processes

Article

Development of a Software Tool for Visualizing a Mine (Wellbore) in the Industrial Drilling of Oil Wells

Fares Abu-Abed [1], Kirill Pivovarov [1], Vitaly Zhironkin [2] and Sergey Zhironkin [3,4,5,*]

[1] Department of Electronic Computers, Faculty of Information Technologies, Tver State Technical University, 22 Afanasiya Nikitina emb., 170026 Tver, Russia
[2] School of Natural Resources Engineering, National Research Tomsk Polytechnic University, 30 Lenina st., 634050 Tomsk, Russia
[3] Department of Open Pit Mining, T.F. Gorbachev Kuzbass State Technical University, 28 Vesennya st., 650000 Kemerovo, Russia
[4] Department of Trade and Marketing, Siberian Federal University, 79 Svobodny av., 660041 Krasnoyarsk, Russia
[5] School of Engineering Entrepreneurship, National Research Tomsk Polytechnic University, 30 Lenina st., 634050 Tomsk, Russia
* Correspondence: zhironkinsa@kuzstu.ru

Abstract: The purpose of the software development presented in the article is to obtain detailed information about the surface of the walls in wells necessary for more efficient and safe drilling and casing of the wellbore. The developed software of the automated system "LogPWin" allows obtaining a clear idea of the mine itself, as well as reducing the time spent on processing and analyzing the model obtained during the logging process. A software product developed as a result of the work provides visualization of data coming from an acoustic profiler, giving a visual and informative idea of the object of study. The program gives detailed information about the surface of the mine's walls through 3D modeling. It possesses the possibility of rotating and scaling the object; there is also a color palette that can be changed depending on the incoming data. The program can be applied in the oil and gas industry and mining, as well as when drilling wells for other purposes. The program interface is quite clear and simple. The structure of the program is made in such a way that it allows obtaining data for building an object not only in real time, but also through interaction with a ready-made .txt file.

Keywords: mine; oil and gas industry; 3D modeling; well drilling; visualization of oilfield facilities; well walls

check for updates

Citation: Abu-Abed, F.; Pivovarov, K.; Zhironkin, V.; Zhironkin, S. Development of a Software Tool for Visualizing a Mine (Wellbore) in the Industrial Drilling of Oil Wells. *Processes* **2023**, *11*, 624. https://doi.org/10.3390/pr11020624

Academic Editor: Jose Carlos Pinto

Received: 22 January 2023
Revised: 13 February 2023
Accepted: 17 February 2023
Published: 18 February 2023

1. Introduction

Oil and gas are important natural resources that are necessary for human life, as well as in industrial production. Well-drilling methods are used to extract these resources located deep underground. The method of drilling oil or gas wells depends on the location of the rock formation. Drilling can be vertical, horizontal, or inclined. The horizontal method of drilling wells is considered the most effective and less costly. Thanks to this method, it becomes possible to extract oil from hard-to-reach places and develop complex rock sections.

The arsenal of applied equipment and technologies for enhanced oil recovery and commissioning of residual oil reserves is constantly being improved. At present, many developments based on existing technology have been proposed and tested, and domestic technology for the construction of such wells has been developed, but some problems, nevertheless, remain. For example, the problem of increasing the flow rate of wells in the face of falling production is acute for most oil-producing countries in the world. Horizontal wells, even with a nominal diameter, have a very rough wall due to the use of rotary tool rotation.

One of the solutions to this problem is the introduction of 3D models of well profiles, which provide a detailed analysis of the field. The current state of affairs with piezoelectric materials and digital electronic devices, as well as with the development of the theory of acoustic logging, allow this combined method to be fully implemented.

In the presented work, the goal is to visualize the data of an acoustic profiler to supplement and refine the results of standard acoustics. The final software product consists of a program that receives and processes data for further modeling. The libraries of this program also contain tables with additional data necessary for the program to work. It should be noted that the final software product can be used by drilling engineers for more efficient and safe drilling and casing of the wellbore, as well as the allocation of intervals for the installation of casing sealing devices (packers).

The disadvantages of steerable drilling systems with hydraulic downhole motors can be significantly reduced by using steerable rotary drilling systems, in which the bit moves along a predetermined trajectory with continuous rotation of the drill string. The intensity of the well curvature is determined by the ratio of the volumes of lateral cutting and drilling in the forward direction. In both systems, at the bit level, the bit rotation axis is always at an angle with respect to the borehole axis.

The value of this angle is determined by the geometry of the tool and the radius of curvature of the hole. Fully automated systems were first used on very costly, high-deviation wells, where they provided the ability to solve problems that existing downhole motor systems cannot. These wells were drilled using Camco's steerable rotary drilling systems and continuous monitoring of the hydrostatic pressure in the well.

According to the works of Li Huang, Sophie Ebersold, Alexander Kogtenkov, Alexandr Naumchev, Bertrand Meyer, Yinling Liu, and ALiyu Alege, reservoir development using acoustic profiler data processing visualization greatly simplifies the understanding of the structure of the resulting bottom hole and also allows obtaining a more detailed definition of the well profile. The technology of using this combination will play a major role both in the drilling phase and, more importantly, in the well completion and workover phase. The widespread recognition of the real benefits of visualization is opening up a new market for continuous tubing drilling [1–7].

In order to assess the efficiency and performance of an oil field, well logging is used.

Logging is carried out by electrical, magnetic, thermal, acoustic, and other methods. When they are carried out along the wellbore with the help of geophysical sensors lowered on a cable, some quantities are measured that depend on one or the totality of the physical properties of the rocks intersected by the well (Figure 1). Signals from the sensor are transmitted to the surface and recorded by mobile ground equipment in analog (in the form of diagrams) or digital form.

Well logging is a rather complex process, the method of which depends on the functionality, type of field, soil features, and other factors. The results obtained can be useful both in the repair of equipment and in the process of developing a discovered oil or gas field. Well logging is carried out in various ways, but all of them are united by one common task: the study of artificial and natural physical fields, which have different natures. The intensity of logging depends on the properties and rock of the soil.

Logging is usually classified as electrical, magnetic, thermal, or acoustic. We are interested in one of the types of acoustic logging—wave acoustic logging.

Unlike conventional sonic logging, wave sonic logging is a method that operates with the parameters not only of the longitudinal wave, but also of other elastic waves propagating in the well. These waves, first of all, include a transverse wave, in which fluctuations in the environment surrounding the well occur perpendicular to the direction of wave propagation—the so-called shear fluctuations. Among the packets of pseudo-Rayleigh waves, a packet of low-frequency, high-amplitude oscillations associated with the Stoneley wave is usually recorded. This wave in the low frequency range (less than 3 kHz) is often called a tube wave. At present, there are three waves in the practice of wave logging with informative waves, i.e., found practical application in solving geological problems—longitudinal wave (P), transverse wave (S), and Stoneley wave (St).

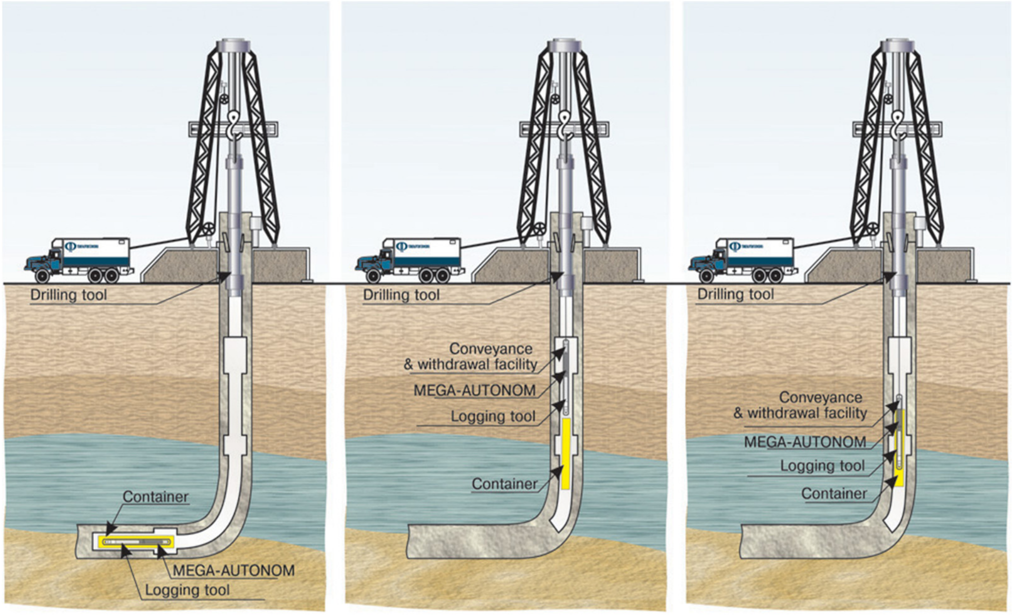


Figure 1. Logging process.

Due to the high frequency dispersion, the parameters of pseudo-Rayleigh waves have not yet found application in production geophysical surveys. The most commonly used modules are as follows:

- Bulk modulus (K)—an indicator of the resistance of the rock to all-round compression, measured in pascals (Pa), and usually has a value of 10 to 100 GPa. The reciprocal of K is called compressibility.
- Shear modulus (G)—an indicator of rock resistance to transverse shear, also measured in pascals and has values approximately 2 times less than K.
- Poisson's ratio (υ). The value of υ is an indicator of rock plasticity, which characterizes changes in the transverse dimensions of an elementary volume relative to longitudinal deformation in the event of longitudinal compression. Its value in rocks varies from 0 to 0.5 (practical range from 0.1 to 0.4) and depends on the mineralogical composition of the rocks. The smallest values of υ are associated with hard rocks composed of quartz or minerals with similar elastic properties. Clays or chemical sediments of the type of non-metamorphosed limestones have the highest values of υ.

A well survey using acoustic logging is presented in Figure 2.

The direct use of mechanical devices of cable tools, such as profilers, when delivering the tool to the well on the drill string is associated with a high risk of an accident. That is why the profilometry of horizontal wells is a very rare type of research, despite the high need for interpreting data from radioactive and electrical methods and solving a number of technical problems. Such tasks include, in particular, the determination of a place for installing a packer that seals the annulus above the formation fluid filter. The most direct non-contact measurement of the dimensions of a horizontal well can be provided by the acoustic method of reflected waves on the principle of emitting and receiving an echo signal. This method is described in the works of Nugraha and O'Sullivan and is used in cable acoustic TVs and scanners to solve problems of wellbore imaging and cement-casing contact density [2].

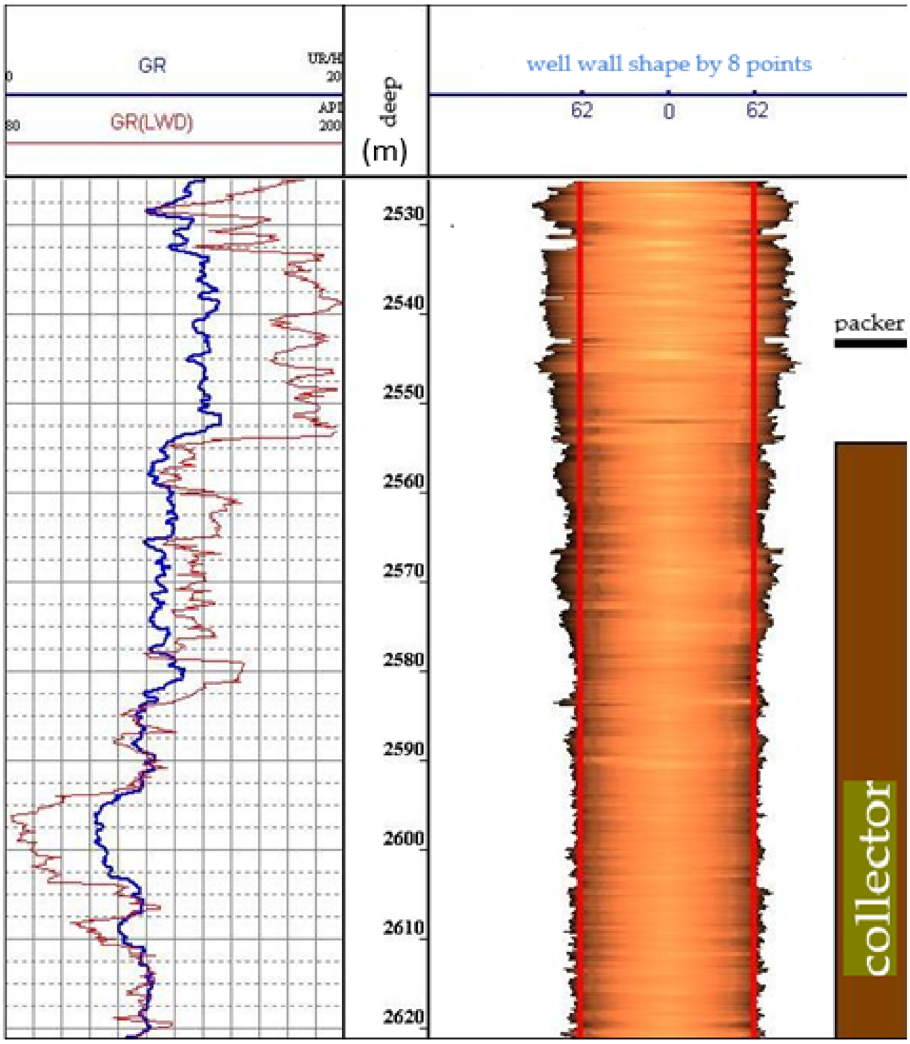


Figure 2. Well survey using acoustic logging.

2. Materials and Methods

Methods for receiving signals are described in the works [8,9]. With the advent of digital recording and processing techniques, more modern spatial filters have evolved, as have true two-dimensional filters. Such filtering was commonly known as velocity filtering or fan filtering (fan filtering) because the filter limits were determined because the rate limits and the corresponding 2D transmission rejection area were obtained using a 2D Fourier transform. Initially, velocity filters were applied directly in the space–time domain to attenuate the horizontal components of seismic arrays. The main purpose of velocity filtering is to suppress coherent linear velocity noise in seismic data. The Butterworth function—a mathematical function used to obtain the smoothest possible frequency response of the filter without taking into account the requirements for the phase response—is used to obtain a spectral picture without noise. The amplitude–frequency filters based on the Butterworth function have no ripple in either the passband or the stopband. The filter has a frequency response that is very close to flat in the middle of the passband and somewhat rounded around the cutoff frequency. Outside the passband, the decay rate increases.

An example of typical data is shown in Figure 3. The first column shows the depth, and the next nine are the phase correlation diagrams (WF 3–WF 9) for each sensor. The rightmost column shows the curve of natural GR gamma activity in μR/h. The wave pattern of the third channel at the depth marked by the cursor is displayed in the upper window, and the wave pattern of the reference sensor is in the lower window, on the right side of the figure. The shown wave pattern of the third channel is observed at the cavity boundary. Obviously, the reflection signals in the wave patterns of the measuring sensors are distorted by the reverberation noise of the sensors themselves (straight lines on the PCD). The shown wave pattern of the third channel gives a visual representation of the ratio of the useful signal and noise in the sections of the cavernous wellbore: the useful signal is not distinguished in the wave pattern. Its entry time (marked with a vertical cursor) is approximately equal to the entry time of the reflection on the reference channel, and the amplitude is 10 times less. Thus, direct measurement of the time of arrival of the reflected signal in the cavernous sections of the well is practically impossible.

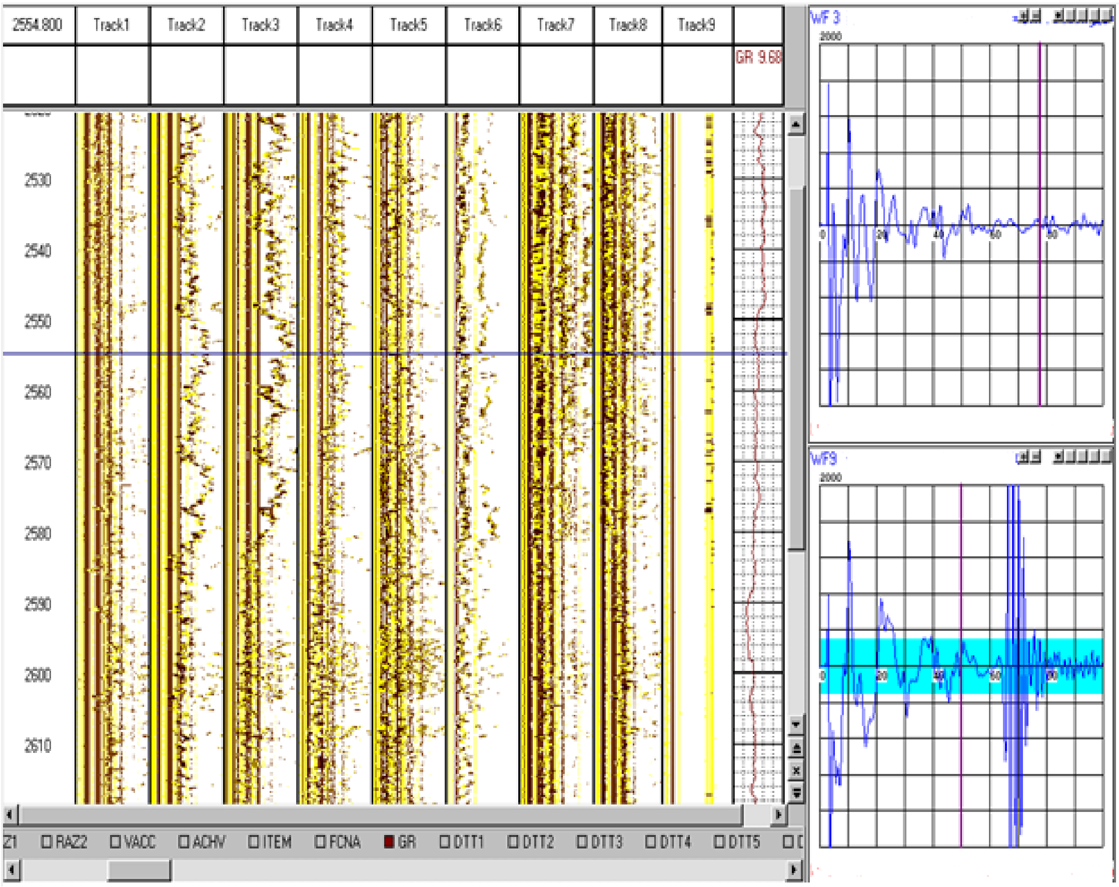


Figure 3. Field Data.

The signal from the third sensor of the profiler (the third radius) is shown in the upper right part of the figure. The abscissa shows the time in ms, and the ordinate shows the amplitude (oscillations of the tool in the wellbore) in conventional units of the tool. In the lower right corner, a wave pattern is presented, which is used to determine the speed of wave propagation in a liquid.

Although the acoustic caliper uses the measuring probe of an acoustic scanner, the conditions for their use are completely different. There are two important circumstances that make it possible to automatically extract the signal reflected from the borehole wall and, accordingly, measure its geometric dimensions:

- During the lifting of the drilling tool, within one candle, the device practically does not turn around its axis; therefore, the reverberation noise in the sensor itself and reflections from the metal parts of the probe adjacent to the sensor, at neighboring depth points, do not change.

- The amplitude of the first reflection is greater than the amplitudes of repeated reflections, and the frequency spectrum of the reflected signal does not depend on the acoustic properties of the surrounding rocks.

The first circumstance makes it possible to subtract noise from wave patterns. Wave patterns of the caliper sensor WF(t, zn) at adjacent depths zn and zn +1 are the sum of reflection signals from the borehole wall X(t) and noise V(t):

$$\mathrm{WF}(t, zn) = V\ (t, zn) + X(t, zn) \tag{1}$$

$$\mathrm{WF}(t, zn{+}1) = V(t, zn{+}1) + X\ (t, zn{+}1) \tag{2}$$

Taking into account the approximate equality of noise signals at neighboring depth points and the change in the arrival time and amplitude of the reflected signal, and by sequential subtraction, you can obtain a new array of wave patterns in which the noise will be subtracted:

$$K\ (t, zn) = X(t, zn) - X(t, zn + 1) \tag{3}$$

In denoised data, there are two reflections at each depth quantum. These signals are recorded on neighboring depth quanta, and in each wave pattern, they can be in different ratios: separated in time or interfering. Since the wavelength at the fundamental frequency of the acoustic pulse of 500 kHz is approximately 3 mm, which is comparable to the size of the borehole wall irregularities, and the duration of the acoustic pulse does not exceed two periods, in most cases the reflection signals are separated in time. The interference of two reflections with complete mutual suppression is the rarest case and can only occur at single measurement points. Consequently, either a pulse corresponding to a reflection at a given depth or a shifted one, but not more than one quantum, usually 0.2 m, can be taken as a reflection signal. This measurement bias is within the accuracy of the depth reference.

Figure 4 shows the data shown in Figure 3 after noise subtraction. As follows from the shape of the phase correlation plot and wave pattern, the quality of the data has changed radically. In almost the entire interval, the amplitudes of the reflection signals significantly exceed the amplitudes of the noise. The arrival times of the reflected signals, after noise subtraction, can be measured by the simplest threshold method. However, the use of this method is inconvenient due to the large amount of data and the need to create settings individually for each channel.

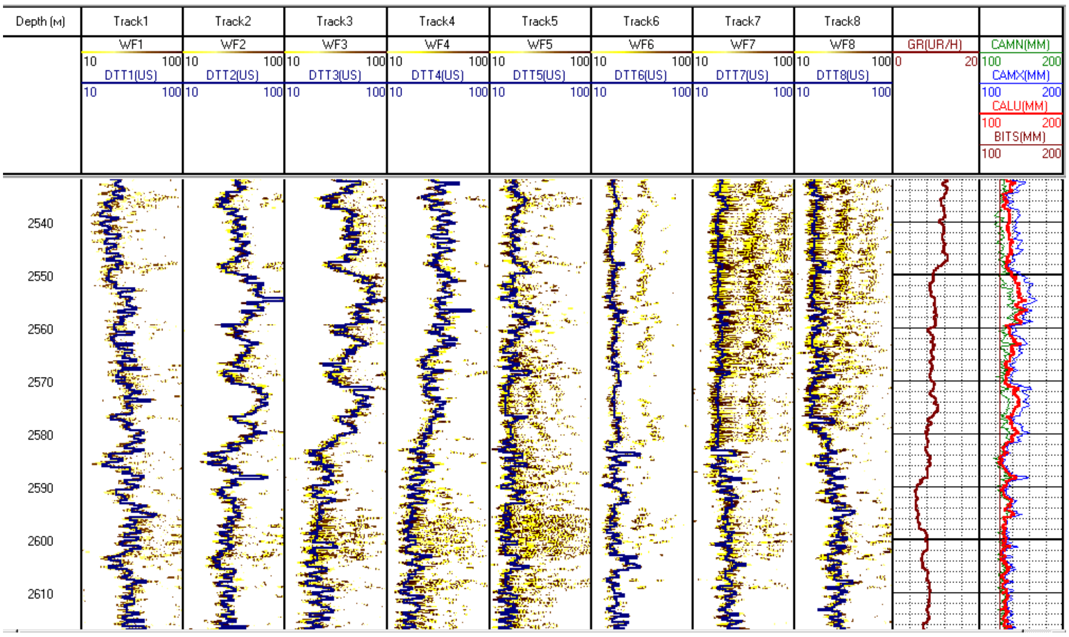


Figure 4. Data processing results.

Geophysical exploration of wells is currently an integral part of geological, drilling, and operational work carried out in the exploration and development of oil and gas fields.

Due to its relative simplicity, oil and gas geology can come to the right conclusions, relying not so much on the study of the core, which is always scarce, but on the materials of geophysical studies. However, geologists and oilmen constantly face one intractable problem, practically unknown to mine geologists. This is the problem of studying the hydrodynamic connectivity of geological bodies (intraobject and interobject, lateral and vertical), traceable at various hierarchical levels—from the level of the sample to the level of the sedimentary cover.

To solve this problem, oil and gas geology has such sources of information as a variety of geophysical and hydrodynamic studies of wells, rich materials of exploration geophysics, data on the history of the development of specific deposits, and special methods for direct study of the hydrodynamic connectivity of productive formations.

Solving the above problems through the introduction of visualization will help to change the situation in the desired direction. According to Jia and Wang, visualization will contribute to the integration of research, methodological, algorithmic, and software developments carried out by geologists, geophysicists, and developers of hydrocarbon deposits, and the integration itself will lead to the accelerated development of methods, technical means and techniques, logical and mathematical apparatuses, algorithms, and software used in the study of the hydrodynamic connectivity of specific geological bodies.

Data visualization is a visual representation of arrays of various information. The simplest and therefore the most common visualization methods are graphs and 3D models.

Visualization goals are the realization of the main idea of information; this is what the selected data need to be shown for and what effect needs to be achieved. After defining the purpose of the visualization, the data type needs to be defined. They can be very heterogeneous in their type and structure, but in the simplest case, they are distinguished between continuous numerical and temporal data, discrete data, and geographical and logical data.

An effective method for studying the saturation of the near-wellbore zone of productive formations and the technical condition of production wells is wave acoustic logging, the wave signals of which in cased wells not only contain information about the state of the string, but also make it possible to evaluate the acoustic properties of productive formations. In the mode of multiple acoustic studies of objects, the information component of the wave signals is determined by comparing the background and control measurements.

Thus, it becomes possible to create a technology for determining the current position of interfluid contacts, including an information base and a scheme for repeated acoustic studies of cased wells.

The practical implementation of repeated acoustic studies using the VAC method in production conditions is quite possible since most geophysical enterprises have fairly modern equipment such as vibration control equipment and noise control equipment, as well as digital full wave signal recorders.

The technology has been tested in production wells of oil and gas fields in Western Siberia.

2.1. Requirements for the Development of a Software Module

When developing the software, the following requirements were set:

- The possibility of functioning regardless of the brand of the data acquisition device;
- Display of caverns of various diameters using a color palette;
- The ability to draw wells of any length;
- The ability to display information on the monitor screen;
- Ensuring high accuracy of visualization.

The software requirements presented above can be implemented by selecting development tools that meet these requirements.

2.2. Development Goals and Objectives

The purpose of the development is to interpret acoustic profiler data through 3D modeling by developing a software tool for visualizing a mine (wellbore) during industrial drilling of oil wells to clarify the geological section of the well and identify reservoir layers in it.

To achieve this goal, the following tasks are solved:

- Selection and implementation of algorithms for drawing the walls of the well, including the conversion of data coming from the device;
- Selection of a high-quality and practical data reading device;
- Selection of tools for creating a color palette of a 3D model and interacting with it;
- Development of an interface for interacting with the developed well using the selected tool;
- The ability to identify the formation of troughs or vugs, as well as a safe place to set the packer.

A review of the literature on this issue showed that similar work is underway to create a database of mineshafts. For example, in the laboratory of aerology and thermal physics of the Mining Institute of the Ural Branch of the Russian Academy of Sciences, an information–analytical system “Shaft Passport” has been created, which is designed for the following:

- Monitoring the current state of the wellbore;
- Creating a database on the construction, operation, and repair of mine shafts;
- Analytical data processing (velocity of lifting vessels, profiling).

According to the available data from the work [10], this base works only in the sweep mode, and information in plan and visualization in a three-dimensional environment is not available in it. In addition, Sight-Power implements a set of modules designed to solve problems that arise during the construction of mine shafts, which is called Shaft Builder.

The information obtained during the drilling of the shaft is unique and should be used in the future. Thus, the created information–analytical system of the mine shaft should be part of the mining and geological system of the enterprise. Only in this case, the maximum use of the received and available information is possible. In this case, it is advisable to use software already available to the subsoil user, in which the mining and geological system of the enterprise is created, and not to buy new software to solve a one-time task.

In the course of searching for a solution under the given conditions, it was revealed that for each job, an application with the smallest value of the objective function (proposed cost and terms of performance of the work) is selected from the existing ones, which confirms the correctness of the operation and software implementation of the algorithm [10]. That is why, in this work, two types of data processing from the recorder were used to calculate the well radii: spectral processing and threshold processing.

2.3. Primary Spectral Processing

Taking into account the fact that the frequency of the emitted high-frequency signal practically does not change upon reflection from the borehole wall, and the first reflection has the maximum amplitude in the wave pattern, it is convenient to use the spectral representation of the signal to determine the time of its arrival. The arrival time is determined by a sliding time window, in which the amplitude spectra are calculated and the maximum of their maximum values is found in the frequency range corresponding to the filter bandwidth. The search starts from the block time. The DTT time count corresponds to the beginning of the window in which the maximum spectra were found.

2.4. Primary Thresholding

This method implements the principle of determining the time of the first extremum of a given sign, the amplitude of which exceeds the discrimination level, and the entry time is the value of the blocking time. Phase sign, discrimination level (threshold), and time blocking are set in the table of technical parameters.

Before processing, it is necessary to set the discrimination level and delay for the measurement and reference channels. The discrimination level must exceed the noise level on all channels. In the VC field, the discrimination level is indicated in blue on the wave pattern, and the delay is indicated by a purple vertical line. Radius and diameter curves are shown in Figure 5.

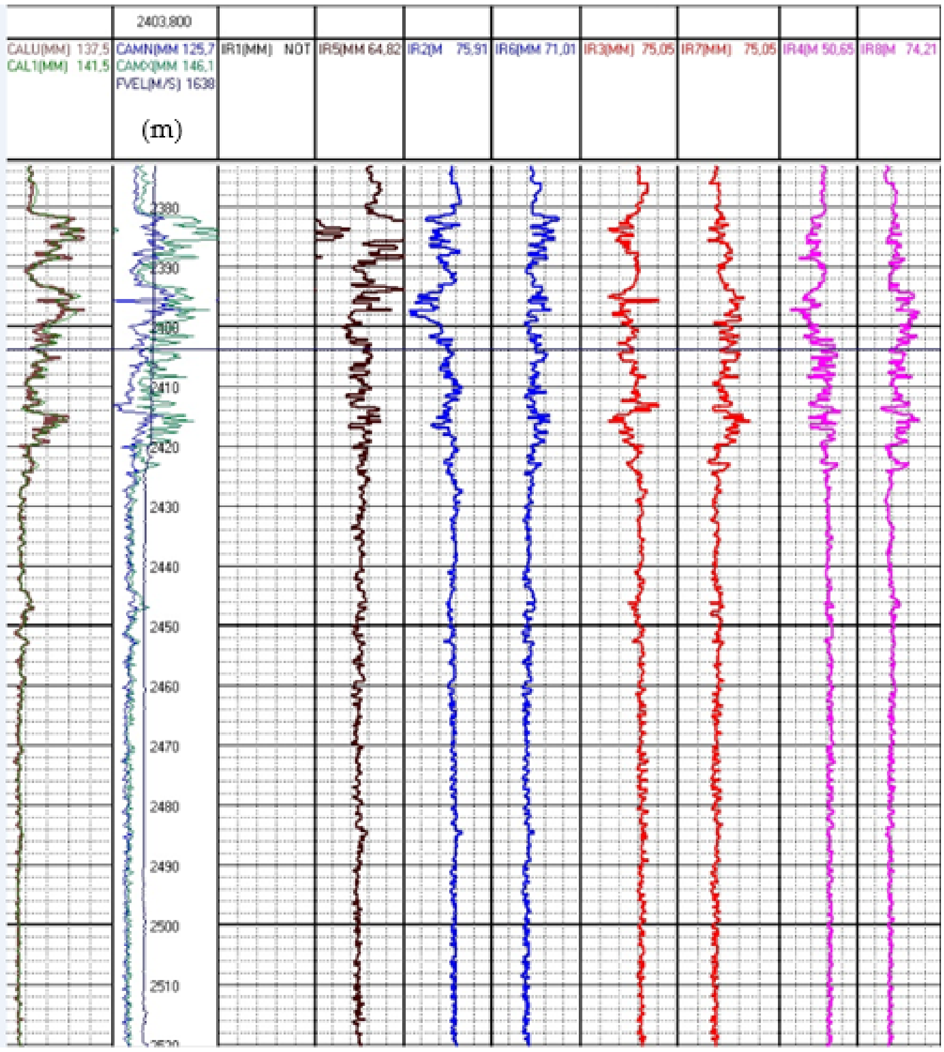


Figure 5. Radius and diameter curves.

The advantage of this method is the visibility of the actions performed, which facilitates the adjustment of processing parameters in the intervals of failures. The disadvantage is the need for interval-by-interval selection of the level of discrimination in cavernous wells with highly indented walls [11].

Processing is carried out for each channel individually or for all channels at once. To do this, in the table of technical parameters, the channel number is set in the line "channel number (0-all channels)", and this channel is visualized on the screen.

2.5. Definition of Radii

The speed of sound in a liquid is determined by the time of arrival of the reflected signal at the calibration transducer (channel 9).

The radii are calculated using the arrival times for the eight measuring channels, the speed of sound in the flushing liquid, and the known distances from the instrument axis to the radiating surfaces of the electro-acoustic transducers (Figure 6).

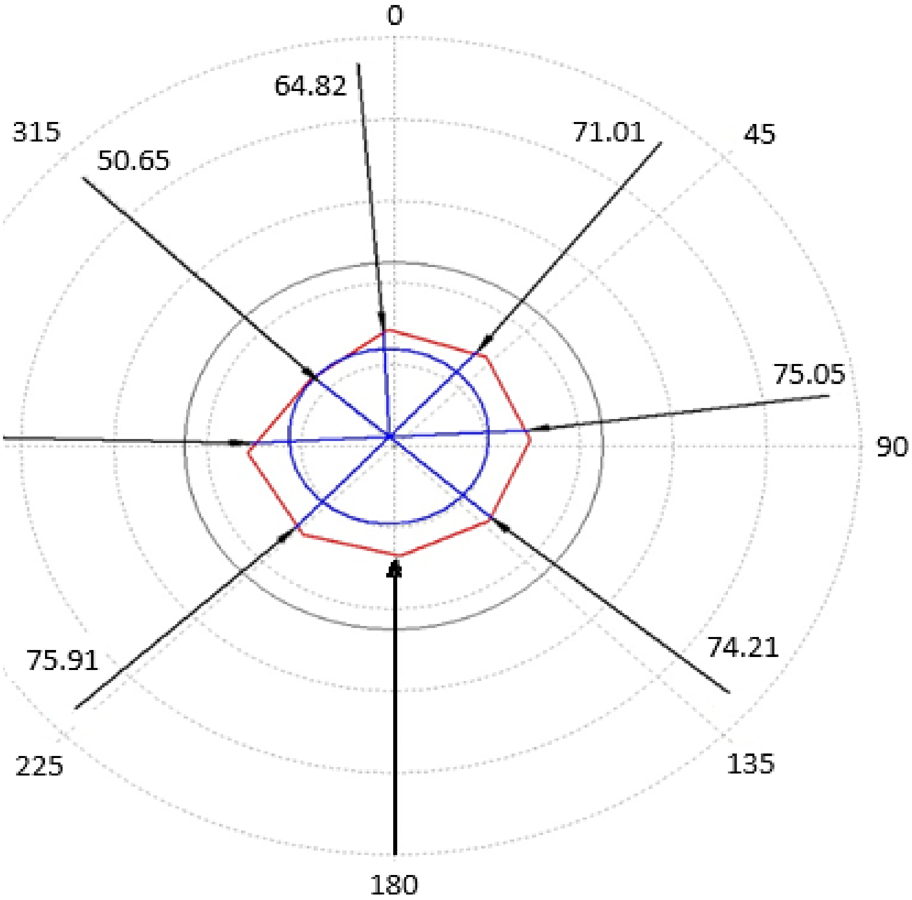


Figure 6. Obtaining radii from 8 points.

Further, the received data are converted by the LogPWin service module.

The blue contour shows the position of the lowered tool relative to the walls of the well, and the red contour is the profile of the well obtained by eight sensors by means of wave reflection.

In instrumentation, computer visualization is of particular importance—a kind of technical visualization based on the capabilities of computer graphics and information technology.

Visualization tools work with the following types of data:

- One-dimensional data—one-dimensional arrays, time series, etc.;
- Two-dimensional data—two-dimensional arrays, coordinates, etc.;
- Multidimensional data—multidimensional arrays, experimental results;
- Texts and hypertexts—articles, reports, web documents, etc.;
- Hierarchical and related data—the structure of subordination in the organization, electronic correspondence of people, hyperlinks of documents, etc.;
- Algorithms, programs, information flows, etc.

To visualize the listed data types, various visual images, methods for their creation, and software products are used. You can map visualization views to different applications (Table 1).

Table 1. Types of visualization and application areas.

Types of Visualization	Application Area
Illustrations	Show existing visualization
Images	Evoke an attitude, emotion, association
Schemes, graphs, diagrams, algorithms	Show quantitative and qualitative relationships, structure of objects
Tables	Structure information
Selecting objects	Focus attention

For the visual presentation of scientific and technical information, various software tools are used:

- Adobe editor family packages;
- Graphic editors (CorelDraw, Adobe PhotoShop, Paint, PhotoEditor, 3D Studio, etc.);
- Spreadsheet processors (Excel, etc.) and Matlab and Mathcad software packages (creation of interactive documents with calculations and visual support);
- LabVIEW (data processing and visualization);
- Various development environments (MS Visual Studio, QT Creator, IDLE).

Based on the analysis of the works of S. Bhat, Ibraheem Alhashim, Peter Wonka, Julia El Zini, Yara Rizk, M. Awad, Nikita Klyuchnikov, A. Zaytsev, D. Koroteev, A. Gruzdev, and G. Ovchinnikov, Visual Studio Professional 2018, the best all-in-one IDE for .NET and C++ developers on Windows, was chosen to develop this program. It contains a complete set of tools and features to enhance every phase of software development. Visual Studio includes a source code editor with support for IntelliSense technology and the ability to easily refactor code. The built-in debugger can work as either a source-level debugger or a machine-level debugger. Other embeddable tools include a form editor to simplify the creation of an application's GUI, a web editor, a class designer, and a database schema designer [12–14].

Based on the work [15], Visual C++ was chosen as the programming language because it supports object-oriented programming (OOP), provides high speed, and has the ability to work with data at a low level.

Visual C++ and Visual Studio are tightly coupled, supporting C++11, C++14, and many C++17 features with the benefits of industry-leading performance, faster builds, and high security. The code supports generic lambda expressions, resumable functions, decltype (auto), constexpr extended expressions, C++ attributes, fold expressions, noexcept in the type system, built-in variables, and other modern features. The .NET Framework 4.5 class library includes significant language and platform enhancements to C++ that make it easier to build, mix control flow in synchronous code, improve user interface responsiveness, and improve web application scalability. The .NET Framework 4.5 also introduces significant improvements in other functional areas, such as OpenGL.

Thanks to Xiao Liu, Fanjin Zhang, Zhenyu Hou, Zhaoyu Wang, Li Mian, Jing Zhang, Jie Tang, Alina Rogulina, A. Zaytsev, L. Ismailova, Dmitry Kovalev, K. Katterbauer, A. Marsala, and F. Abu-Abed, OpenGL was chosen as the API (Open Graphics Library), which is a specification that defines a programming interface for writing applications that use two-dimensional and three-dimensional computer graphics (includes over 300 functions for drawing complex 3D scenes from simple primitives) [16–18].

The OpenGL standard, with the advent of new technologies, allows adding functionality to the library through the extension mechanism. There are a number of libraries built on top of or in addition to OpenGL. For example, the GLU library, which is almost a standard addition to OpenGL and always accompanies it, is built on top of OpenGL; that is, it uses its functions to implement its capabilities.

To confirm the independence of the programming language, various options for binding (binding) OpenGL functions have been developed or completely ported to other languages. There are also OpenGL binding options for C++ and C#. GLM (OpenGL Mathematics) is a utility library that provides C++ programmers with classes and functions to perform mathematical operations. The library can be used when creating 3D programs using OpenGL. Works of scientists Xingye Liu, Guangzhou Shao, Yuwei Liu, Xiwu Liu, Jingye Li, Xiaohong Chen, Y. Chen, Cyrus Ashayeri, and B. Jha made it possible to compile a comparison table between OpenGL and Vulcan [19,20]. A comparison of OpenGL and Vulcan is shown in Table 2.

Table 2. Data in a text document.

Depth (m)	R1	R2	R3	R4	R5	R6	R7	R8
2287.00	98.20	94.27	98.23	97.25	96.94	96.15	97.02	96.36
2286.80	98.19	94.26	98.22	97.24	96.93	96.14	97.01	98.35
2286.60	95.36	100.46	96.12	98.35	98.56	96.83	98.16	98.89
2286.40	94.41	99.50	97.29	97.66	98.13	97.12	97.87	99.68
2286.20	96.74	97.68	98.48	97.95	98.62	96.05	99.27	99.45
2286.00	95.50	97.65	96.19	96.44	96.48	96.71	96.94	96.23
2285.80	95.20	97.48	95.64	97.13	94.89	97.38	96.92	95.67
2285.60	96.91	97.88	97.18	99.32	94.11	100.00	98.16	96.57

2.6. Implementation of the Borehole Wall Rendering Algorithm

In many areas of applications, such as computer-aided design systems for machine building, natural graphic primitives, in addition to line segments and text lines, are also conic sections, i.e., circles, ellipses, parabolas, and hyperbolas. The most commonly used primitive, of course, is the circle. To solve this problem, it is required to develop a method for constructing a circle on an integer coordinate grid using the coordinates of the center and radius. One of the simplest and most efficient circle generation algorithms was developed by Bresenham [21–24].

For simplicity and without loss of generality, consider the generation of a 1/8 circle whose center lies at the origin. The remaining parts of the circle can be obtained by successive reflections (using the symmetry of points on the circle about the center and coordinate axes).

A circle centered at the origin is described by the following equation:

$$X^2 + Y^2 = R^2 \tag{4}$$

Bresenham's algorithm generates successive points of the circle step by step, choosing at each step to enter the pixel raster point P_i (X_i, Y_i) closest to the true circle so that the error

$$E_i(P_i) = (X_i^2 - Y_i^2) - R^2 \tag{5}$$

was minimal. Moreover, as in the Bresenham algorithm for generating segments, the choice of the nearest point is performed by analyzing the values of the control variables, the calculation of which does not require real arithmetic. To select the next point, it is enough to analyze the signs. Consider generating a 1/8 circle clockwise starting from the point X = 0, Y = R.

Let us analyze the possible options for entering the i + 1st point, after entering the i-th (Figure 7).

When generating a circle clockwise after entering the point (Xi, Yi), the next point can be (see Figure 7a) Pg = (Xi + 1, Yi), horizontal movement; Pd = (Xi + 1, Yi − 1), diagonal movement; or Pv = (Xi, Yi−1), vertical movement. For these possible points, we calculate and compare the absolute values of the differences in the squares of the distances from the center of the circle to the point and the circle:

$$|D_g| = \left|(X+1)^2 + Y^2 - R^2\right| \tag{6}$$

$$|D_d| = \left|(X+1)^2 + (Y-1)^2 - R^2\right| \tag{7}$$

$$|D_v| = \left|X^2 + (Y-1)^2 - R^2\right| \tag{8}$$

The point for which this value is minimal is selected and entered.

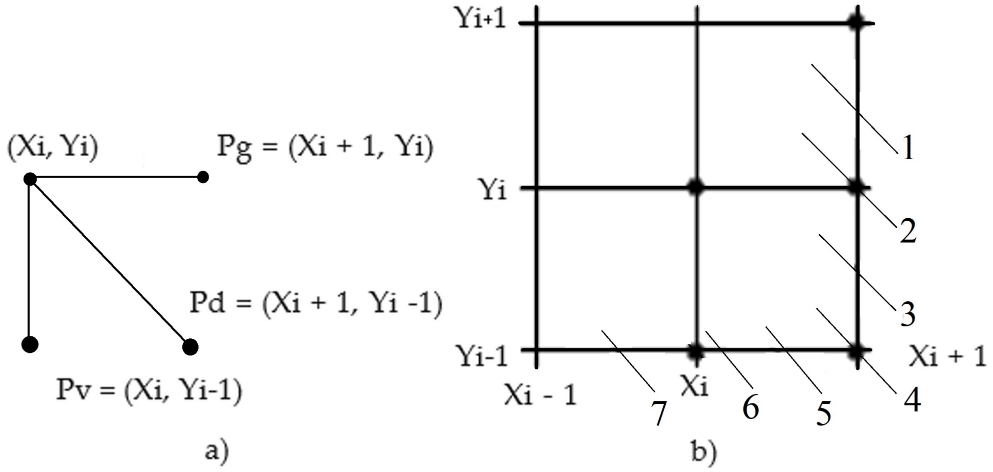


Figure 7. Options for the location of the next pixel of the circle: (**a**) Choosing a possible way to calculate the next step; (**b**) Selection of possible point movement.

The choice of calculation method is determined by the value of Dd. If Dd < 0, then the diagonal point is inside the circle. These are options 1–3 (see Figure 7b). If Dd > 0, then the diagonal point is outside the circle. These are options 5–7. Finally, if Dd = 0, then the diagonal point lies exactly on the circle. This is option 4. Consider the cases of different values of Dd in the sequence just given.

Case Dd < 0

Here, either horizontal (Pg) or diagonal (Pd) can be chosen as the next pixel.

To determine which pixel to choose, Pg or Pd, we determine the difference

$$d_i = |D_g| - |D_d| \tag{9}$$

We will choose the point Pg for di J 0; otherwise, we will choose Pd.

Consider the calculation of di for different options.

For options 2 and 3:

Dg i 0 and Dd < 0, since the horizontal pixel is either outside or on the circle, while the diagonal one is inside.

$$d_i = (X+1)^2 + Y^2 - R^2 + (X+1)^2 + (Y-1)^2 - R^2 \tag{10}$$

Adding and subtracting $(Y-1)^2$, we obtain

$$d_i = 2[(X+1)^2 + (Y-1)^2 - R^2] + 2Y - 1 \tag{11}$$

Dd is in square brackets, so di = 2(Dd + Y)−1.

For option 1:

It is clear that a horizontal pixel Pg must be chosen. A check of the di components shows that Dg < 0 and Dd < 0, and di < 0, since the diagonal point is more distant from the circle; i.e., according to the criterion di < 0, as in the previous cases, one should choose a horizontal pixel Pg, which is correct.

Case Dd > 0

Here, either diagonal (Pd) or vertical (Pv) can be chosen as the next pixel.

To determine which pixel to choose, Pd or Pv, we determine the difference

$$S_i = |D_d| - |D_v| \tag{12}$$

If si ≤ 0, then the distance to the vertical point is greater and we need to choose the diagonal pixel Pd; if si > 0, then we choose the vertical pixel Pv.

Consider the calculation of si for different options.

For options 5 and 6:

Dd > 0 and Dv ≤ 0, since the diagonal pixel is outside, while the vertical pixel is either outside or on the circle.

$$S_i = (X+1)^2 + (Y-1)^2 - R^2 + X^2 + (Y-1)^2 - R^2 \tag{13}$$

Adding and subtracting (X+1)^2, we obtain

$$S_i = 2[(X+1)^2 + (Y-1)^2 - R^2] - 2X - 1 \tag{14}$$

Dd is in square brackets, so si = 2(Dd−X)−1.

For option 7:

It is clear that the vertical pixel Pv must be chosen. Checking the si components shows that Dd > 0 and Dv > 0, and si > 0, since the diagonal point is more distant from the circle; i.e., according to the criterion si > 0, as in the previous cases, one should choose a vertical pixel Pv, which corresponds to the choice for options 5 and 6.

Case Dd = 0

For the components di we have Dg > 0 and Dd = 0; therefore, according to the criterion di > 0, we choose a diagonal pixel.

On the other hand, for the components si we have Dd = 0 and Dv < 0, so according to the criterion si ≤ 0, we also choose a diagonal pixel.

So:

Dd < 0

di J 0—choice of horizontal pixel Pg

di > 0—selection of the diagonal pixel Pd

Dd > 0

si ≤ 0—choice of diagonal pixel Pd

si > 0—selection of vertical pixel Pv

Dd = 0

selection of the diagonal pixel Pd.

Let us derive recurrence relations for calculating Dd for the (i + 1)-th step, after the execution of the i-th.

1. For a horizontal step to X_{i+1}, Y_i,

$$X_{i+1} = X_i + 1 \tag{15}$$

$$Dd_{i+1} = ((X)_{i+1} + 1)^2 - (Y)_{i+1} + 1)^2 - R^2 = Dd_{i+1} + 2X_{i+1} + 1 \tag{16}$$

2. For a diagonal step to X_{i+1}, Y_{i-1},

$$X_{i+1} = X_i + 1 \tag{17}$$

$$Dd_{i+1} = Dd_i + 2X_{i+1} - 2Y_{i+1} + 2 \tag{18}$$

3. For a vertical step to X_i, Y_{i-1},

$$X_{i+1} = X_i \tag{19}$$

$$Dd_{i+1} = Dd_i - 2Y_{i+1} + 1 \tag{20}$$

Construction of a circle by 8 points is shown in Figure 8.

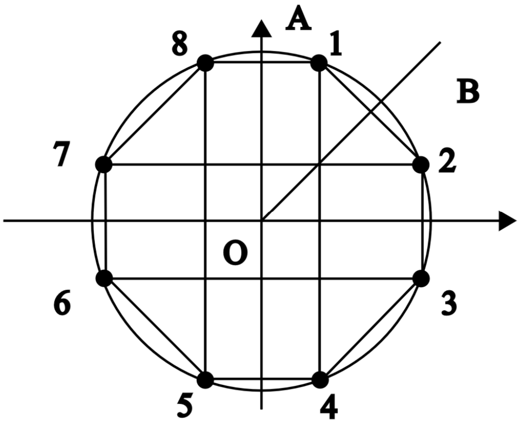


Figure 8. Construction of a circle by 8 points.

The subroutine Vector2 is given, which implements the algorithm described above and builds an arc of a circle in the first quadrant. Pixels in other quarters can be obtained by reflection. In addition, thanks to the works of A.D. Chernikov, N. Eremin, V. Stolyarov, A. Sboev, O.K. Semenova-Chashchina, and L.K. Fitsner, it is enough to form an arc only in the second octant and form the remaining pixels for symmetry reasons, for example, with the help of a Vertex structure that shifts symmetrical pixels clockwise from the source [25,26]. The mine model based on the Bresenham method is shown in Figure 9.

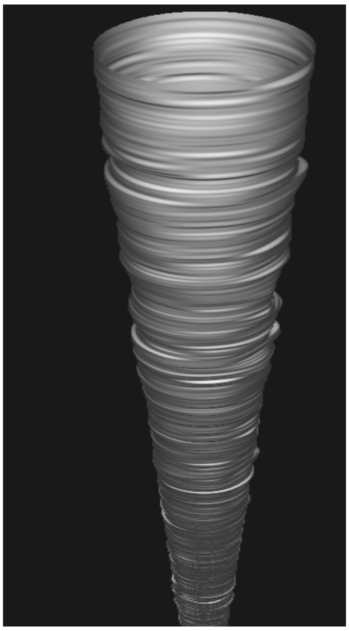


Figure 9. Obtaining a mine model based on the Bresenham method.

The created information and analytical system differs significantly from traditional geological, mine surveying, or mining projects of geophysical research of wells. It requires a specialized geoinformation framework. The software allows collapsing the data obtained on the sweep into a three-dimensional model. Recalculation of coordinates from a cylindrical scan to geographic form and vice versa also does not cause difficulties.

3. Results

3.1. Rationale for Choosing a Color Palette

Based on the works of M. Aly, A. Ibrahim, S. Elkatatny, and A. Abdulraheem, in general, the algorithm for painting an arbitrary contour, which is already drawn in a raster, involves the following actions:

- Find a pixel inside the contour of the figure;
- Change the color of this pixel to the desired fill color;
- Analyze neighboring pixels;
- If the color of some neighboring pixel is not equal to the color of the contour border or the fill color, then the color of this pixel changes to the fill color;
- Analyze the color of the pixels adjacent to the previous one, and so on, until all the pixels inside the outline are repainted in the fill color.

Contour pixels are a boundary that cannot be exceeded during the sequential enumeration of all neighboring pixels. Only four pixels (right, left, top, bottom—four-connected) or all eight (eight-connected) can be considered neighbors (Figure 10). Not every contour can serve as a border for painting [27].

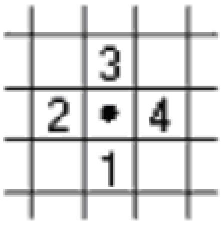


Figure 10. Neighbor pixel enumeration order.

For all painting algorithms, it is necessary to set the starting point with x, y coordinates inside the outline. The simplest algorithm can be described as follows:

```
void FillThePixel(int x, int y, int borderQ, int fillQ)
{ // 4-connected variant
int b=getThePixel(x,y); // pixel color
if ((b!=borderQ)&&(b!=fillQ)) // not bordered or filled?
{setThePixel(x,y,fillQ); // paint and "look around"
FillThePixel (x + 1, y, borderQ, fillQ); //Recursion
FillThePixel (x, y + 1, borderQ, fillQ); //
FillThePixel (x − 1, y, borderQ, fillQ); //
FillThePixel (x, y − 1, borderQ, fillQ); } //
}
```

This procedure definition is recursive. Recursion makes it easier to write some algorithms. However, for this algorithm, recursion creates significant problems—recursive calls to the FillPixel procedure are made for each pixel, which usually leads to a stack overflow during the execution of a computer program. The area bypass example is shown in Figure 11.

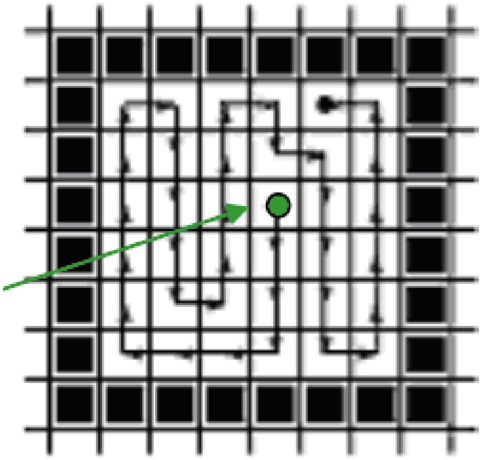


Figure 11. Area bypass example.

Practice shows that this algorithm is unacceptable for figures with an area of a thousand or more pixels. The wave algorithm is more rational (see Figure 12). To store the pixel coordinates of the current diamond-shaped wavefront, it uses dynamic arrays of up to 10,000 elements.

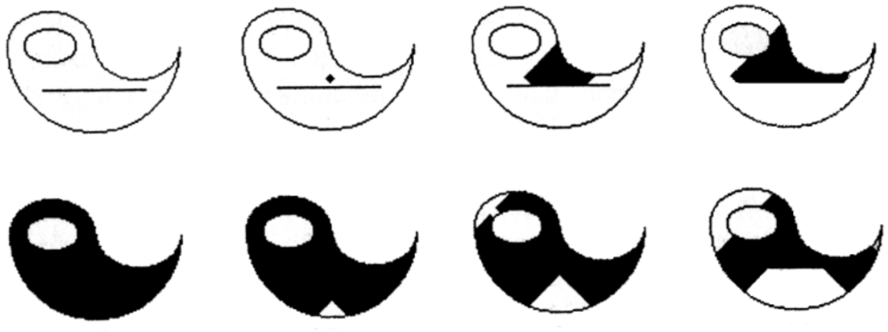


Figure 12. Wave algorithm.

For an initial point (a vertex on the graph), there are neighboring points (other vertices) that meet two conditions: first, these vertices are connected to the initial one; secondly, they have not yet been marked, i.e., considered for the first time. The neighboring vertices of the current iteration are marked in the vertex description array, and each of them becomes the current point for searching for new neighboring vertices in the next iteration. If each vertex is marked with an iteration number in a special array, then when the end point is reached, it is possible to perform a reverse cycle—from the end point to the start point in descending order of iteration numbers. During the reverse cycle, we can find all the shortest paths (if there are several) between two given points on the graph.

Regarding the shading of raster figures, the vertices of the graph are pixels, and the marking of passed pixels is done directly in the raster with the color of the shading. This almost completely repeats the idea of the simplest algorithm, but recursion is not used here, which causes a different sequence of processing pixels when painting.

From the starting point, a wave of filling pixels propagates in the form of a rhombus. In one cycle, pixels are painted along the perimeter line of the rhombus (or several rhombuses, depending on the complexity of the figure).

It should be noted that this algorithm is not the fastest of the known filling algorithms. Algorithms that process not individual pixels, but blocks of many pixels at once, which form rectangles or lines, provide a high filling speed.

The line-filling algorithm is widely used in computer graphics. It differs from the simplest recursive algorithm given earlier in that at each step the horizontal line between the pixels of the contour is painted over and unfilled pixels above and below the line are detected. The algorithm is recursive, but the recursion depth is proportional only to the number of pixels in the line. In the case of creating a stack, it is enough to store one pixel for each painted area. The algorithm is recursive, but since the function is called for the line, and not for each individual pixel, the number of nested calls decreases in proportion to the length of the line. This reduces the load on the computer’s stack memory. In the works of Liqun Shan, Yanchang Liu, M. Tang, Ming Yang, and B. Xueyuan, the algorithm looks as follows:

```
int lineFill(int x, int y,int dy,int XL, int XR)
{ int xleft=x, xright=x; int c; //left border of the current horizontal
do {xleft–; c=getPixelll(xleft,y):} while (b!=BORDERER);
do{xright++; c=getPixelll(xright,y):} while (b!=BORDERER);
xleft++; right–;
LineX(xleft,y,xright,y,fillCC); // paint over horizontal
//Recursions: 1) checking and painting over the horizontal
for (x=xleft; x<=xright; x++) {c=getPixelll(x,y+dy);
if (b!=BORDER) x=FillLine (x,y+dy,dy,xleft,xright);}
```

```
// 2) looked back and painted over to the left of the previous left
// and to the right of the previous right border
for (x = xleft; x <= XL; x ++) { c = getPixelll (x, y − dy);
if (b!=BORDERER) x= FillLine(x,y-dy,-dy,xl,xr);}
for (x=XR; x<=xright; x++) {c=getPixelll(x,y-dy);
if (b!=BORDERER) x= FillLine(x,y-dy,-dy,xleft,xright);}
return xright; }
```

Appeal in seed: FillLine (xWt,yWt,xQt,xQt);

The principle of operation of the RenderThread procedure of the final qualification work is based on this algorithm [28]. The line fill method is shown in Figure 13; the distribution of colors on the model is shown in Figure 14.

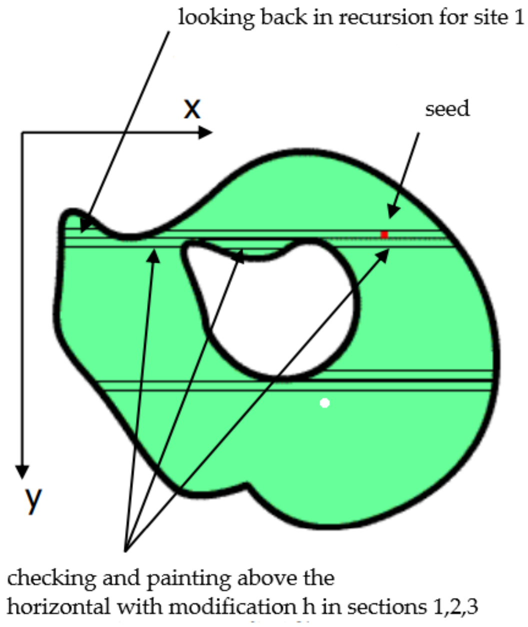
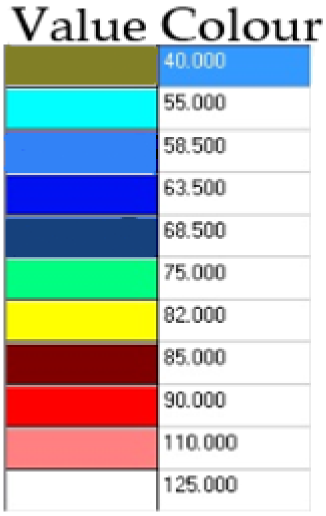

looking back in recursion for site 1
x
seed
y
checking and painting above the
horizontal with modification h in sections 1,2,3

Figure 13. Line fill method.



Value	Colour
🟫	40.000
🟦	55.000
🟦	58.500
🟦	63.500
🟦	68.500
🟩	75.000
🟨	82.000
🟥	85.000
🟥	90.000
🟥	110.000
⬜	125.000

Figure 14. Distribution of colors on the model.

The color palette of the resulting model is put down manually and depends on the preferences of the specialist working with the program. In this case, based on an analysis of the work of A. Abdulraheem, the following filling was chosen [29,30]:

It should be noted that areas of increased danger (red color) signal the direction of development of caverns. The experience of processing data from an acoustic caliper in more than a hundred wells in various geological and technical conditions has shown the reliability of determining the geometric dimensions of the profiles (well model in color is shown in Figure 15). Limitations may occur when working in heavy drilling fluids (density above 1.3 g/cm^3) and cavern sizes larger than 2D. A large volume of measurements of horizontal well profiles using the program can reveal some patterns.

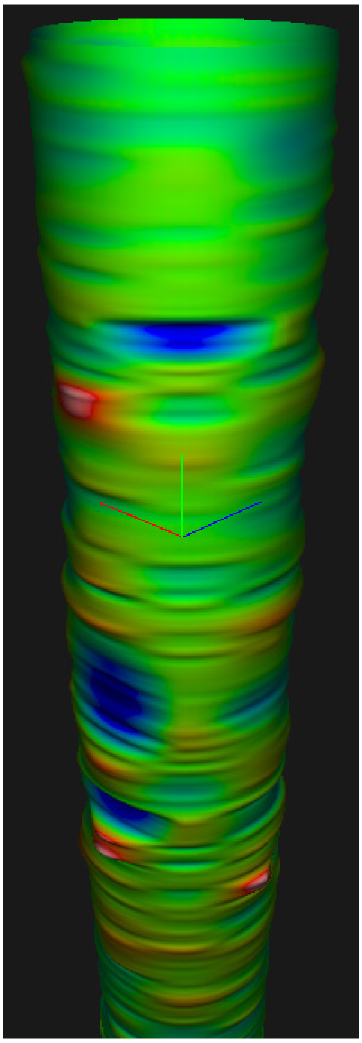
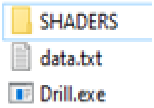


Figure 15. Well model in color.

The project consists of the SHADERS folder, the data text document, and the Drill application (Figure 16).



SHADERS
data.txt
Drill.exe

Figure 16. Source project folder.

The SHADERS folder contains a package of shaders for drawing a well model in 3D format.

First, you need to launch the Drill application. Drag and drop .txt data to an open Drill display. We evaluate the resulting well model. The ease of using the application is an integral plus for the user and was realized in the analysis of the works of Shanshan Lium and Zhiming Wang [31].

The program for constructing a circle by eight points is implemented in the ConstructGeometry function:

```
void ConstructGeometry(const Array<f32> values, Array<Vertex> vertices)
{
s64 slices = values.Length/9;
static Vector3 directions [8] =
{
RotateY({ 1, 0, 0 }, MATH_2PI / 8.0 * 0),
RotateY({ 1, 0, 0 }, MATH_2PI / 8.0 * 1),
RotateY({ 1, 0, 0 }, MATH_2PI / 8.0 * 2),
RotateY({ 1, 0, 0 }, MATH_2PI / 8.0 * 3),
RotateY({ 1, 0, 0 }, MATH_2PI / 8.0 * 4),
RotateY({ 1, 0, 0 }, MATH_2PI / 8.0 * 5),
RotateY({ 1, 0, 0 }, MATH_2PI / 8.0 * 6),
RotateY({ 1, 0, 0 }, MATH_2PI / 8.0 * 7),
};
// fill points
for (s64 i = 0; i < slices; i++)
{
f32 height = values.Data[i * 9 + 0];
// surface points from input data
for (s64 n = 0; n < 8; n++)
{
f32 value = values.Data[i * 9 + n + 1] * 0.001f; // mm!
Vector3& v = vertices.Data[i * (8 + 8) * 3 + (n * 2 + 0)].Position;
v = directions[n] * value;
vY=height;
}
// B2 ring control points
for (s64 n = 0; n < 8; n++)
{
Vector3& l = vertices.Data[i * (8 + 8) * 3 + ((n * 2 + 0) & 0b1111)].Position;
Vector3& v = vertices.Data[i * (8 + 8) * 3 + ((n * 2 + 1) & 0b1111)].Position;
Vector3& r = vertices.Data[i * (8 + 8) * 3 + ((n * 2 + 2) & 0b1111)].Position;
v = Weird(l, r);
vY=height;
}
}
// endpoints
constexpr f32 delta = 0.4f;// [0.0f, 0.5f]
{
// top
for (s64 n = 0; n < (8 + 8); n++)
{
Vector3& t = vertices.Data[0 * (8 + 8) * 3 + n].Position;
Vector3& b = vertices.Data[1 * (8 + 8) * 3 + n].Position;
Vector3& o = vertices.Data[0 * (8 + 8) * 3 + 1 * (8 + 8) + n].Position;
o = t + delta * (b−t);
}
// bottom
for (s64 n = 0; n < (8 + 8); n++)
{
```

```
Vector3& t = vertices.Data[(slices—2) * (8 + 8) * 3 + n].Position;
Vector3& b = vertices.Data[(slices—1) * (8 + 8) * 3 + n].Position;
Vector3& o = vertices.Data[(slices—2) * (8 + 8) * 3 + 2 * (8 + 8) + n].Position;
o = b + delta * (t−b);
}
}
// B3 control vertical control points
for (s64 i = 1; i < (slices—1); i++)
{
for (s64 n = 0; n < 8 + 8; n++)
{
Vector3& t = vertices.Data[(i − 1) * (8 + 8) * 3 + n].Position;
Vector3& m = vertices.Data[(i + 0) * (8 + 8) * 3 + n].Position;
Vector3& b = vertices.Data[(i + 1) * (8 + 8) * 3 + n].Position;
Vector3& u = vertices.Data[(i − 1) * (8 + 8) * 3 + 2 * (8 + 8) + n].Position;
Vector3& d = vertices.Data[(i + 0) * (8 + 8) * 3 + 1 * (8 + 8) + n].Position;
Vector3 dir = b − t;
u = m − 0.5f * delta * dir;
d = m + 0.5f * delta * dir;
}
}
}
```

For better understanding, below is a flow chart of the algorithm for the first octant (Figure 17, drawn by authors).

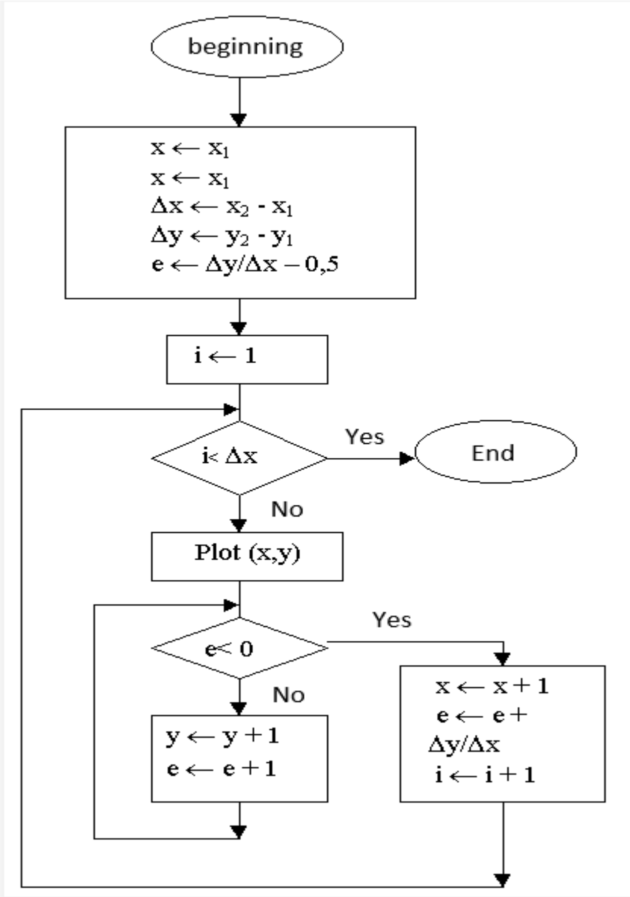


Figure 17. A flow chart.

Thanks to papers [31–34], the algorithm is implemented in such a way that the usage of operational memory is minimized. Due to this, the performance of the software increases significantly.

3.2. Analysis of the Resulting Well

In the works of Jiangbo Xu, Wei Wei, H. Bao, Keke Zhang, H. Lan, Changgen Yan, Weifeng Sun, and Faquan Wu, the use of well visualization provides a complementary and clarifying method for evaluating the results of data processing for more efficient and safe drilling and casing of the wellbore and installation of packers [35].

Oil in oil fields is located in thin channels (capillaries), and productive formations are under pressure, which is called reservoir pressure. The reasons for the presence of reservoir pressure in the reservoirs are mostly associated with the pressure of water and gas, which are in contact with oil (water–oil and gas–oil contacts), as well as with the elastic compression of the rocks of the reservoirs (stresses).

For a number of geological and technical reasons, the actual borehole diameter differs from the nominal diameter, i.e., from the diameter of the bit used. An increase in diameter is usually observed when the well intersects clays, clayey rocks, and salts; a decrease in diameter (as a result of the formation of a clay cake) is observed for permeable sandstones; and nominal diameter is observed for dense sandstones, limestones, and dolomites.

For a three-dimensional representation of the input data, it was necessary to obtain a three-dimensional geometry with the preservation of attribute data. To facilitate and speed up the process of converting data from a plane to 3D, it was necessary to create an additional software module that allows the convolution of flat data into a cylinder for a given center and radius (the well model based on real data is shown in Figure 18, made by authors). It should be noted that the resulting 3D geometry is stored in the true coordinate system, which allows combining the results with other data at any time.

The experience of maintaining the system has shown that there is a deviation from the design solutions during the period of sinking the shaft; this problem is also reflected in the works of Hang Lin, Xing Zhang, R. Cao, and Zhijie Wen. The depth of the caverns changes, and the displacement of the caverns occurs. It is necessary to assess quickly the degree of impact of these changes on the safety of the operation of the structure. Therefore, a mandatory element of such an information model is a three-dimensional visualization of the available data, which allows, based on a graphical method, the comparison of project data with actual results and the identification of deviations from the project and possible errors [36].

Based on the experience of Asra Aslam and M.S. Ansari, it can be said that in horizontal wells, the production string is most often not cemented, and a packer is installed to seal the annulus. The packer is installed above the oil reservoir (in the seal), i.e., as a rule, in shale formations subject to failure. To ensure reliable sealing of the string, the packer should be installed in a well section with a wellbore diameter close to the nominal one and a relatively smooth wall [37].

The task of choosing the location of the packer is the simplest and is solved by analyzing the well profile without the risk of an accident. Thus, according to the figure, the packer will be installed in the places of the green and blue palette.

The main purpose of this model is the selection of troughs or caverns (a trough is understood as any deviation of the shape of the cross-section of the well from a circle, which is characterized by the presence of significantly different mutually perpendicular diameters) on the walls of the well. Various factors affect the process of trough formation: lithological composition of rocks, inclination angle and intensity of well curvature, properties of drilling fluid, drilling technology, number of trips, etc. The troughs are clearly visible in the figure (red palette).

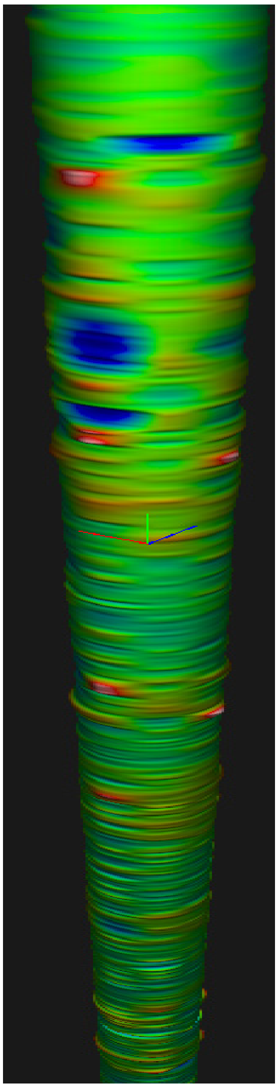


Figure 18. Well model based on real data.

4. Discussion

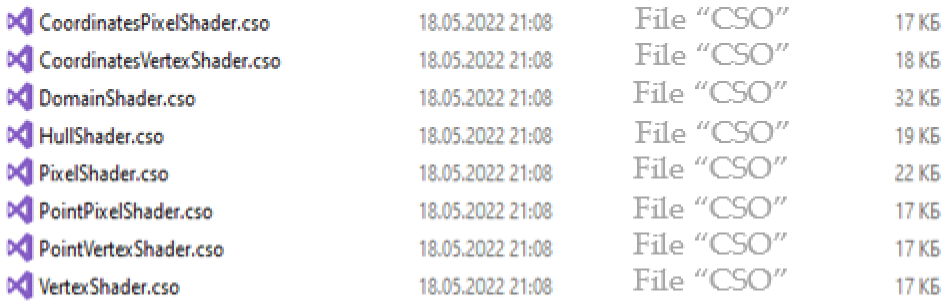
Today, at various complex engineering and technical facilities, emergencies periodically occur that could be avoided if complete and reliable information was available about the structure and the geological stratum being the base of the facility or its environment. The developed model is an analog of BIM modeling and will be able to provide engineers with reliable and up-to-date information at all stages of drilling a well. An obligatory element of this model is three-dimensional data visualization, which makes it possible to carry out data analysis from an acoustic profiler on a new level and timely detect caverns. The paper proposes an approach to three-dimensional data visualization of a technically complex engineering object (a mine shaft) located inside the geological environment using LogPWin.

The software can be divided into three components:

- Acoustic profiler data;
- Required SHADERS shader pack for object modeling;
- A program that directly draws an object.

The text document data contain the data for building the mine. In each line, the first number is the drilling depth, and the remaining eight numbers are the wall radii at that depth (File "CSO").

Based on the experience of scientists Lewis Crawford And Michael O'Boyle, SHADERS contains the minimum set of shaders required to draw a well model in 3D format [38] (shown in Figure 19).



CoordinatesPixelShader.cso	18.05.2022 21:08	File "CSO"	17 КБ
CoordinatesVertexShader.cso	18.05.2022 21:08	File "CSO"	18 КБ
DomainShader.cso	18.05.2022 21:08	File "CSO"	32 КБ
HullShader.cso	18.05.2022 21:08	File "CSO"	19 КБ
PixelShader.cso	18.05.2022 21:08	File "CSO"	22 КБ
PointPixelShader.cso	18.05.2022 21:08	File "CSO"	17 КБ
PointVertexShader.cso	18.05.2022 21:08	File "CSO"	17 КБ
VertexShader.cso	18.05.2022 21:08	File "CSO"	17 КБ

Figure 19. SHADERS content.

Pixel shaders (Pixel in the name) work with bitmap fragments and with textures—they process data associated with pixels (for example, color, depth, texture coordinates). According to the works of Jose Javier Luis Tello, Alfonso Burgos-Risco, and G.R. Reinaldos, the pixel shader is used in the last stage of the graphics pipeline to form the image fragment [39].

Vertex shaders (Vertex is present in the name) operate with data associated with the vertices of polyhedra, for example, with the coordinates of a vertex (point) in space, with texture coordinates, with a vertex color, with a tangent vector, with a binormal vector, and with a normal vector. According to the works of Yuchi Huo, Shi Li, Yazhen Yuan, Xu Chen, Rui Wang, Wenting Zheng, Hai Lin, and H. Bao, the vertex shader is used for view and perspective transformation of vertices, for generating texture coordinates, for calculating lighting, etc. [40]

Geometric shaders (Hull and Domain), unlike vertex shaders, are able to process not only one vertex, but the whole primitive. A primitive can be a segment (two vertices) or a triangle (three vertices) if there is information about adjacent vertices. In the works of J. Hasselgren, J. Munkberg, J. Lehtinen, M. Aittala, and S. Laine, it is noted that the geometry shader is able to generate primitives on the fly (without using the CPU) [41].

The main advantage of the developed software product is its flexibility of application: it can be used in any type of logging for a visual representation of the structure of the resulting mine. It should be noted that the final software product can be used by drilling engineers for more efficient and safe drilling and casing of the wellbore, as well as the allocation of intervals for setting packers. The flexibility of the software is due to the fact that the set of shaders can be expanded, thereby obtaining a more detailed view of the mine. From this, we can conclude that the software will function on a relatively weak computer in terms of performance.

The main drawback of the software is shaders. Based on the work of B. Catanzaro, Renato Farias, and Marcelo Kallmann, the program presents the minimum set that can function on operating systems from Windows 7. If you need to obtain a more detailed idea of the mine, you will need newer shaders that do not have a mechanism for interacting with the old platform. Thus, shader compatibility must be taken into account [42,43].

The developed technical solutions in the future will provide more efficient processing, synthesis, and visualization of data at a fundamentally new qualitative level. The information and analytical system created by the authors will provide not only engineers but also specialists in the technical services of mining enterprises and related scientific and design organizations with reliable and operational information at all stages of the life cycle of the structure.

5. Conclusions

Works on sinking vertical mine shafts are among the most complex and time-consuming in the construction of mines and take 40–50% of the total construction time and up to 15% of the total capital costs. The shaft is the main opening working, the failure of which leads to the shutdown of the entire mining enterprise. During the construction of shafts in difficult mining and geological conditions and an increase in the depth of their implementation, due to the lack of available information, significant complications arise. If there is an up-to-date information and analytical system that allows controlling the quality of construction work quickly and making timely changes to the project when new data appear, some of the above causes of emergencies can be eliminated.

The methodological approach to the engineering–geological modeling of the system is analogous to BIM modeling and allows reducing the risks of emergencies during the driving of mine shafts and their operation.

The developed technical solutions provide efficient processing, synthesis, and visualization of data at a fundamentally new qualitative level.

With the system of acoustic profilometry, it was revealed that in the horizontal sections of some wells, vertical caverns predominate; in others, horizontal caverns predominate. However, there are wells in which there are a large number of inclined caverns. Experts are investigating the cause of the change in the geometry of the wellbore, using both additional data from the well and visualization.

The resulting software product can be used by drilling engineers to more efficiently and safely drive and case a wellbore, as well as allocate intervals for setting packers.

The result of the work done is a program that provides visualization of data coming from an acoustic profiler, which allows obtaining a visual and informative idea of the wellbore. The original project also contains a shader with additional data needed to model the software.

The resulting well model allows us to state some features of cavern formation, indicating the direction of mechanical stresses in rocks, and the use of this program by specialists in the oil and gas sector is promising.

Author Contributions: Conceptualization, F.A.-A. and S.Z.; methodology, F.A.-A.; software, F.A.-A. and K.P.; validation, K.P., V.Z., and S.Z.; formal analysis, S.Z.; investigation, F.A.-A.; resources, F.A.-A.; data curation, K.P. and V.Z.; writing—original draft preparation, V.Z., K.P., F.A.-A., and S.Z.; writing—review and editing, S.Z.; visualization, K.P. and V.Z.; supervision, S.Z.; project administration, F.A.-A. and S.Z.; funding acquisition, S.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhou, Z.; Hu, Y.; Liu, B.; Dai, K.; Zhang, Y. Development of Automatic Electric Drive Drilling System for Core Drilling. *Appl. Sci.* **2023**, *13*, 1059. [CrossRef]
2. Khadisov, M.; Hagen, H.; Jakobsen, A.; Sui, D. Developments and experimental tests on a laboratory-scale drilling automation system. *J. Pet. Explor. Prod. Technol.* **2019**, *10*, 605–621. [CrossRef]
3. Smirnov, N.A.; Varykhalov, A.S. Determination of Horizontal Borehole Geometry by Ultrasonic Pulse-Echo Method (Russian). In Proceedings of the SPE Russian Oil and Gas Technical Conference and Exhibition, Moscow, Russia, 28–30 October 2008. [CrossRef]
4. Dowell, I.A.; Seiler, D.; York, P. Logging While Drilling Borehole Imaging and Dipmeter Device. US Patent 5,899,958, 1999.5.4.
5. Wanasinghe, T.R.; Gosine, R.G.; Petersen, B.K.; Warrian, P.J. Digitalization and the Future of Employment: A Case Study on the Canadian Offshore Oil and Gas Drilling Occupations. *IEEE Trans. Autom. Sci. Eng.* **2023**, *99*, 1–21. [CrossRef]

6. Zeidouni, M. Drilling, Completion, and Monitoring Operations. In *Shale Hydrocarbon Recovery*; Springer International Publishing: Cham, Switzerland, 2023; pp. 11–27. [CrossRef]
7. Isbell, M.; Fonseca, I.S.; Boone, S.; Avdeev, D.; Neal, J.; Farrow, B. Drilling Limiters, Drilling Process Automation, and Drilling Data—Relating Downhole Measurements to Surface Process Automation. In Proceedings of the SPE Annual Technical Conference and Exhibition, Houston, TX, USA, 3–5 October 2022. [CrossRef]
8. Chen, T.; Saxena, S.; Li, L.; Fleet, D.J.; Hinton, G. A Language Modeling Framework for Object Detection. *arXiv* **2019**, arXiv:2109.10852. [CrossRef]
9. Chen, T.; Saxena, S.; Li, L.; Tsung-Yi, L.; Fleet, D.J.; Hinton, G. A Unified Sequence Interface for Vision Tasks. *arXiv* **2022**, *2206*. arXiv:2206.07669.
10. Abu-Abed, F. Automation of work distribution in the management of mining enterprises. *E3S Web Conf.* **2021**, *278*, 01008. [CrossRef]
11. Smirnov, N.A.; Varykhalov, A.; Mukhin, A. *Methodological Guidelines for the Use of AK Equipment*; GERS Publishing House: Moscow, Russia, 2011; p. 266.
12. Bhat, S.; Alhashim, I.; Wonka, P. Depth Estimation Using Adaptive Bins. *arXiv* **2021**, arXiv:2011.14141. [CrossRef]
13. El Zini, J.; Rizk, Y.; Awad, M. A Deep Transfer Learning Framework for Seismic Data Analysis: A Case Study on Bright Spot Detection. *IEEE Trans. Geosci. Remote Sens.* **2020**, *99*, 1–11. [CrossRef]
14. Klyuchnikov, N.; Zaytsev, A.; Koroteev, D.; Gruzdev, A.; Ovchinnikov, G. Data-driven model for the identification of the rock type at a drilling bit. *J. Pet. Sci. Eng.* **2019**, *178*, 506–516. [CrossRef]
15. Romanenkova, E.; Rogulina, A.; Shakirov, A.; Stulov, N.; Zaytsev, A. Similarity learning for wells based on logging data. *J. Pet. Sci. Eng.* **2022**, *215*, 110690. [CrossRef]
16. Abu-Abed, F. Development of three-dimensional models of mining industry objects. *E3S Web Conf.* **2021**, *278*, 01002. [CrossRef]
17. Liu, X.; Zhang, F.; Hou, Z.; Wang, Z.; Mian, L.; Zhang, J.; Tang, J. Self-Supervised Learning: Generative or Contrastive. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 857–876. [CrossRef]
18. Rogulina, A.; Zaytsev, A.; Ismailova, L.; Kovalev, D.; Katterbauer, K.; Marsalav, A. Generative adversarial networks for synthetic wellbore data: Expert perception vs mathematical metrics. *J. Pet. Sci. Eng.* **2022**, *211*, 110106. [CrossRef]
19. Liu, X.; Shao, G.; Liu, Y.; Liu, X.J.; Li, X.; Chen, Y. Deep classified autoencoder for lithofacies identification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5909914. [CrossRef]
20. Ashayeri, C.; Jha, B. Evaluation of transfer learning in data-driven methods in the assessment of unconventional resources. *J. Pet. Sci. Eng.* **2021**, *207*, 109178. [CrossRef]
21. Alolayan, O.S.; Raymond, S.J.; Montgomery, J.; Williams, J.R. Towards Better Shale Gas Production Forecasting Using Transfer Learning. *Upstream Oil Gas Technol.* **2020**, *5*, 100022. [CrossRef]
22. The New Zealand Petroleum & Minerals Online Exploration Database. Available online: https://data.nzpam.govt.nz/GOLD/system/mainframe.asp (accessed on 7 January 2023).
23. Rogulina, A.; Baramiia, N.; Kornilov, V.; Petrakov, S.; Zaytsev, A. Unsupervised construction of representations for oil wells via Transformers. *arXiv* **2021**, *2212*, arXiv.2212.14246. [CrossRef]
24. Parapuram, G.; Mokhtari, M.; Ben Hmida, J. An Artificially Intelligent Technique to Generate Synthetic Geomechanical Well Logs for the Bakken Formation. *Energies* **2018**, *11*, 680. [CrossRef]
25. Bai, S.; Kolter, J.Z.; Koltun, V. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. *arXiv* **2018**, *1803*. arXiv:1803.01271.
26. Chernikov, A.D.; Eremin, N.; Stolyarov, V.; Sboev, A.; Semenova-Chashchina, O.K.; Fitsner, L.K. Application of artificial intelligence methods for identifying and predicting complications in the construction of oil and gas wells: Problems and solutions. *Georesursy* **2020**, *22*, 87–96. [CrossRef]
27. Abu-Abed, F.; Ivanov, A. Application of computer modeling software for mining vehicle fleet telemetry monitoring. *Acta Montan. Slovaca* **2021**, *26*, 593–602. [CrossRef]
28. Chong, L.; Singh, H.; Creason, C.; Seol, Y.; Myshakin, E. Application of machine learning to characterize gas hydrate reservoirs in Mackenzie Delta (Canada) and on the Alaska north slope (USA). *Comput. Geosci. Comput. Geosci.* **2022**, *26*, 1151–1165. [CrossRef]
29. Aly, M.; Ibrahim, A.; Elkatatny, S.; Abdulraheem, A. Artificial intelligence models for real-time synthetic gamma-ray log generation using surface drilling data in Middle East Oil Field. *J. Appl. Geophys.* **2021**, *194*, 104462. [CrossRef]
30. Shan, L.; Liu, Y.; Tang, M.; Yang, M.; Xueyuan, B. CNN-BiLSTM Hybrid Neural Networks with Attention Mechanism for Well Log Prediction. *J. Pet. Sci. Eng.* **2021**, *205*, 108838. [CrossRef]
31. Lium, S.; Wang, Z. Formation Grain Size Profile Prediction Model Considering the Longitudinal Continuity of Reservoir Using Artificial Intelligence. In Proceedings of the SPE/IATMI Asia Pacific Oil & Gas Conference and Exhibition, virtually, 12–14 October 2021; Available online: https://www.spe.org/events/en/2021/conference/21apog/schedule-overview.html (accessed on 7 January 2023).
32. Abdulraheem, A. Generation of Synthetic Sonic Slowness Logs from Real-Time Drilling Sensors Using Artificial Neural Network. *J. Energy Resour. Technology. Trans. ASME* **2021**, *144*, 013201. [CrossRef]
33. Gamal, H.; Elkatatny, S.; Alsaihati, A.; Raheem, A. Intelligent Prediction for Rock Porosity While Drilling Complex Lithology in Real Time. *Comput. Intell. Neurosci.* **2021**, *2021*, 9960478. [CrossRef] [PubMed]

34. Wang, H.; Toksöz, M.; Fehler, M. Borehole Acoustic Logging—Theory and Methods. In *Petroleum Engineering*; Kalantari-Dahaghi, A., Shahkarami, A., Ferno, M., Eds.; Springer: New York, NY, USA, 2020; pp. 124–141. [CrossRef]
35. Xu, J.; Wei, W.; Bao, H.; Zhang, K.; Lan, H.; Yan, C.; Sun, W.; Wu, F. Study on shear strength characteristics of loess dam materials under saturated conditions. *Environ. Earth Sci.* **2020**, *79*, 1–25. [CrossRef]
36. Lin, H.; Zhang, X.; Cao, R.; Wen, Z. Improved nonlinear Burgers shear creep model based on the time-dependent shear strength for rock. *Environ. Earth Sci.* **2020**, *79*, 149. [CrossRef]
37. Aslam, A.; Ansari, M.S. Depth-Map Generation using Pixel Matching in Stereoscopic Pair of Images. *arXiv* **2019**, *1902*. arXiv:1902.03471.
38. Crawford, L.; O'Boyle, M. Circle C++ Shaders. In Proceedings of the 28th International Conference on Parallel Architectures and Compilation Techniques (PACT), Seattle, WA, USA, 23–26 September 2019.
39. Tello, J.J.L.; Burgos-Risco, A.; Reinaldos, G.R. Interaction According to Immersion in Virtual Environments: Graphic Development and PBRS in Environments with Real-Time Rendering and Virtual Reality. *Interacción* **2022**, *1*, 1–3.
40. Huo, Y.; Li, S.; Yuan, Y.; Chen, X.; Wang, R.; Zheng, W.; Lin, H.; Bao, H. Shader Transformer: Predicting Shader Quality via One-shot Embedding for Fast Simplification. In Proceedings of the Conference SIGGRAPH '22: Special Interest Group on Computer Graphics and Interactive Techniques Conference, ACMSIGGRAPH, Vancouver, BC, Canada, 8–11 August 2022.
41. Hasselgren, J.; Munkberg, J.; Lehtinen, J.; Aittala, M.; Laine, S. Appearance-Driven Automatic 3D Model Simplification. In Proceedings of the 32nd Eurographics symposium on rendering, EGSR, Saarbrucken, Germany, 29 June–2 July 2021.
42. Catanzaro, B. Video-to-video synthesis. *Adv. Neural Inf. Process. Syst.* **2018**, *12*, 1–24.
43. Farias, R.; Kallmann, M. GPU-Based Max Flow Maps in the Plane. In Proceedings of the Conference: Robotics: Science and Systems XIV, Carnegie Mellon University, Pittsburgh, PA, USA, 26–30 June 2018.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.