis, and new components can be added any time according to the requirements, as shown in Figure 6.

Axis box (AxeBox class) is the most important foundation component, it can display X, Y, and Z grids and labels, and as a container manages all other 3D components, it can dynamically change label's position and always displays the labels at appropriate positions.

There are eight color-map classes implemented IColorMap interface, and they can produce eight different color maps. It is also very easy to add new color-map class into software, as shown in Figure 7.

The basic component is responsible for the interaction with the mouse, realizing the translation, rotation, and scaling operations of the three-dimensional scene, depth alignment, coordinate system drawing, and graphics object management. Rotation, translation, and scaling are three basic operations for spatial three-dimensional objects, which enable users to view drilling three-dimensional scenes from different angles and positions. Although formulas can be used to represent each transformation, the formulas can become very complex when representing multiple continuous transformations. Using the coordinate transformation matrix, various basic transformations and combinational transformations can be realized more conveniently.

(1) Basic transformation of three-dimensional graphics

Rotation: spatial rotation can be decomposed into two-dimensional rotation around three coordinate axes. The

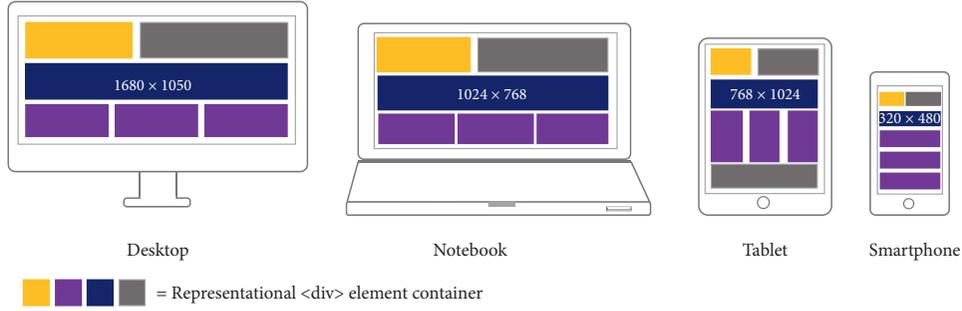


FIGURE 4: Responsive web design for multiple devices.

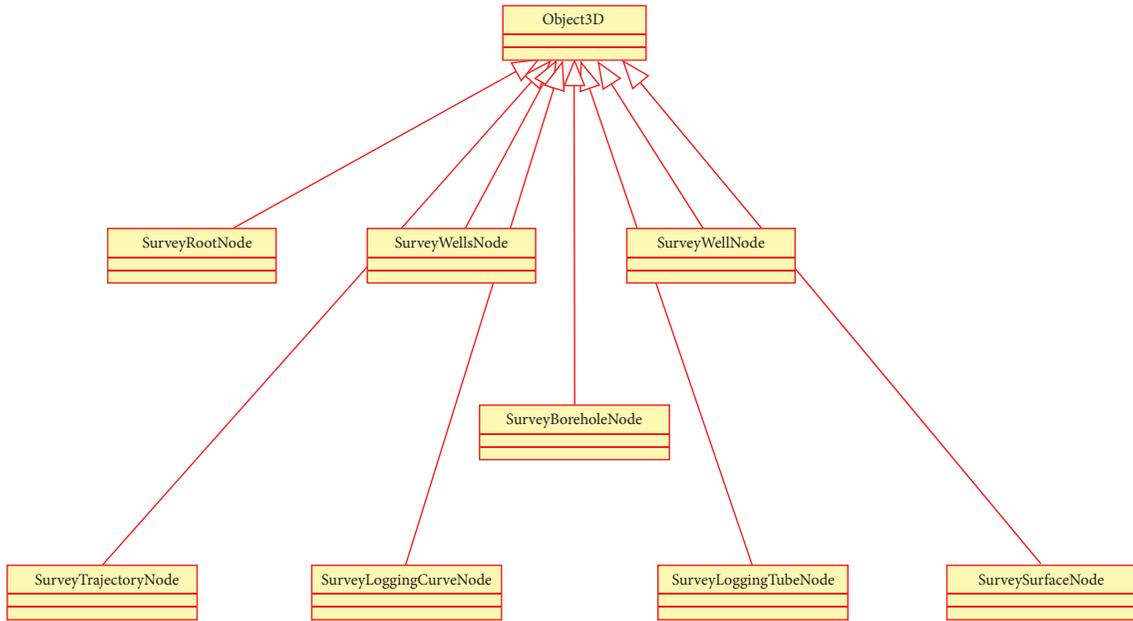


FIGURE 5: Class diagram of 3D visualization components for drilling.

rotation angles ψ , φ , and θ around the X , Y , and Z axes are transformed into

$$\begin{aligned}
 R_x(\psi) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \psi & \sin \psi \\ 0 & -\sin \psi & \cos \psi \end{bmatrix}, \\
 R_y(\varphi) &= \begin{bmatrix} \cos \varphi & 0 & -\sin \varphi \\ 0 & 1 & 0 \\ \sin \varphi & 0 & \cos \varphi \end{bmatrix}, \\
 R_z(\theta) &= \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}.
 \end{aligned} \quad (1)$$

The complete transformation is

$$R = R_x R_y R_z. \quad (2)$$

Translation: spatial translation is the movement of an object in any distance and direction. The transformation of the moving distances t_x , t_y , and t_z is as follows:

$$\begin{pmatrix} x' & y' & z' & 1 \end{pmatrix} = \begin{pmatrix} x & y & z & 1 \end{pmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t_x & t_y & t_z & 1 \end{bmatrix}. \quad (3)$$

Scaling: scale the object. In the direction of X , Y , and Z , the scaling factors s_x , s_y , and s_z are converted into

$$\begin{pmatrix} x' & y' & z' & 1 \end{pmatrix} = \begin{pmatrix} s_x x & s_y y & s_z z & 1 \end{pmatrix} \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (4)$$

Based on the above basic transformation, any complex three-dimensional transformation can be decomposed into a combination of three basic transformations; for example, the object is rotated around any point (x, y, z) in space. The transformation steps are as follows: first, the center point (x, y, z) is translated to the origin $(0, 0, 0)$ and then rotated around the origin, and finally, the origin is translated to the

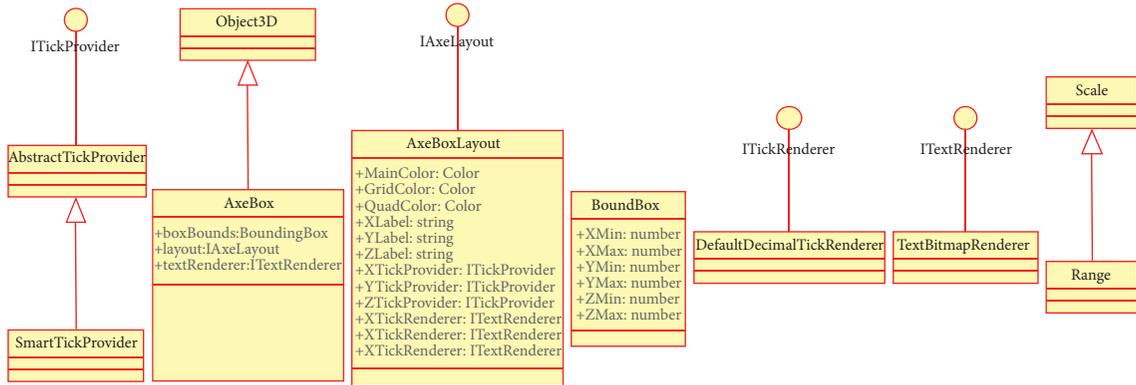


FIGURE 6: Class diagram of 3D visualization fundamental components.

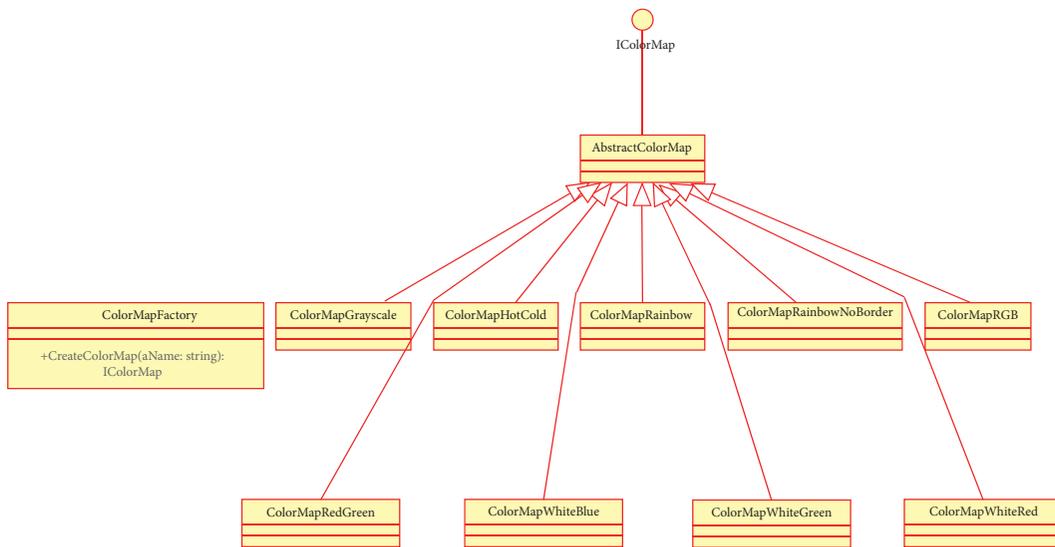


FIGURE 7: Color-maps classes diagram.

center point (x, y, z) . The final transformation matrix can be obtained by multiplying the matrices in the transformation process. In practical application, only the final matrix is calculated, and the complex process in the middle is not concerned:

$$M = T(-x, -y, -z)R(\psi, \varphi, \theta)T(x, y, z),$$

$$\begin{pmatrix} x' & y' & z' & 1 \end{pmatrix} = \begin{pmatrix} x & y & z & 1 \end{pmatrix} M. \quad (5)$$

(2) Depth alignment

The NS, EW, and TVD for a single well are relative to the wellhead. For different wells, the map-north, map-east, and elevation of the well are different. If multiwell data are placed in the same coordinate system, all wellhead data must be corrected to a unified coordinate system. The coordinate Z

takes the sea level as zero and the upward direction as positive:

$$(u_E, u_N, u_z) = (W_E + W_i, W_N + N_i, W_{bh} - TVD_i), \quad (6)$$

where (u_E, u_N, u_z) are the uniform coordinates, m; (E_i, N_i, TVD_i) are well data coordinates, m; W_E are eastern coordinates of wellhead, m; and W_N are borehole north coordinates, m.

(3) Perspective transformation and mouse interaction

The nature of perspective is the same as photographic imaging. It is projecting a 3D shape onto a 2D plane. Perspective is very close to human vision; for example, when the human eye looks at a road, the road in the distance looks narrower. The basic principles of perspective are shown in Figure 8.

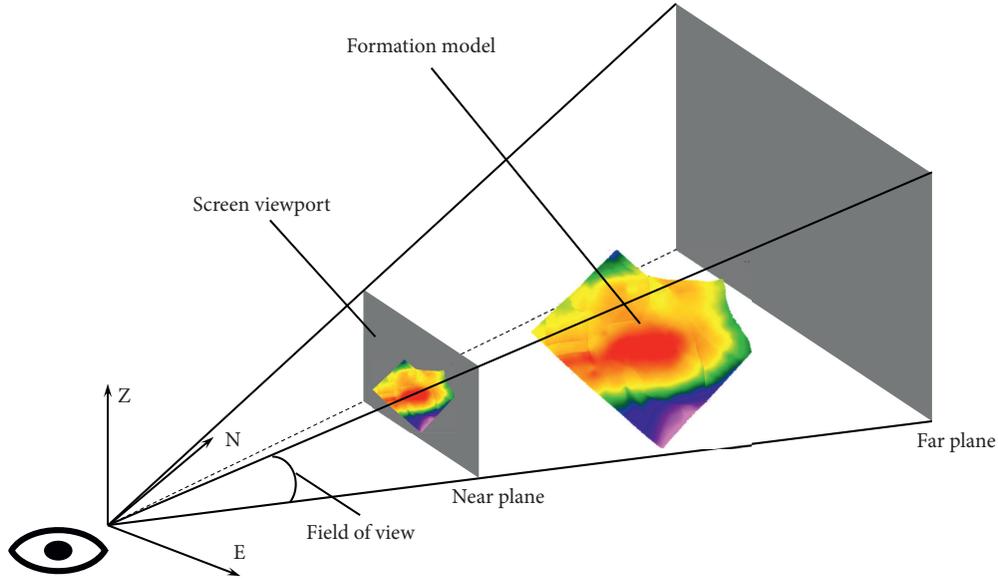


FIGURE 8: Perspective schematic diagram.

Through perspective transformation, the 3D graphics are presented on the 2D screen. When users want to observe the 3D scene with different distances and angles, they can move the camera position through the mouse. Mouse movement is a 2D action, so it is necessary to unproject the movement of the mouse to the 2D space according to the position of the camera. The users can achieve the effect of rotation and scaling by moving the position of the camera.

The software can trace the position of the camera in the process of mouse movement and dynamically adjust the display content; for example, in the coordinate system, only displaying three grid-faces of the axe-box located far-end can achieve better display effect. By calculating the distance between six planes of the axe-box and the camera, three distant planes are selected to display. The relative relationship between the ticks' text and the sight on the screen is calculated to reasonably determine the position of the calibration text and obtain better visual effect.

3.3.2. Trajectory Component. The trajectory component is used to display the design and drilling trajectories of the wellbore including static and real-time data.

The calculation of the trajectory is the basis of the well visualization. The track data, returned by the measurement tool mainly including measure depth (MD), deviation angle (Dev), azimuth angle (Azi), are shown in Table 1. The space coordinates of the trajectory are obtained by calculation.

The spatial coordinate parameters of the trajectory are defined as follows: North/South coordinate (NS), East/West coordinate (EW), and the true vertical depth (TVD), which represent the displacement of a point on the trajectory relative to the North/South, East/West, and the vertical direction of the wellhead, respectively, corresponding to the

TABLE 1: Trajectory measurement data.

MD (m)	Inc (deg)	Azi (deg)
0	0	0
153.32	0.45	3.21
180.8	0.39	21.9
208.48	0.46	26.04
...

Y, X, and Z axes of the direct coordinate system, in which the North, East, and Bottom directions are positive; for example, if the vertical depth of a well is 1000 m, the TVD of the wellhead is 0 and the TVD of the bottom hole is 1000.

The calculation process is as follows: the wellhead is taken as the origin, where $(TVD, NS, EW) = (0, 0, 0)$, starting from the wellhead, in turn, recursive calculation of two adjacent points as $(L_1, \alpha_1, \varphi_1, D_1, N_1, E_1)$, $(L_2, \alpha_2, \varphi_2, D_2, N_2, E_2)$.

By assuming the 3D curve shape of wellbore section between two measuring points, the vertical increment ΔD , North/South increment ΔN , and East/West increment ΔE of measured section are calculated and then add them up:

$$\begin{aligned}
 D_2 &= D_1 + \Delta D, \\
 N_2 &= N_1 + \Delta N, \\
 E_2 &= E_1 + \Delta E.
 \end{aligned} \tag{7}$$

The wellhead coordinates are known; therefore, D_1 and E_1 are always known during the recursive process. The 3D coordinate values of the trajectory can be obtained by the recursive calculation. The methods of calculation are different due to the different assumed curve types of the well segments. This article supports number of commonly used

calculation methods in the industry. The following are the most commonly used minimum curvature method:

$$\begin{aligned}\Delta D &= \frac{\Delta L}{2} (\cos \alpha_1 + \cos \alpha_2) \frac{2}{r} \tan \frac{r}{2}, \\ \Delta E &= \frac{\Delta L}{2} (\sin \alpha_1 \sin \beta_1 + \sin \alpha_2 \sin \beta_2) \frac{2}{r} \tan \frac{r}{2}, \\ \Delta N &= \frac{\Delta L}{2} (\sin \alpha_1 \cos \beta_1 + \sin \alpha_2 \cos \beta_2) \frac{2}{r} \tan \frac{r}{2}, \\ \gamma &= \cos^{-1} (\cos \alpha_1 - \cos \alpha_2 + \sin \alpha_1 \sin \alpha_2 \cos (\beta_1 - \beta_2)),\end{aligned}\quad (8)$$

where L is the measuring depth, m; α is the deviation angle, deg; φ is the azimuth angle, deg; D is the true vertical depth, m; N is the North/South coordinate, m; E is the East/West coordinate, m; ΔD is the vertical depth increment, m; ΔN is the North/South coordinate increments, m; ΔE is the East/West coordinate increments, m; ΔL is the length of well depth between two adjacent points, m; and γ is Dogleg, deg.

In the well trajectory model, a new method of natural curve method is proposed. Because of the high complexity of the model, it cannot be solved analytically. It needs to solve the complex implicit equation, and it is easy to encounter the problem of iterative divergence. This paper discusses the natural curve calculation methods under fixed point problem and designs a new solution process.

Fixed point problem (build turn point): given $\Delta N \Delta E \Delta Z$ solving ΔL , K_α , and K_ϕ , as shown in Figure 9.

Assume the starting and ending points of the well section are A and B, and the known conditions are $(L_A, \alpha_A, \phi_A, N_A, E_A, Z_A)$ and (N_B, E_B, Z_B) , where L_A, α_A, ϕ_A and L_B, α_B, ϕ_B are depth measurement, well deviation angle, and azimuth angle of the two measuring points above and below the well section, K_α is the rate of deviation change (build-up rate), K_ϕ is the azimuth rate of change (steering rate), and $\Delta N, \Delta E, \Delta Z$ are the increments of the well section in the North, East, and vertical directions, respectively.

The process of solving the model is as follows:

(1) Calculate the slope of AB line:

$$\alpha_{A,B} = \cos^{-1} \left(\frac{\Delta Z}{L_{A,B}} \right), \quad (9)$$

where

$$\begin{aligned}L_{A,B} &= \sqrt{\Delta N^2 + \Delta E^2 + \Delta Z^2}, \\ \Delta N &= N_B - N_A, \Delta E = E_B - E_A, \Delta Z = Z_B - Z_A.\end{aligned}\quad (10)$$

When $\alpha_{A,B} > 0$, the well section is building up. When $\alpha_{A,B} < 0$, then the well section is dropping, and when $\alpha_{A,B} = 0$, the slope is holding.

(2) Assign an initial value to α_B :

$$\alpha_B^0 = 2\alpha_{A,B} - \alpha_A. \quad (11)$$

(3) Confirm the initial value of ϕ_B as ϕ_B^0 .

According to the azimuth and coordinate offset of point A, the azimuth change direction and initial value of azimuth are solved. Solving the initial value of azimuth, the transformation matrix M is established according to ϕ_A , and N and E are transformed to determine the coordinate increment along the direction of azimuth ϕ_A , and the initial value of azimuth change $\Delta\phi^0$ is determined:

$$\begin{bmatrix} \Delta N' \\ \Delta E' \end{bmatrix} = \begin{bmatrix} \cos \phi_A & \sin \phi_A \\ -\sin \phi_A & \cos \phi_A \end{bmatrix} \begin{bmatrix} \Delta N \\ \Delta E \end{bmatrix}, \quad (12)$$

$$\Delta\phi^0 = \tan^{-1} \left(\frac{\Delta E'}{\Delta N'} \right),$$

$$\phi_B^0 = \phi_A + \Delta\phi^0.$$

(4) Calculate well section length ΔL .

According to the equation,

$$\Delta L = \frac{\Delta Z}{c \cos \bar{\alpha}}. \quad (13)$$

(5) Solve azimuth ϕ_B :

$$\frac{\Delta N}{\Delta E} = \frac{1/2\Delta L ((\cos(\alpha_A + \phi_A) - \cos(\alpha_B + \phi_B))/\alpha_B - \alpha_A + \phi_B - \phi_A) + (\cos(\alpha_A - \phi_A) - \cos(\alpha_B - \phi_B))/\alpha_B - \alpha_A - \phi_B + \phi_A)}{1/2\Delta L ((\sin(\alpha_B - \phi_B) - \sin(\alpha_A - \phi_A))/\alpha_B - \alpha_A - \phi_B + \phi_A) + (\sin(\alpha_B + \phi_B) - \sin(\alpha_A + \phi_A))/\alpha_B - \alpha_A + \phi_B - \phi_A)}, \quad (14)$$

$$\alpha_B - \alpha_A - \phi_B + \phi_A = \Delta\alpha - \Delta\phi, \quad (15)$$

$$\alpha_B - \alpha_A + \phi_B - \phi_A = \Delta\alpha + \Delta\phi, \quad (16)$$

$$f(\phi_B) = \Delta N \left(\frac{\sin(\alpha_B - \phi_B) - \sin(\alpha_A - \phi_A)}{\Delta\alpha - \Delta\phi} - \frac{\sin(\alpha_B + \phi_B) - \sin(\alpha_A + \phi_A)}{\Delta\alpha + \Delta\phi} \right) - \Delta E \left(\frac{\cos(\alpha_A + \phi_A) - \cos(\alpha_B + \phi_B)}{\Delta\alpha + \Delta\phi} + \frac{\cos(\alpha_A - \phi_A) - \cos(\alpha_B - \phi_B)}{\Delta\alpha - \Delta\phi} \right) = 0. \quad (17)$$

In equation (17), there is no analyzing expression for ϕ_B . ϕ_B is monotonically differentiable in the critical region, so the Newton iteration method is used to calculate ϕ_B :

$$f'(\phi_B) = \Delta N \left(\frac{\sin(\alpha_B - \phi_B) - \sin(\alpha_A - \phi_A)}{(\Delta\alpha + \Delta\phi)^2} - \frac{\cos((\alpha_B + \phi_B))}{\Delta\alpha + \Delta\phi} + \frac{\sin(\alpha_B + \phi_B) - \sin(\alpha_A + \phi_A)}{(\Delta\alpha + \Delta\phi)^2} - \frac{\cos(\alpha_B - \phi_B)}{\Delta\alpha - \Delta\phi} \right) - \Delta E \left(\frac{\cos(\alpha_B + \phi_B) - \cos(\alpha_A + \phi_A)}{(\Delta\alpha + \Delta\phi)^2} + \frac{\sin((\alpha_B + \phi_B))}{\Delta\alpha + \Delta\phi} - \frac{\cos(\alpha_B - \phi_B) - \cos(\alpha_A - \phi_A)}{(\Delta\alpha + \Delta\phi)^2} + \frac{\sin(\alpha_B - \phi_B)}{\Delta\alpha - \Delta\phi} \right). \quad (18)$$

By substituting ϕ_B^0 into equation (16), $f[\phi_B^0]$ is obtained. If $|f[\phi_B^0]|$ is less than the set error, this paper uses 1.0E-6, then ϕ_B^0 is the solved azimuth angle ϕ_B ; otherwise, use the following formula to calculate the new ϕ_B^0 :

$$\phi_B^0 = \phi_B^0 - \frac{f[\phi_B^0]}{f'[\phi_B^0]} \quad (19)$$

By repeating the above iterative process, ϕ_B can be obtained.

It should be noted that, in formula (17) and formula (18), when there are two molecular terms $|\Delta\alpha \pm \Delta\phi| \rightarrow 0$, due to the floating-point calculation error of the computer, the calculation result will not be accurate. When it is equal to zero, there will be an error of dividing zero. In the process of iteration, this situation is avoided. Therefore, the algorithm must consider the limit problem of $|\Delta\alpha \pm \Delta\phi| \rightarrow 0$. In this critical region, limit is used instead of calculation to avoid calculation error.

The following is the limit problem in the above formula:

$$\begin{aligned} \lim_{\Delta\alpha + \Delta\phi \rightarrow 0} \frac{\sin(\alpha_B + \phi_B) - \sin(\alpha_A + \phi_A)}{\Delta\alpha + \Delta\phi} &= \cos(\alpha_A + \phi_A), \\ \lim_{\Delta\alpha - \Delta\phi \rightarrow 0} \frac{\sin(\alpha_B - \phi_B) - \sin(\alpha_A - \phi_A)}{\Delta\alpha - \Delta\phi} &= \cos(\alpha_A - \phi_A), \\ \lim_{\Delta\alpha - \Delta\phi \rightarrow 0} \frac{\cos(\alpha_A + \phi_A) - \cos(\alpha_B + \phi_B)}{\Delta\alpha + \Delta\phi} &= \sin(\alpha_A + \phi_A), \\ \lim_{\Delta\alpha - \Delta\phi \rightarrow 0} \frac{\cos(\alpha_A - \phi_A) - \cos(\alpha_B - \phi_B)}{\Delta\alpha - \Delta\phi} &= \sin(\alpha_A - \phi_A), \\ \lim_{\Delta\alpha - \Delta\phi \rightarrow 0} \frac{\sin(\alpha_B + \phi_B) - \sin(\alpha_A + \phi_A)}{(\Delta\alpha + \Delta\phi)^2} - \frac{\cos(\alpha_B + \phi_B)}{\Delta\alpha + \Delta\phi} &= \frac{1}{2} \sin(\alpha_A + \phi_A), \\ \lim_{\Delta\alpha - \Delta\phi \rightarrow 0} \frac{\sin(\alpha_B - \phi_B) - \sin(\alpha_A - \phi_A)}{(\Delta\alpha - \Delta\phi)^2} - \frac{\cos(\alpha_B - \phi_B)}{\Delta\alpha - \Delta\phi} &= \frac{1}{2} \sin(\alpha_A - \phi_A), \\ \lim_{\Delta\alpha - \Delta\phi \rightarrow 0} \frac{\cos(\alpha_B + \phi_B) - \cos(\alpha_A + \phi_A)}{(\Delta\alpha + \Delta\phi)^2} + \frac{\sin(\alpha_B + \phi_B)}{\Delta\alpha + \Delta\phi} &= \frac{1}{2} \cos(\alpha_A + \phi_A), \\ \lim_{\Delta\alpha - \Delta\phi \rightarrow 0} \frac{\cos(\alpha_B - \phi_B) - \cos(\alpha_A - \phi_A)}{(\Delta\alpha - \Delta\phi)^2} + \frac{\sin(\alpha_B - \phi_B)}{\Delta\alpha - \Delta\phi} &= \frac{1}{2} \cos(\alpha_A - \phi_A). \end{aligned} \quad (20)$$

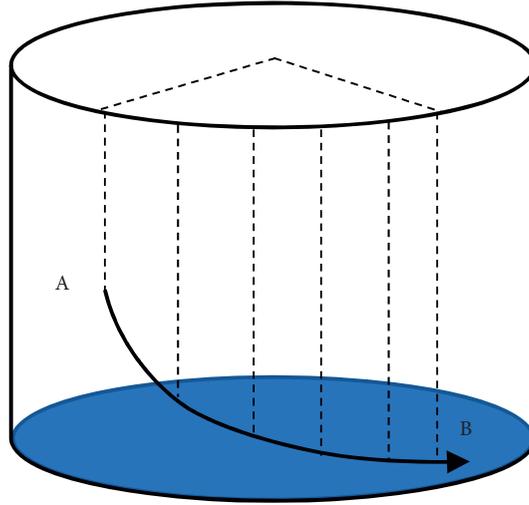


FIGURE 9: Schematic diagram of natural curve method.

(6) Solve well deviation angle α_B .

There is the following relationship:

$$\Delta L = \frac{2\Delta N}{((\cos(\alpha_A + \phi_A) - \cos(\alpha_B + \phi_B))/(\Delta\alpha + \Delta\phi)) + ((\cos(\alpha_A - \phi_A) - \cos(\alpha_B - \phi_B))/(\Delta\alpha - \Delta\phi))}, \quad (21)$$

$$\Delta L = \frac{2\Delta E}{((\sin(\alpha_B - \phi_B) - \sin(\alpha_A - \phi_A))/(\Delta\alpha - \Delta\phi)) - ((\sin(\alpha_B + \phi_B) - \sin(\alpha_A + \phi_A))/(\Delta\alpha + \Delta\phi))}. \quad (22)$$

Equation (21) or equation (22) is selected to calculate the current ΔL according to the absolute values of ΔN and ΔE . Equation (23) is derived from formula $\Delta Z = c\Delta L\cos\bar{\alpha}$, and the new well deviation angle α_B^0 can be calculated as

$$\alpha_B^0 = 2 \cos^{-1} \left(\frac{\Delta Z}{c\Delta L} \right) - \alpha_A. \quad (23)$$

If the error of α_B^0 between the two calculations is small enough (1.0E-6 is adopted in this paper), then the currents α_B^0 and ϕ_B^0 are the results of the final required solution, and the build-up slope and steering rate can be calculated by formulas (24) and (25). Otherwise, go to Step 4.

$$K_\alpha = \frac{\Delta\alpha}{\Delta L}, \quad (24)$$

$$K_\phi = \frac{\Delta\phi}{\Delta L}. \quad (25)$$

3.3.3. Curve Components. The curve component displays the curve parameter changes along the wellbore trajectory, such as torque, hook load, riser pressure, drilling time, rotating speed of turntable and other engineering data, or logging data, e.g., gamma ray. Each curve is displayed in a different color. Another curve style is displayed in the form of tube. The component represents the change in one parameter with the different colors of the color-map, and the change in the

thickness of the cylinder can represent the change in another parameter (such as well diameter).

3.3.4. Formation Tops Component. Formation tops are described using the following data, as shown in Table 2.

Formation tops are defined in the grid mode, including M columns and N rows grid vertex. It contains $(M - 1) \times (N - 1)$ quadrangles, but some of them are not included in the formation tops model. The data only contain the vertexes of visible quadrangles, and each vertex includes north coordinate, east coordinate, altitude, column number, and row number. The formation tops display is mainly realized by creating a Three.js mesh. The method is as follows:

- (1) Create material based on color-map.
- (2) Calculate the depth range of all visible vertexes (D_{\min}, D_{\max}).
- (3) Convert all vertex coordinates into a uniform coordinate array position.
- (4) According to the indexes position, each visible quadrangle is treated as two triangles to form an index array of vertex coordinates of the triangle indices.
- (5) Calculate material coordinates of vertexes $(u, v) = ((Di - D_{\min}) / (D_{\max} - D_{\min}), 0)$, and form material coordinate array— uvs .

TABLE 2: Formation tops data format.

Map north (m)	Map east (m)	Altitude (m)	Column	Row
18496205.34	3106995.77	2020.84	315	1
18496254.58	3106987.09	2021.12	316	1
18496214.02	3107045.01	2027.11	315	2
18496263.26	3107036.33	2028.75	316	2
18496017.06	3107079.74	2028.87	307	3
18496066.30	3107071.06	2025.83	308	3
⋮	⋮	⋮	⋮	⋮

- (6) Create mesh object and compute normal vectors automatically.

3.3.5. Tube Curve Component. Tube curve component is used to display the tube curve along the path of borehole trajectory. The amplitude of the tube curve is displayed by different diameters and colors, which can show the change in a certain parameter along the borehole intuitively. Figure 10 is the basic principle of drawing the component. It divides the wellbore circumferentially into certain parts, such as 20, and takes values at certain depth intervals on the curve. Therefore, the whole barrel curve is transformed into drawing a series of quadrilateral, and each quadrilateral can be drawn with two triangles, as shown in Figure 10.

3.3.6. Seismic Slice Component. Generally, the amount of seismic data is very large. The seismic slice data near the drilling profile are processed and extracted by the pre-processing program and then saved as an image file. In this way, the seismic slice component only needs to load a small amount of data from the server to display the slice.

4. RESTful Data Service

It is necessary to develop corresponding background data services to provide data interface support for front-end application. In the past, traditional technologies such as Java and .NET were usually used to develop back-end data services. The development and deployment of services were complex, and it was difficult to combine them with advanced cloud and container technologies. For a long time, web service was the most mainstream way to build SOA web application, and it uses a complex and heavy SOAP message format based on XML. The concept of RE-presentational State Transfer (REST) was first proposed by Dr. Roy Thomas Fielding in his doctoral thesis in 2000 [16] as a way to provide interoperability between the computer systems in the Internet. It overcomes difficulties that are associated with conventional web services, realize more efficient data communications on the web. This paper presents a lightweight restful data service solution for the system. The scheme adopts microservice architecture and chooses Node.js—a lightweight development platform suitable for cloud platform [17]. TypeScript development language in Restify development framework is used for developing a highly available REST-style data service. Figure 11 shows the REST request basic process.

5. Real-Time Data Transmission Scheme

WITSML (Wellsite Information Transfer Standard Markup Language) is a standard markup language for well site information transmission based on XML, which can realize seamless data exchange between service companies and oil companies [18], but the WITSML standard is huge; the cost of implementing a WITSML server is high. Referring to some design ideas of WITSML, this paper designs a set of real-time data transmission scheme. Data transmission adopts the HTTP REST mode, and data objects are encapsulated in the JSON format. The scheme is simple to implement and can make up for the shortcomings of WITS0. The concept of timestamp is adopted in the design, which can not only acquire real-time data but also historical data, and can expand the interface of other static data at any time.

6. Data Service API

6.1. Data Type Definition. Visual drilling data are defined below, including well basic data, drilling trajectory data, logging data, layer interface data, geological slice data, and reservoir grid data.

6.1.1. Trajectory Data. The inclinometer data are used to describe the drilling trajectory, which comes from the drilling design or actual drilling. After receiving the data, the client needs to calculate (x, y, z) coordinates of each measuring point according to the drilling trajectory calculation method. These data are an array of objects, and the type definition of array elements is shown in Table 3.

Here are some examples of data:

```
{
  "md": 0, "dev": 0, "azi": 0 },
{ "md": 24.003, "dev": 0.44, "azi": 25.45 },
{ "md": 36.54, "dev": 0.79, "azi": 327.32 }
]
```

6.1.2. Logging Curve Data. Logging data are used to describe a set of measurement data along the drilling trajectory. The data consist of a series of measurement depth and measurement values. The data may come from LWD, MWD, geological logging, and logging. Continuous or tubular curves along drilling trajectory can be generated by logging data. Data-type definitions are shown in Table 4.

Here are some examples of data:

```
{
  "name": "GR",
```

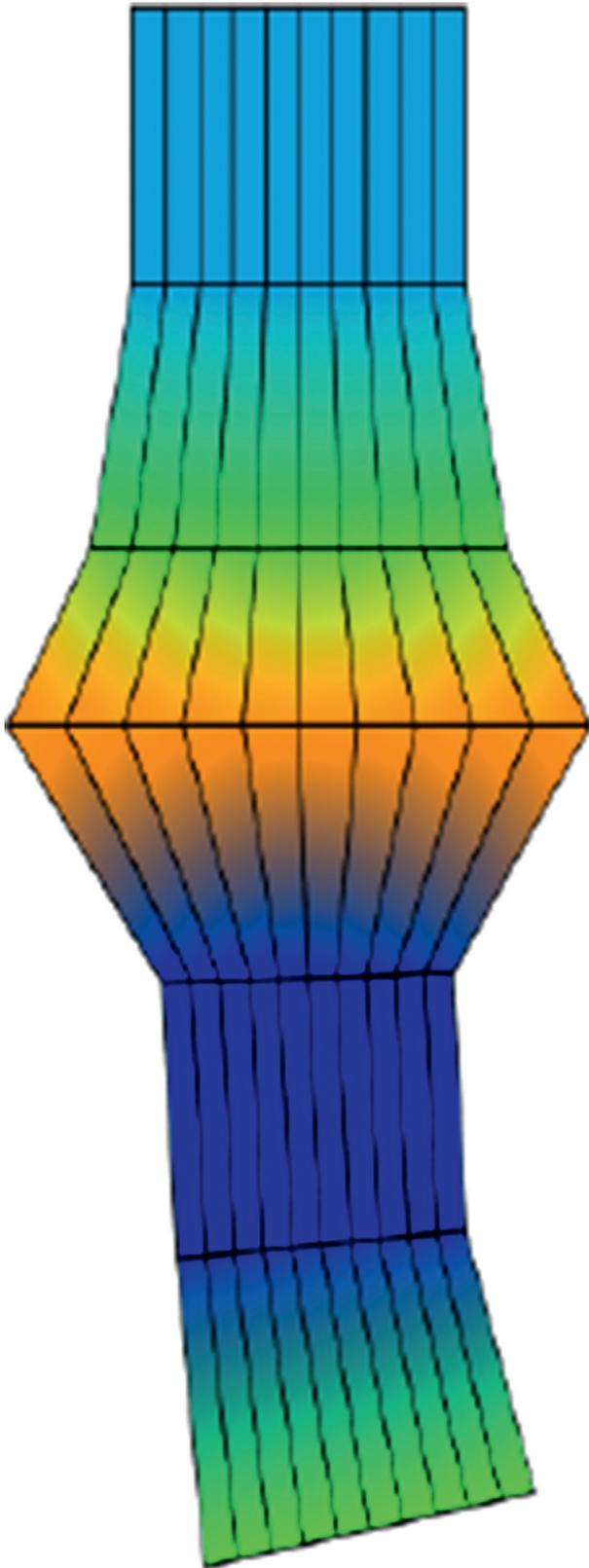


FIGURE 10: Tube curve schematic.

```

“min”: 0,
“max”: 400,
“depth”: [100, 100.5, 101, 101.5, 102],

```

```

“value”: [78.460, 81.242, 75.677, 80.046, 81.659]
}

```

6.1.3. *Formation Tops Data.* Formation tops data are used to describe the top interface of geological horizon. The data volume of layer interface is a two-dimensional array. The definition is shown in Table 5.

Here are some examples of data:

```

{
“nx”:201,
“ny”:101,
“vectors”: [
[18459025, 3269000, 2687.109375],
[18459050, 3269000, 2687.5],
[18459075, 3269000, 2687.109375],
[18459100, 3269000, 2687.890625],
[18459125, 3269000, 2686.71875],
[18459150, 3269000, 2687.5],
[18459175, 3269000, 2686.71875],
[18459200, 3269000, 2687.5],
...
]
}

```

6.1.4. *Geological Slice Data.* The geological data are often very large, such as seismic data volume of a block, which often reaches tens of GB. It is impossible and unnecessary to load the data of the whole data volume into the client for 3D display. In the process of drilling, usually only the geological information near the drilling trajectory is concerned. It is only necessary to display information of some facets to the 3D scene. The geological slice data format defined by this software is shown in Table 6.

Image is slice data. It is a base64 encoding string converted from PNG format pictures in the same format as HTML embedded pictures. Corners is the coordinates of the four corners of the slice $\{[x_1, y_1, z_1], [x_2, y_2, z_2], [x_3, y_3, z_3], [x_4, y_4, z_4]\}$.

Here is the sample data:

```

{
“name”: “slice1”,
“image”: “data:image/png; base64,iVBORw0A...”,
“corners”: [
{“x”: 4881500, “y”: 15617500, “z”: 825 },
{“x”: 4881500, “y”: 15621400, “z”: 825 },
{“x”: 4881500, “y”: 15621400, “z”: 2600 },
{“x”: 4881500, “y”: 15617500, “z”: 2600 }
]
}

```

6.2. *Service Interface Design.* According to the definition of drilling visualization data, the corresponding data interface is designed. To access data services, you only need to use HTTP interface address. The address format of the interface is `//host:port/api/interface/{parameter}`; for example, the host is local host, the port is 4000, “the block list,” and the interface name is “block,” no parameters, and the interface

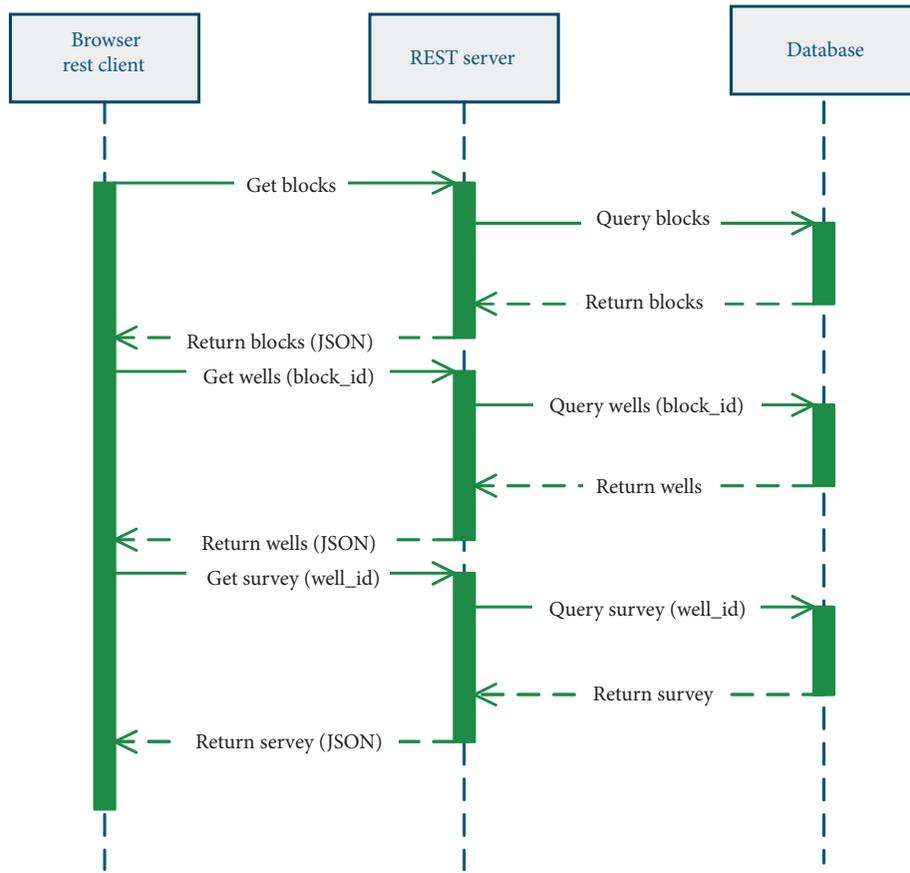


FIGURE 11: REST request basic process.

TABLE 3: Survey data.

Field	Type	Unit	Explanation
Md	Number	m	Measuring well depth
Dev	Number	deg	Well deviation angle
Azi	Number	deg	Azimuth

TABLE 4: Curve data.

Field	Type	Unit	Explanation
Name	String		Curve name
Min	Number		Left scale
Max	Number		Right scale
Color	String		Default color
Depth	Number[]	m	Depth of measurement
Value	Number[]		Curve value

address is //localhost:4000/blocks. The result is in the JSON format. Interface definitions are shown in Table 7.

The design has strong expansibility and can easily expand support for other drilling and completion and geological data by adding new data interface.

6.3. Cross-Domain Problem. When browsers want to access data service interfaces under different domain names, it needs to solve the problem of cross-domain access considering that data services may not be in the same server with

Web servers. The commonly used method is JSONP. JSONP (JSON with Padding) is a “usage mode” of JSON. For security reasons, browsers have a homologous policy. Generally speaking, web pages located at server 1.example.com cannot communicate with servers that are not server 1.example.com, but HTML `<script>` element is an exception. Using the open strategy of `<script>` elements, web pages can get JavaScript code generated dynamically from other sources. Through ingenious design, executing this code can dynamically generate JSON data needed by users. This mode of use is called JSONP. The service program is designed to support both JSON and JSONP. It only needs to add “? Callback=jsonp” after the interface address. The program returns JavaScript code in the JSONP mode; otherwise, it returns JSON data directly. The web page can get JavaScript code generated dynamically from other sources with the open strategy of `<script>` element. Through ingenious design, executing this code can generate JSON data dynamically required by users. This usage pattern is called JSONP.

6.4. Service Deployment. Service program can be deployed conveniently on the cloud server using PM2 and NodeJS, and the update and execution of the program can be monitored to ensure the uninterrupted operation and hot update of the program through the powerful process management function of PM2 [19, 20]. PM2 is a Node.js production environment process management tool, which

TABLE 5: Formation tops data.

Field	Type	Unit	Explanation
Nx	Number		X-direction points
Ny	Number		Y-direction points
Vectors	Number[3]	m	nx * ny group (x, y, z) coordinates

TABLE 6: Geological slice data type.

Field	Type	Unit	Explanation
Name	String		Slice name
Image	String		Slice data
Corners	Vector3[4]	m	Slice corner coordinate data

TABLE 7: Service interface definition.

Interface name	Interface address
Block list	Blocks
Well list	Wells/{block_id}
Formation tops list	Surfaces/{block_id}
Seismic slice list	Slices/{block_id}
Trajectory data	Survey/{well_id}
Curve list	Curves/{well_id}
Events list	Events/{well_id}
Curve data	Curve/{curve_id}
Formation tops data	Surface/{surface_id}

has built-in load balancing function, especially for micro-service-based applications. The use of PM2 guarantees the uninterrupted operation of Node.js services. Once a program error occurs, PM2 will automatically restart the process, which is critical for data service program. The REST-style data service developed by using Node.js development platform is a lightweight technology, which is more suitable for cloud platform at present. Through JSONP, cross-domain access is realized. PM2 is used to deploy the service program. The drilling 3D visualization system developed by this service has the characteristics of high reliability, reliability, and easy expansion. With the advantage of static type of advanced TypeScript language, the software can eliminate most potential errors in the development stage and solve the problem of weak type and poor reliability of JavaScript language. By compiling to JavaScript execution, it can give full play to the advantages of Node.js in cross-platform and lightweight. The lightweight data service solution described in this paper is completely based on open-source technology, does not rely on any company's proprietary technology and platform, can freely choose the server deployment platform, and can be combined with advanced cloud and container technology.

7. Applications and Discussion

The software is now installed/used in the RTOC is shown in Figures 12 and 13, for example, block 291, which contains 11 wells of carbonate rock, including 4 vertical wells, 1 directional well, and 6 horizontal wells, among which W291-H9 is the latest horizontal well and the target formation is Ordovician. The drilling trajectory, logging data,

and complex and accidents and seismic slice data along the well trajectory of the target formation are loaded into the database.

The software can display the complex and accidents of drilling in the block, the seismic characteristics of the target layer, the tops of the target formation, drilling trajectories, and LWD data and can be used for auxiliary analysis and decision-making. The drilling and geological experts can not only use the workstation of the Office but also use the mobile equipment to view the data and graphics.

Due to the consistency and efficiency of WebGL on various platforms, the software has good performance and consistency on the main platforms, including Windows, Linux, macOS, Android, and iOS, and the responsive layout can adapt well for the different resolutions of desktop devices and mobile devices; for small mobile screen, the software will automatically hide the left tree menu, making the software more practical. The goal of the development is achieved.

In Figure 14, drilling 3D scene is visualized on iPad, Android, and iPhone different devices. Compared with non-cross-platform software, WebDrillingViz provides a powerful fundamental tool for the drilling visualization applications. Different from current studies on visualization of 3D drilling visualization, WebDrillingViz is lightweighted, cross-platform, and open. We can anticipate that WebDrillingViz can be rewarding to more applications during the life-cycle drilling. At the end, it is verified that WebDrillingViz is efficient, compatible with mainstream devices through extensive real projects' drilling and geological data, and the rest interface of data service software fully meets the actual needs in terms of function, performance, and

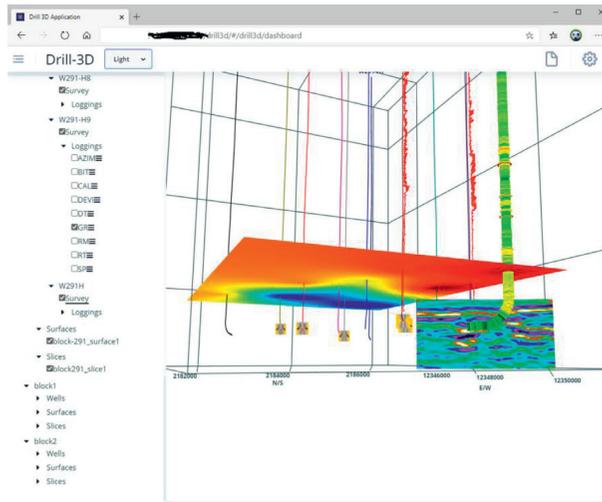


FIGURE 12: Drilling 3D scene: well tracks, surface, and loggings on Windows.

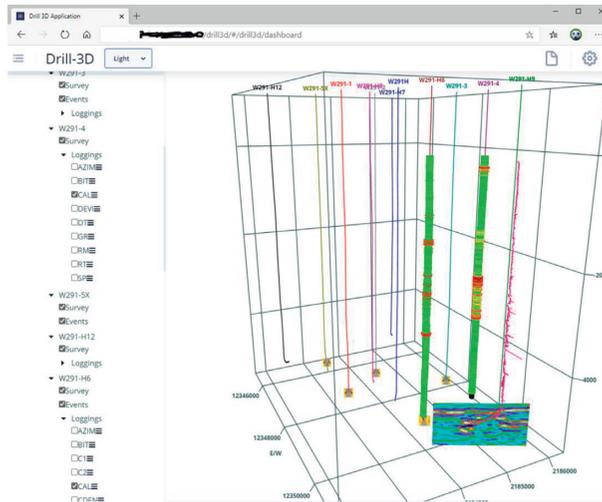


FIGURE 13: Drilling 3D scene: well tracks, loggings, and events.

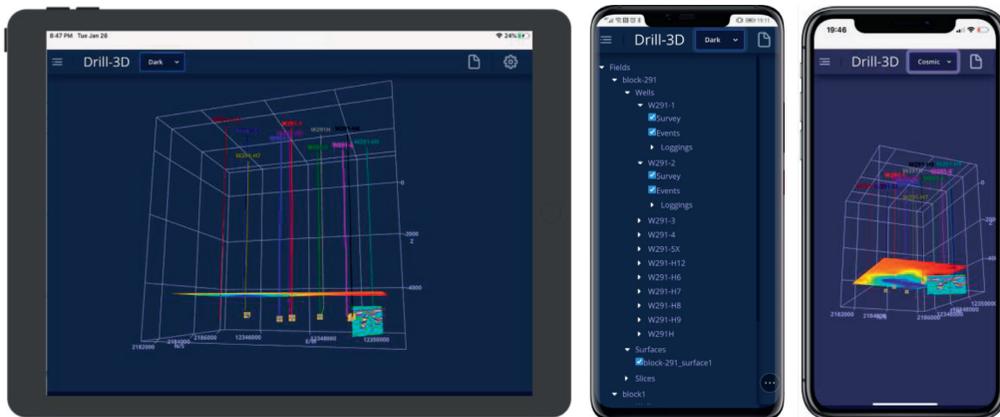


FIGURE 14: Drilling 3D Scene on iPad, Android, and iPhone.

reliability, which proves that the architecture proposed in this paper is a very valuable lightweight solution.

8. Conclusions

This paper introduces the structure and development of 3D visualization software for drilling engineering and geology. The software is based on WebGL and developed with TypeScript, an open-source framework. Combined with the real-time transmission scheme proposed, a set of low-cost and practical real-time 3D visualization monitoring software is provided for RTOC. The conclusions can be summarized as follows:

- (1) The use of native Web technologies allows detailed renderings of the drilling and geological data with no additional software to install. 3D drilling scene renderings are processed entirely within the modern Web browser without a significant tradeoff in performance. It is firstly suggested that an online 3D drilling visualization system has to meet three basic requirements: lightweighted, cross-platform, and open. Based on these basic rules, we developed a novel online 3D drilling visualization system based on HTML5 and WebGL, termed as WebDrillingViz. It is now installed in the RTOC, interfacing with other software systems. This software will enable decision-makers to have better insights into the status of the well and formation surrounding the well and thus makes better and quicker decisions.
- (2) WebDrillingViz integrates responsive design and high-performance back-end services, using the latest technologies, such as NodeJS, Angular, and Bootstrap frameworks with excellent compatibility. The front-end components can also quickly be embedding into other web applications and the back-end efficient service based on NodeJS platform. The software is completely implemented in TypeScript language on both sides, and the whole development cycle reduced significantly because using same language. The REST-style data service is a lightweight technology, which is more suitable for cloud platform at present. Through JSONP, cross-domain access is realized. PM2 is used to deploy the service program. The service has the characteristics of high reliability, reliability, and easy expansion.

Data Availability

The data used to support the findings of this study are available from the first author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to thank China University of Petroleum (Beijing) for the support during this research and permission to publish these results. This work was financially

supported by the National Natural Science Foundation Project (Grant no. 51374222), National Major Project (Grant no. 2017ZX05032004-002), National Key Basic Research and Development Program (Grant no. 2015CB250905), and CNPC's Major Scientific and Technological Project (Grant no. 2017E-0405).

References

- [1] D. H. Kaminiski, N. M. Pellerin, and J. H. Williams, "A new data integration and work process system for providing online real-time drilling collaboration," in *Proceedings of the European Petroleum Conference*, Aberdeen, UK, October 2002.
- [2] A. L. F. Madaleno, S. L. S. Neto, L. A. Dos Santos, and C. A. L. De Oliveira, "Operational and safety improvements of applying real-time analytics in a drilling contractor RTOC," in *Proceedings of the Annual Offshore Technology Conference*, Houston, TX, USA, May 2018.
- [3] L. J. Ursem, J. H. Williams, N. M. Pellerin, and D. H. Kaminski, "Real time operations centers; the people aspects of drilling decision making," in *Proceedings of the Drilling Conference*, Amsterdam, The Netherlands, February 2003.
- [4] G. A. Dorn, K. Touyinhthiphonexay, J. Bradley, and A. Jamieson, "Immersive 3-D visualization applied to drilling planning," *The Leading Edge*, vol. 20, no. 12, pp. 1389–1392, 2001.
- [5] J. Holt, W. J. Wright, H. Nicholson, A. Kuhn-De-Chizelle, and C. Ramshorn, "Mungo field: improved communication through 3D visualization of drilling problems," in *Proceedings of the SPE/AAPG Western Regional Meetings*, vol. 53, Long Beach, CA, USA, June 2000.
- [6] R. Rommetveit, K. S. Bjørkevold, S. I. Ødegård, M. Herbert, and G. W. Halsey, "Automatic real-time drilling supervision, simulation, 3D visualization and diagnosis on ekofisk," in *Proceedings of the SPE/IADC Drilling Conference* Orlando, FL, USA, March 2008.
- [7] R. Rommetveit, K. S. Bjørkevold, S. I. Ødegård, O. Sandve, B. Larsen, and M. Herbert, "EDrilling: linking advanced models and 3D visualization to drilling control systems in real time," in *Proceedings of the Offshore Mediterranean Conference and Exhibition 2007, OMC 2007*, Ravenna, Italy, March 2007.
- [8] W. C. Sanstrom and M. J. Hawkins, "Perceiving drilling learning through visualization," in *Proceedings of the IADC/SPE Asia Pacific Drilling Technology Conference, APDT*, Limerick, Ireland, September 2000.
- [9] P. Song, L. Shi, Y. Zhou, Q. Zhao, H. Jiang, and P. Cui, "Study and applications on integrated drilling engineering software for drilling engineering design and real-time optimization (Russian)," in *Proceedings of the Society of Petroleum Engineers - SPE Annual Caspian Technical Conference and Exhibition*, Astana, Kazakhstan, November 2014.
- [10] W. Phillips, "Interactive web-based 3D wellbore viewer enables collaborative analysis," *SPE Production & Operations*, vol. 33, no. 02, pp. 345–352, 2018.
- [11] X. Liu, N. Xie, K. Tang, and J. Jia, "Lightweighting for Web3D visualization of large-scale BIM scenes in real-time," *Graphical Models*, vol. 88, pp. 40–56, 2016.
- [12] I. Morozov, G. Chubak, and S. Blyth, "Interactive 3D/2D visualization for geophysical data processing and interpretation," *Computers and Geosciences*, vol. 35, no. 7, pp. 1397–1408, 2009.

- [13] G. Bierman, M. Abadi, and M. Torgersen, *Understanding TypeScript*, vol. 8586, pp. 257–281, ECOOP 2014 - Object-Oriented Programming, Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2014.
- [14] J. Dickey, *Write Modern Web Apps with the Mean Stack: Mongo, Express, AngularJS, and Node.js (Develop and Design)*, Peachpit Press, Berkeley, CA, USA, 1st edition, 2014.
- [15] T. A. Walsh, P. McMinn, and G. M. Kapfhammer, “Automatic detection of potential layout faults following changes to responsive web pages (N),” in *Proceedings of the 2015 30th IEEE/ACM International Conference on Automated Software Engineering*, Lincoln, NE, USA, November 2015.
- [16] R. T. Fielding and R. N. Taylor, “Principled design of the modern web architecture,” *ACM Transactions on Internet Technology*, vol. 2, no. 2, pp. 115–150, 2002.
- [17] M. Cantelon, M. Harter, T. J. Holowaychuk, and N. Rajlich, *Node.js in Action*, American: Manning Publications, Shelter Island, NY, USA, 2013.
- [18] M. Khudiri, J. James, M. Amer, B. Otaibi, M. Nefai, and J. Curtis, “The integration of drilling sensor real-time data with drilling reporting data at Saudi Aramco using WITSML,” in *Proceedings of the Society of Petroleum Engineers - SPE Intelligent Energy International 2014*, Utrecht, The Netherlands, April 2014.
- [19] X. Chenjie, Z. Xiang, P. Xin, and Z. Wenyun, “Microservice system oriented runtime deployment optimization,” *Computer Application Software*, vol. 35, no. 10, pp. 85–93, 2019.
- [20] L. Yongyi, “Research on cross-domain request scheme for large distributed web system,” *J. Chang. Univ.* vol. 35, no. 2, pp. 54–56, 2018.