

Chapter 13

Real-Time Visualization and Geometry

Reconstruction of Large Oil and Gas Boreholes Based on Caliper Database

Fabiana R. Leta, Esteban Clua, Diego C. Barboza,
João Gabriel Felipe M. Gazolla, Mauro Biondi
and Maria S do Souza

Abstract The evaluation of technical and economical viability before starting the drilling process of a gas or oil reserve is very important and strategic. Among other attributes, the soil structure around the borehole must be analyzed in order to minimize the risks of a collapse. This stability analysis of a gas or oil reserve is a challenge for specialists in this area and a good result at this stage could bring a deep impact in reduction of drilling costs and security. A tool known as caliper [1] is inserted into the drilling spot to perform a series of measurements used to evaluate the well's viability. For each position along the borehole, information such as sensors' position, orientation, well's resistivity, and acoustic data are obtained and recorded. These data allow the user to find flaws in the soil, leaving to the geologist the decision whether the well is feasible or not, and help them to study possible actions to minimize its usage risk. Currently, the data obtained by the caliper are used for the visualization of individual sections of the well, projected in a bi-dimensional plane, considering a cylinder projection. However, an

F. R. Leta (✉) · M. Biondi · M. S do Souza
Mechanical Engineering Department, Computational and Dimensional Metrology
Laboratory, Universidade Federal Fluminense—UFF, 24210-240 Niterói, RJ, Brazil
e-mail: fabianaleta@id.uff.br

M. Biondi
e-mail: mbiondi@lmdc.uff.br

E. Clua · D. C. Barboza · J. G. F. M. Gazolla
Computer Science Institute, Universidade Federal Fluminense—UFF,
24210-240 Niterói, RJ, Brazil
e-mail: esteban@ic.uff.br

D. C. Barboza
e-mail: dbarboza@ic.uff.br

J. G. F. M. Gazolla
e-mail: gazolla@ic.uff.br

overview of the borehole's entire structure is necessary for a higher quality analysis. This work proposes a novel technique for a precise geometry reconstruction of the borehole from these data, allowing the geologist to visualize the borehole, making easier to find possible critical points and allowing an intuitive visualization of a large set of associated data. The three-dimensional geometry reconstruction is made from data collected from the caliper log, which includes the tool orientation and sensors' measures for each section. These measures are used as control points for the construction of smooth layers, through splines interpolation [2]. Finally, the sections are joined in sequence to form a polygonal mesh that represents a reliable vision of the borehole in three dimensions.

Keywords Well bore model · Oil well visualization · Caliper log · Three-dimensional visualization · Closed natural cubic splines

13.1 Introduction

Scientific real-time visualization techniques are important tools to understand what is almost impossible to be seen in numbers. When this visualization approaches are built into an interactive environment, a powerful workflow is available for the production and analysis pipeline. In the oil and gas industry, it is always very important to evaluate the technical and economical feasibility before starting the drilling process of a gas or oil reserve. The soil structure around the well must be known, in order to minimize the risks of collapse. The stability analysis of a gas or oil reserve is a central issue and a challenge for specialists. Good results at this stage could bring a deep impact in reduction of drilling costs. In the later years, wells have become longer and with higher inclination making it even more challenger and expensive.

According to Aadnoy [3], for years people have estimated the time loss associated with unexpected borehole stability problems to account for 10–15 % of the time required to drill a well. Since the rig time is the major cost factor in drilling operation, we understand that borehole stability problems are very costly for the industry. Borehole collapse is possibly the most costly single problem encountered during drilling of a well, and there is not a trivial solution for the problem.

This theory states that drilling generates changes in the stress field of the formation due to supporting material losses, inducing stresses that can result in more trouble [4]. If the stress is higher than the rock strength, rocks can cause the borehole collapse. So the main fundamental of the solution proposed in this work is to possibly prevent and reduce instability problems.

The better the planning of a well, the greater are the chances of success to be achieved. In this context, the word success means achieving the objectives of the project according to the safety standards of the company, time, and costs consistent with the ones of the market. However, to achieve this success, Geologists,

Reservoir Engineers and Production Engineers should not only care to the optimization of the production of the reservoir but should also be aware of the operational risks and expenses of the project.

When the well is drilled and a part of its formation is removed, there is instability of stresses. In that moment comes one of the most important stages of the project of a well which is ensuring the stability of it, and to achieve this goal it is necessary to evaluate factors like: rock resistance, temperature variations, well trajectory, and geometry.

This work proposes a novel framework for a 3D geometric reconstruction and visualization of the bare holes and its components. Data structures and processes adopted allow a real-time interaction, using dedicated but not expensive graphic hardwares. New interfaces and data manipulation are proposed, increasing the workflow process. In order to maintain this real-time requirement, our system builds the geometry using a thread-based reconstruction model and level of detail algorithms for maintaining huge datasets in the memory.

13.2 Related Works

In [1] it presented a borehole visualization approach, more focused on stress and breakout analysis. Barton and Zoback [5] and Peska and Zoback [6] presents important factors to be considered in order to model the requirements associated to a detailed data processing and analysis of this kind of data, that will be considered on the proposed system.

In [7] the authors present a technique for modeling a 2D approach, so important requirements of these structures are maintained and can be analyzed, such as perforation fluid weight, rock strength, and hole geometry. Figure 13.1 shows the data obtained by a four arms caliper log. The 2D developed system shows an approximation of the borehole shape in a specific height (Fig. 13.2). Each colored profile is related to a caliper arm and is used for the 3D reconstruction.

Despite allowing the user to obtain various types of information, this tool does not present a global spatial view of the three-dimensional geometry of the borehole, even the most experienced specialists can experience difficulties to imagine its geometry, being a restriction, but a key point to the proposed three-dimensional tool.

Debogurski [8] presents a procedural terrain generation using the recent Marching Cubes Histogram Pyramids (also known as HPmarcher) implementation. Perlin Noise function is used to procedurally create the terrain. This is an important approach related to our proposal, since it is a GPU-based approach for the modeling optimization, which allows a huge number of polygons to be handling in a Real-Time System.

This work merges characteristics from most of these works, enhancing the visualization process, since important data may be mapped into a huge and massively geometry borehole reconstruction.

Fig. 13.1 Profiles obtained by a 4-arm caliper log

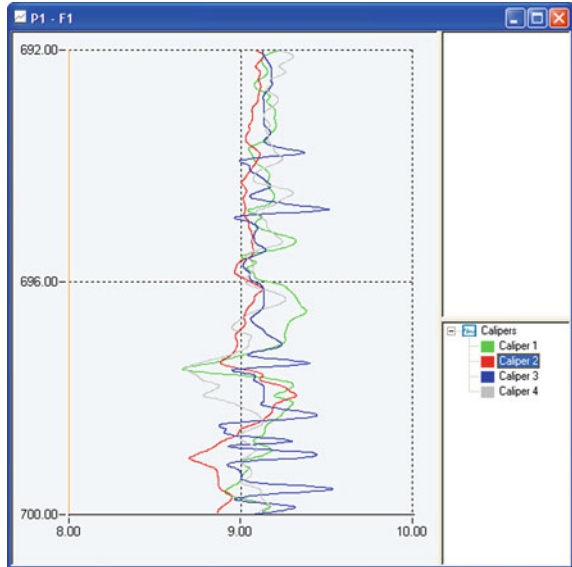
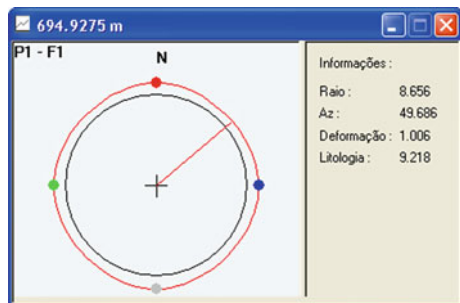


Fig. 13.2 A screen of the 2D caliper analysis system



13.3 Methodology

Our methodology is divided into six steps (Fig. 13.3). The process begins describing how the data is acquired by the Caliper tool. After this, the data is converted into a specific data structure. The third step consists on the calculations necessary for the geometric consistency, and on the fourth step some optimization are done, which means that we try to send to the next step only the necessary vertices. To finish the process the user can make the analysis with the tool information and finally obtain useful knowledge. A more detailed explanation is described in the following subsections. This workflow is related to the usage process of the proposed system.



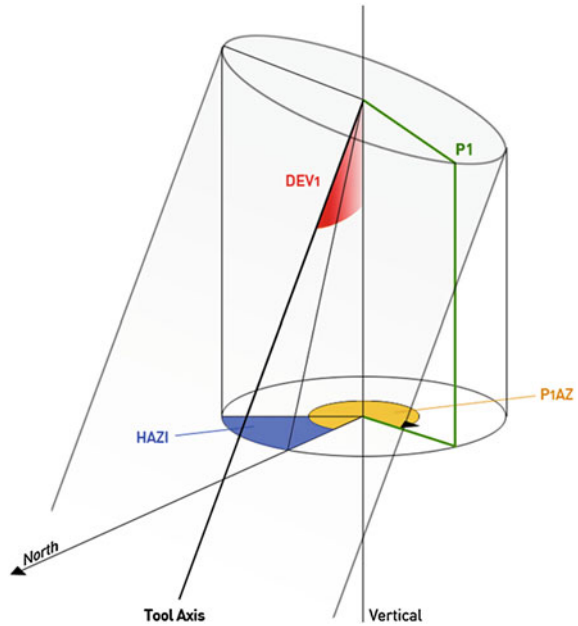
Fig. 13.3 The framework of the 3D Caliper tool (GeoPoço 3D)

13.3.1 Data Acquisition

A borehole failure can be detected by different measurement instruments, like the following caliper logs: Borehole Geometry Tool; Resistivity logs; Ultrasonic logs. The caliper tool is inserted into drilling spot to perform a series of measurements used to evaluate the borehole’s feasibility [1]. It measures the borehole diameter variation by depth, i.e., its shape and can also give other important information, as well hole irregularities or damages. The caliper log can have two, four, or more extendable arms arranged around the tool. Figure 13.4 shows a simple schema of the tools and obtained data. Among all these measures PIAZ represents the alignment of the first arm with the magnetic north in degrees, DEVI represents the tool’s inclination and combined with HAZI gives information about the direction. All this informations are strictly necessary to reconstruct the borehole geometry, and these variations on every section will allow the borehole to make curves, creating the sensation of a real borehole and not just a simple geometric form like a cylinder.

For each position along the borehole, information such as sensors’ position and orientation, rock’s resistivity and acoustic data are obtained. These data allow the user to find flaws in the soil, leaving to the geologist the decision whether the well

Fig. 13.4 A 3D sketch of the Caliper tool orientation



is feasible or not, and possible actions to minimize its usage risk. Figure 13.5 shows an example of a caliper sample file, as it is provided by the tool.

Table 13.1 shows an example of a single line data obtained from a file of the Caliper Tool.

13.3.2 Preprocessing

The borehole data is stored in a file and then read and loaded into the main memory. The reading process takes a while, due the large file size and the transformations that are applied to transform it into tridimensional data. For example, a 4 Km borehole is stored in a file of about 180 MB of raw data that must still be read and converted into the three-dimensional borehole geometry.

From a usability point of view, it is not convenient that the system stays locked until all the geometry is ready for rendering. Hence, we propose in our process to show some visual information in the screen as soon as possible. This is done by combining two working threads, one responsible for reading and transforming the data and the other that takes this data and builds the mesh geometry.

This threading part is managed with C++ Boost's thread module [9], a portable library which allows execution of multiple threads with shared data and provides functions for thread management and synchronization.

Arquivo	Editar	Formatar	Ejibir	Ayudas							
0.0375	9.45771	8.96247	9.23667	8.23771	9.28362	9.53060	0.63458	0.00837	359.97920		
0.0400	9.45022	8.96564	9.22697	8.24537	9.28225	9.53265	0.66412	0.00913	359.97555		
0.0425	9.44753	8.96611	9.21854	8.24730	9.28166	9.54059	0.72516	0.00973	359.97555		
0.0450	9.44545	8.97111	9.21681	8.24855	9.27179	9.54910	0.80914	0.01045	359.97666		
0.0475	9.43775	8.97478	9.21162	8.25772	9.26549	9.55789	0.86895	0.01122	359.97781		
0.0500	9.42866	8.98322	9.20406	8.26461	9.25636	9.56357	0.88213	0.01160	359.97807		
0.0525	9.41933	8.98442	9.19563	8.26741	9.25353	9.56800	0.97549	0.01201	359.97838		
0.0550	9.41238	8.98759	9.19117	8.27182	9.24631	9.57163	1.03794	0.01209	359.97990		
0.0575	9.40249	8.98812	9.18807	8.27572	9.24145	9.57178	1.10218	0.01283	359.98237		
0.0600	9.39424	9.99346	9.18736	8.28046	9.23429	9.57829	1.10242	0.01336	359.98610		
0.0625	9.38902	8.99552	9.17764	8.28931	9.23418	9.57871	1.13501	0.01454	359.98663		
0.0650	9.38403	9.00023	9.17504	8.29188	9.23190	9.58644	1.15543	0.01456	359.99151		
0.0675	9.38296	9.00791	9.16893	8.29391	9.22423	9.59101	1.18033	0.01508	359.99640		
0.0700	9.37659	9.01049	9.15924	8.29474	9.21944	9.59320	1.23034	0.01543	359.99916		
0.0725	9.37330	9.02030	9.15766	8.29807	9.20952	9.59284	1.25364	0.01658	359.99186		
0.0750	9.36869	9.02314	9.15191	8.30805	9.20339	9.59599	1.28062	0.01675	359.99163		
0.0775	9.36017	9.02919	9.14509	8.31126	9.19679	9.60365	1.32765	0.01676	359.98919		
0.0800	9.35081	9.02979	9.14402	8.32056	9.18738	9.61215	1.41941	0.01681	359.98709		
0.0825	9.34255	9.03518	9.13779	8.32934	9.18713	9.61276	1.48434	0.01692	359.98541		
0.0850	9.33516	9.04228	9.13111	8.33090	9.18539	9.61736	1.52253	0.01696	359.98064		
0.0875	9.32659	9.05172	9.12647	8.34082	9.18186	9.62048	1.59575	0.01755	359.97912		
0.0900	9.32326	9.05472	9.11906	8.34607	9.17220	9.62474	1.61193	0.01791	359.97776		

Fig. 13.5 Sample data obtained from Caliper tool

Table 13.1 A sample of some caliper fundamental measures data

Data	Sample value	Definition
Depth measure	0.07 m	Measure's depth
Arm 01	9.37619 in	Length of Arm 01
Arm 02	9.01049 in	Length of Arm 02
Arm 03	9.15924 in	Length of Arm 03
Arm 04	8.29704 in	Length of Arm 04
Arm 05	9.21574 in	Length of Arm 05
Arm 06	9.59120 in	Length of Arm 06
DEVI	1.23091°	Deviation from the surface
HAZI	0.01578°	Projection of the borehole axis in the horizontal plane
PIAZ	359.99640°	Projection of pad 1 in the horizontal plane

13.3.3 Math

At this stage we implement one of the most important parts of the visualization tool, which is the transformation process of the 2D data obtained by the Caliper tool into 3D data. This effort is divided in 6 steps as can be see on the following subtopics.

13.3.3.1 Transformation of the Caliper Arms Length into a 2D Data

The Caliper data file stores the arms' length as a single value that represents the distance from the tool center and the arm pad. This data must be transformed into 2D coordinates (and later 3D) in order to build the borehole geometry.

For the sake of simplicity, the tool axes are located in the center of the Cartesian coordinate system and the direction of the first arm coincides with the positive X-axis, as shown in Fig. 13.6. The arms can be described in polar coordinates, so each coordinate for the other arms can be obtained from equations in Table 13.2.

Fig. 13.6 Top View of the Caliper tool inside a well bore

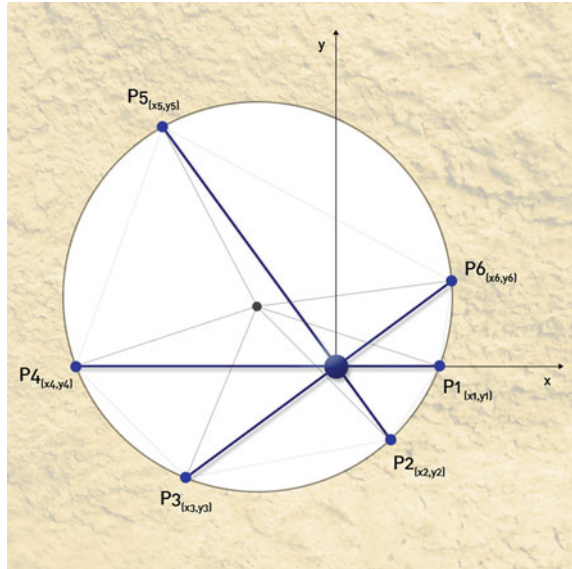


Table 13.2 How to calculate the X and Y coordinates

Data	X-Value formula	Y-Value formula
Arm C1	$P1 \cos 0^\circ$	$P1 \sin 0^\circ$
Arm C2	$P2 \cos 300^\circ$	$P2 \sin 300^\circ$
Arm C3	$P3 \cos 240^\circ$	$P3 \sin 240^\circ$
Arm C4	$P4 \cos 180^\circ$	$P4 \sin 180^\circ$
Arm C5	$P5 \cos 120^\circ$	$P5 \sin 120^\circ$
Arm C6	$P6 \cos 60^\circ$	$P6 \sin 60^\circ$

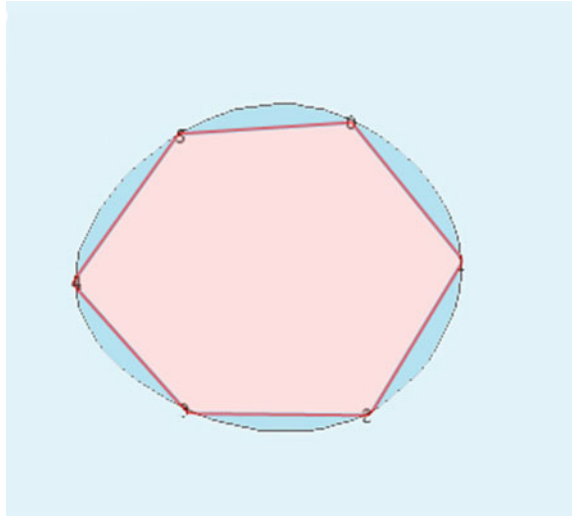
13.3.3.2 Interpolation of the Data with Splines to Obtain a Section with the Desirable Resolution

Given six 2D control points (obtained from the Caliper’s arm data) instead of simple connect the points (red lines in Fig. 13.7) a smooth curve is created (black lines in the same figure) to interpolate them using a closed natural cubic splines algorithm. An interesting demonstration can be found in [10].

13.3.3.3 Inclusion of the Rotation in P1AZ

The interpolated points are rotated around the section center in the Y-axis. The rotation angle is defined by the P1AZ value read by the sonde. This aligns the first arm with the earth’s magnetic north.

Fig. 13.7 Drawn in *red* is the original data acquired from the six-arms caliper control points. In *blue* is an approximation representation created with an interpolation of a closed natural cubic splines



13.3.3.4 Inclusion of the Rotation in DEVI

DEVI value defines the sonde deviation from the surface. Combined with HAZI, it's used to determine the tool's inclination and direction.

13.3.3.5 Inclusion of the Rotation in HAZI

HAZI tells the borehole angle with the earth's magnetic north. This value, combined with DEVI, allows finding out tool's inclination and direction.

13.4 Optimization

Optimizations are necessary in order to keep the visualization system running in real-time at acceptable frame rates. This section describes some techniques used for this purpose for this work.

13.4.1 Level of Detail

Due to the common large size of the boreholes, it is not practical to render its entire geometry in every frame. This could easily lead to GPU memory overflow and even if memory is not an issue, processing time should still be taken into

account. But as in many visualization systems, it is not mandatory to render the entire geometry with full detail and some simplification for distant objects is accepted. So, the level of detail technique [11] can be applied to increase the system performance.

Keeping a low polygon count is the key point to achieve high frame rates, so before sending geometry to the render stage in the graphics pipeline, the model is processed and only parts near to the camera are built in high level of detail. Distant parts are built in lower level of detail, but it is well-balanced enough that major features of the borehole (such as its direction and curvature) are not lost and can still be visualized from distance, while that small details from distant sections are ignored. As the camera moves through the borehole, the level of detail is dynamically updated, so the user always visualizes the high-resolution mesh when it's near enough, ensuring that the user will never see problems on the geometry due to the low resolution.

In Fig. 13.8, we present a fake borehole built for demonstration only. The first frame (a) shows a borehole represented with full resolution. Here, all the borehole data is used for rendering. The next frame (b) shows how the borehole should be represented with a lower resolution. The sections are not interpolated, so each one is in fact a hexagon, and only a small fraction of the original data is used to build the geometry, this maintains the overall shape of the borehole, but small variations are lost. The last frame (c) mixes these two visualizations and shows a high-resolution borehole for parts that are near to the camera. As the camera moves away, lower detail level is required, while the viewer still has the impression that the geometry is unchanged. This happens because, different from this example, in the actual system we only apply the level of detail for parts that are really far away from the camera and smaller geometry changes may go unnoticed.

13.4.2 Concurrent Mesh Manipulation and Parallelism

As described in Sect. 13.3.2—Preprocessing, the borehole data is loaded without locking the visualization window. Whenever a thread finishes loading a part of the borehole, it is immediately sent to the rendering without major performance costs. The process for changing the level of detail after the model is fully loaded uses a similar technique.

For memory space reasons, only one mesh for a given part of the borehole is stored in the GPU memory at a time. Whenever the level of details changes, a new mesh with a different level of detail is again built by the CPU and sent to the GPU. The older mesh is still rendered until the new mesh is ready, so a gap is never formed while the meshes are changed. The building process of a new mesh is done by a concurrent thread, so no slow down is noticed. When it's ready for rendering, the older mesh is removed from memory and the new one becomes visible.

The borehole is divided in a variable number of meshes depending on the borehole size. Anytime the level of detail should be changed, i.e., when the camera

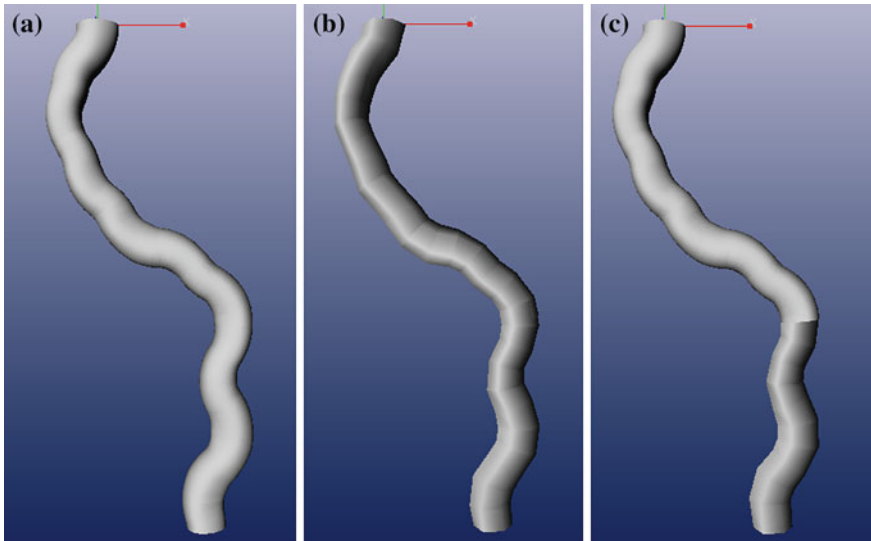


Fig. 13.8 A fake borehole three-dimensional mesh with different levels of detail

moves, only the closer meshes are updated, not the borehole model. This is a different approach from the commonly used technique, where the model is fully changed for each level of detail.

13.5 Rendering

Once the data is fully loaded into the main memory and processed, it's possible to reconstruct a three-dimensional representation of the borehole. As previously described, some transformations are applied to caliper arm's points in order to transform them from a bi-dimensional plane to a three-dimensional space. The next step is to combine each section and create a 3D model of the borehole that can be used for visualization and analysis. If this combination had not been done, the model would simply look like as a simple stack of slices instead of a reliable three-dimensional representation of the borehole.

The representation of the borehole as a series of connected sections makes the usage of triangle strips [12] a natural choice for building the geometry. The connection between two adjacent sections is built with a single triangle strip, instead of a collection of independent triangles. This results in lower memory usage by the GPU and better performance. Figure 13.9 shows how we start from points obtained from the spline interpolation (a) and build the borehole geometry (b and c).

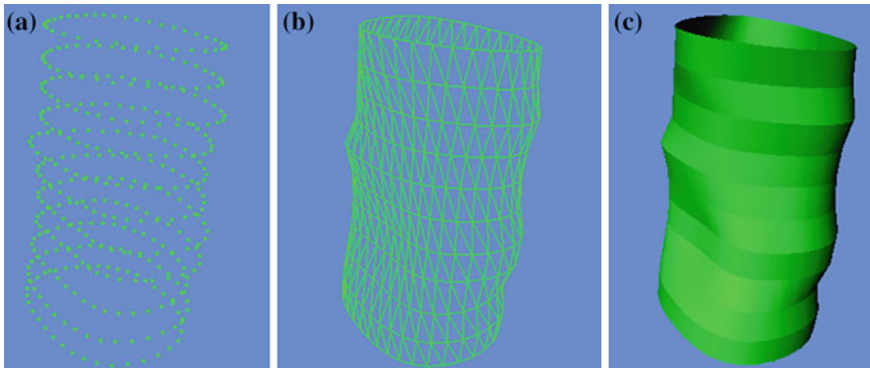


Fig. 13.9 Borehole reconstruction using triangle strips

Figure 13.10 shows one simple example of this process. Given a pair of sections, let's call A the upper section indicated by black dots and B the white ones. From left to right (i.e., from the first arm or control point to the last one), we add to the triangle strip the first vertex from A and B, in this order (points 1 and 7 in the figure). Then, we move to the next pair and add the new points from A and B (2 and 8), and so on until all vertex from both sections are added. The first two points are added again in the end, so the strip closes the section. Each new point added after the first pair results in the creation of a new triangle (1-7-2 and 7-2-8, for example). When the process is over, a small mesh connecting two sections is created. Applying these steps to all sections it results in a 3D representation of the borehole. Listing 13.1 describes a pseudo-code for this process.

13.6 Analysis and Information

Study of borehole stability requires not only a deep knowledge of problems' sources, but it is accurate identification and modeling. When a borehole is drilled in a nonisotropic prestressed rock, we can note a failure, which is a result from the reorientation of the stress field and the stress concentration around the borehole wall. The stability of an oil reservoir is a challenge for specialists of the petroleum industry; therefore, a correct analysis of this question can reduce the perforation cost significantly. It can be considering that a considerable time expense to perforate a hole is related to the analysis of its stability, which means an annually high cost.

The caliper tool is used in this purpose for gathering data such as resistivity and acoustic measures of the borehole wall, as well as diameter for each section. These data can be used to identify problematic points where tension could lead to the borehole's collapse.

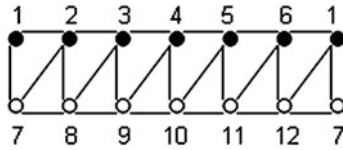


Fig. 13.10 Sample triangle strip connecting two sections

```

for i = 0; i < sections.size - 1; i++
    A = sections[i]
    B = sections[i+1]
    for j = 0; j < A.vertexCount; j++
        addVertex (A[j])
        addVertex (B[j])

```

Listing 13.1 Pseudocode for creating the triangle strips

The three-dimensional model of the borehole allows the user to visualize the possible problematic points and have access to collected data for the sections. For each section, the user is able to request many calculations, such as sections diameters relation, section angle, and cut type classification. This combination of data and visual information allows an easier way to identify whether the borehole is stable and which actions should be taken.

13.7 The Implementation of the Three-Dimensional Visualization System

The visualization system, named “GeoPoço 3D”, was written in C++ with an OpenGL-based graphics framework. An initial prototype was made using Irrlicht [13], an open-source 3D engine that provides a certain level of abstraction of hardware resources by its API and great performance. A new version is under development using Open Inventor [14] instead of Irrlicht.

The tests were carried out on a desktop with the Windows XP Professional SP2 operating system was installed, with a Core 2 Duo processor E7500 2.93 Ghz and 2 GB of RAM. The graphic card used it was an NVidia GeForce 9600 GT with 512 MB of RAM and 64 stream processors. The former prototype.

13.8 Results

A key point in this visualization system is that very large boreholes can be analyzed without visible performance loss. Should the borehole be rendered entirely in high-resolution, the GPU would quickly run out of video memory and its processing power is very likely to not be able to deal with this data flow.

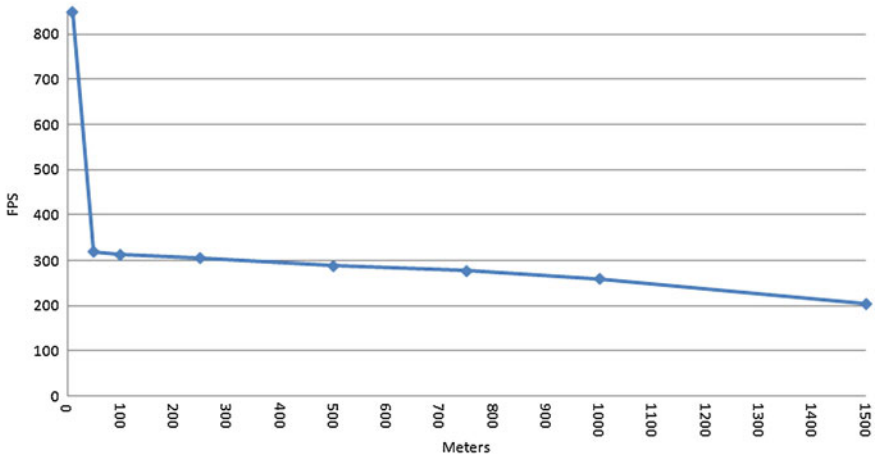


Fig. 13.11 Comparison of performance of the FPS rate versus the growth of the well length

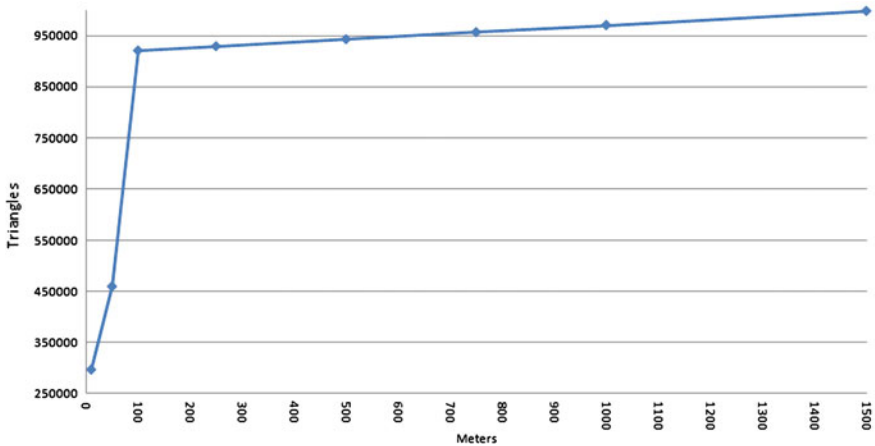


Fig. 13.12 Comparison of the growth of the well length versus the amount of triangles

The level of detail technique allows the borehole to be fully visualized, but only the nearest parts are rendered with full detail level. Distant parts are built with very simple geometry that still looks like the original borehole, but lacks detail.

This technique allows the visualization of very large boreholes without major performance issues. As the borehole gets bigger, only a small fraction of its structure is added to the scene graph, so the frame rate tends to be stable. Figure 13.11 demonstrates how the system behaves as the borehole size increases. A 10 meter borehole runs at more than 800 frames per second. In this case, no

level of detail was applied. The frame rate drops more than 2.5 times for a borehole 10 times larger (100 m), with level of detail already applied. One should expect that when we double the borehole size, the frame rate should drop 5 times and so on. But with the level of detail applied, the triangle count does not grow much fast and the frame rate becomes stable.

The system is scalable for visualization of very large boreholes. Since it's estimated that specialists will analyze sections of 500 m or less at a time, the current performance is more than enough for a completely smooth analysis.

Figure 13.12 shows how the number of triangles does not increase as fast as the borehole length, consequence of the level of detail used to render the scene, what means that only the necessary triangles are sent to GPU memory. This allows larger sections and higher resolutions without significant rate loss.

13.9 Conclusions

In this chapter, the authors presented a framework for real-time visualization and geometry reconstruction of large oil and gas boreholes.

The visualization of a borehole is a relevant issue for the oil and gas industry as well for the productivity of the field. If you know where the problem can occur or where it is better to drill you can increase your production.

The tool proved to be able to scale up for really huge wells since in all the instances clearly show that even on the wellbores with thousands of meters the system provides an interactive rate of at least 200 frames per second for all the well-visualization tasks like moving, zooming, rotation, and translation.

The tool can help you spend less time analyzing wellbore stability issues, helping decisions, reducing project risks, drilling costs, and increasing the project profit.

However, there are still many features to be implemented as you can see on the future work section.

13.10 Future Works

At present, we are only using a simple interface with a mouse and a 3D view. The next aim is to provide a better visualization of the well using a touch surface table, controllers, and 3D glasses. As future work, we also intend to target it at further development, increasing interaction and automated diagnosis of failures on the well.

Acknowledgments The authors would like to acknowledge Petrobras Oil for the financial support.

References

1. Jarosinski, M., Zoback, M.D.: Comparison of six-arm caliper and borehole televiewer data for detection of stress induced wellbore breakouts: application to six wheels. Polish Carpathians, pp. F8-1–F8-23+12 figures (1998)
2. Reinsch, C.: Smoothing by spline functions. *Numer. Math.* **10**, 177–183 (1999). In: Proceedings of the XII SIBGRAPI pp. 101–104 (1967)
3. Aadnoy, B.S., Balkema, A.A.: Modern well design. Rotterdam. ISBN 90 54106336 (1996)
4. Jiménez, J.C., Lara, L.V. et al.: Geomechanical wellbore stability modelling of exploratory wells—study case at middle magdalena basin. *C.T.F Cienc. Technol. Futuro* **3**(3), 85–102 (2007)
5. Barton, C., Zoback, M.D.: Stress perturbations associated with active faults penetrated by boreholes: possible evidence for near-complete stress drop and a new technique for stress magnitude measurements. *J. Geophys. Res.* **99**, 9,373–9,390 (1994)
6. Peska, P., Zoback, M.D.: Compressive and tensile failure of inclined well-bores and direct determination of in situ stress and rock strength. *J. Geophys. Res.* **100**, 12,791–12,811 (1995)
7. Leta, F.R., Souza, M., Clua, E., Biondi, M., Pacheco, T.: Computational system to help the stress analysis around boreholes in petroleum industry. In: Proceedings of the ECCOMAS 2008, Venice (2008)
8. Dembogurski, B., Clua, E., Leta, F.R., Procedural terrain generation at GPU level with marching cubes. In: VII Brazilian Symposium of Games and Digital Entertainment—Computing Track. Proceedings of the VII Brazilian Symposium of Games and Digital Entertainment—Short Papers. Porto Alegre: Sociedade Brasileira da Computação pp. 37–40 (2008)
9. Williams, A.: Boost C++ Libraries Chapter 22—Thread http://www.boost.org/doc/libs/1_43_0/doc/html/thread.html (2008)
10. Lambert, T.: Closed natural cubic splines. www.cse.unsw.edu.au/lambert/splines/natcubicclosed.html
11. Luebke, D., Reddy, M., et al.: Level of detail for 3D graphics. Morgan Kaufmann, San Francisco 432 p (2003)
12. Schroeder, W.J.: Modeling of surfaces employing polygon strips. <http://patft1.uspto.gov/netacgi/nphParser?Sect1=PTO1&Sect2=HITOFF&d=PALL&p=1&u=%252Fnetacgi%252FPTO%252Fsrchnum.htm&r=1&f=G&l=50&s1=5,561,749.PN.&OS=PN/5,561,749> (1996)
13. Gebhardt, N.: Irrlicht Engine. <http://irrlicht.sourceforge.net/> (2010)
14. Open Inventor http://www.vsg3d.com/vsg_prod_openinventor.php (2010)