

IMMERSIVE VISUALIZATION OF GEOPHYSICAL DATA

Philippe A. Cerfontaine, Anne-Sophie Mreyen, Hans-Balder Havenith

University of Liège
Geology Department
Allée du six Août 14 - B 4000 Liège

ABSTRACT

In this application paper we'll explain the work flow we use to create immersive visualizations and spatial interaction for geophysical data with a head mounted device (HMD). The data that we analyze consists of two dimensional geographical map data and raw geophysical measurements with devices like seismometers, Electrical Resistivity Tomography (ERT) and seismic tomography profiles as well as other geophysical and geoscientific data. We show the tool chain that we use while explaining the choices that we made along the way. The technical description will be followed by a brief assessment of the added benefit of rendering our data in virtual reality (VR). After the technical description we conclude this paper with some outlook on the (likely) future use of VR in geosciences.

Index Terms— virtual reality, visualization, geology, geophysics, geography.

1. INTRODUCTION AND MOTIVATION

In modern geophysics the three dimensional (3D) analysis of processed measurement data is very important. Being able to integrate data sets from different sources into a single spatial visualization thus becomes primordial to finding, understanding and analyzing the geology of the site where the measurements were made. The multitude, variety and density of the resulting integrated data often overlap when they are projected to a flat screen. The need for increased depth perception drove us towards the recently rediscovered VR technology. Combined with the limited and often cumbersome interaction capabilities provided by simple desktop environments with a mouse and keyboard we had several reasons to test our data in VR. Our approach is supported by recent technological advancements, both in the field of hardware and software development. Further this technology has become affordable through widespread use in the entertainment industry, so we are able to use it for the visualization of our geological models.

2. REFERENCES AND RELATED WORK

Visualization of geophysical data traditionally involves a lot of 2D representations: Maps, profiles, seismograms and other graphs. Just as most types of geoscientific exploration, geophysics requires 3D to investigate the relationship between all those 2D data. To stay on topic we won't go into details and specifics of 2D visualization in geophysics but we'd like to reference some of the major developments for 3D representation in geology. GOCAD [9] is one example of a once academic development in this field that has long since then become a commercial product. Due to the necessity for geological data visualization in geothermal, oil, gas or mining industries (among others) it is no wonder that commercial software solutions like CoViz [2] have long since taken the lead in that domain and are still being developed. Even though such software mostly includes modules that allow representing complex 3D data sets, including time-dependent information that can also be jointly interpreted, these solutions rarely offer support for VR technology. In most cases this was probably because of the complexity and high price of VR equipment. This publication [6] shows that the recent developments in VR technology (hard- but also software) are changing that paradigm. Through the use of the Unreal Engine from EPIC [3] that already had its first VR application in [7]. Two good examples of immersive visualization applied to surface and fault data in geo-sciences using VR would be [8] and [11] respectively.

3. 3D VISUALIZATION OF GEOPHYSICAL DATA

For us the first step to enable the spatial analysis of geophysical data is to integrate the available data. Combining data from various measurement methods performed in the field into a single 3D model with real world coordinates is necessary to get useful information on the subsurface structure of studied sites. The measurements cover most applied geophysical techniques, from electrical profiling to ambient noise recordings (including seismics, electro-magnetic measurements, among others). Once all the pieces have been accurately assembled, visualization matters like uniform color mapping across several profiles or transparencies have to be

addressed. After these problems have been solved the second step will be to use the created model in VR.

This simplistic two stage approach is more important for us than it may seem at first sight. Because it lets us split the workload more evenly. Ideally stage one could be accomplished by an expert in geology and geophysics modeling while in stage two the results would be further post-processed by someone whose expertise is in the field of visualization and virtual reality.

3.1. Geographic raster data

Typically, the first available data for a new site are aerial imagery (ortho images), a digital elevation model (DEM) and in some cases a geological or topographic map. With geographic information systems (GIS) that rely on two dimensional information representation, we use the DEM to extract the depth information of the terrain to propagate it to otherwise flat data.

The DEM is a georeferenced raster image where each pixel contains an elevation value. It is essential for building our 3D model as it will be used as basic input representing topography, and as support for many types of textures including remote imagery.

To generate a 3D surface from the DEM we convert the raster image to point data with XYZ information. The resulting point cloud can either be read in Paraview [1] by exporting the attribute table from the GIS or directly importing the shape file to Blender [5] using the appropriate BlenderGIS plug-in. Both tools provide algorithms to compute a triangulated surface geometry. Because we want to render our model in VR (which is more demanding for the graphics hardware) the surface geometry could be decimated in flat areas if necessary.

The main purpose of the surface model is to give the viewer a sense of orientation. A good way to even further the sense of orientation is to apply aerial imagery or topographic map data as a texture to the obtained surface geometry. The Blender GIS plug in has functionality to accurately map georeferenced raster images to geometries in the same coordinate system (see Figure 1). One thing to keep in mind with geographic raster data is that it often contains huge amounts of pixels and might not have dimensions that exactly match a power of two, some early adjustments using tools in the GIS will generally save a lot of trouble down the line when dealing with large data sets.

As surface data might hinder viewing subsurface information, it might be required to make them partly transparent. Which is certainly a good idea while you don't have to manage a lot of transparency layers in the scene. But as we will see in the next section adding multiple intersecting profile data can rapidly make it tedious to manage transparency effects while avoiding to confuse the viewer. Another technique is to draw the surface points only. The problem is that simple points don't provide depth information to the viewer. Rep-



Fig. 1. The textured surface model helps viewers to get a sense of orientation in the scene.

resenting the surface in wire frame mode might provide the viewer with more depth cues but it also clutters the display a lot, depending on the resolution of the surface model. This is especially true for surfaces that have not been subject to mesh decimation but at least their regular structure improves depth perception, while decimated meshes are generally too chaotic to be visualized in wire frame, especially without re-triangulation.

Another way to enable the viewer to see through the surface model while preserving graphics compute power is to use contour lines. This widespread method for representing height on maps actually works quite well in 3D. They can easily be computed based on the initial DEM using the GIS. Due to their familiarity and the implicit depth cue of each line representing a certain height with constant vertical spacing between lines, in our experience, they give the viewer a good indication of the terrain while minimizing display clutter (see Figure 2). The only downside of contour lines is that they don't have surface normals and thus no illumination, like we already explained for points.

3.2. Geographic vector data

Now that we have seen how to integrate imagery and raster data from the GIS we still have to transfer measurement points and profile lines in short vector data to the 3D scene. Points can be imported in exactly the same way that we already explained for the DEM point cloud in the previous section. To represent them in the scene we use simple geometries like spheres or cylinders as glyphs with different colors or color scales to symbolize different values. The glyph's size can also be used as a means to convey additional information to the viewer. Glyphs that are located on the surface model should be big enough to reach through it. This way they can serve as

landmarks to help the viewer get his bearings from above and below ground level.

Line data can be represented with tubes instead of simple lines since they can be illuminated for better depth perception. Lines that we want to represent on the surface should be subdivide to prevent the tubes from completely cutting through the geometry. N.B.: This technique might be applied to the aforementioned contour lines but as for glyphs it raises the polygon count in the scene.

Polygon data, especially large area polygons, have to undergo a special procedure when they have to be represented in 3D. To avoid the generated 3D polygons from cutting through the model's surface we have to insert additional points inside each polygon. In order to match the DEM surface for each polygon we want to render as geometry we use the clipping functionality from the GIS to create point subsets from the DEM point cloud that we then have to triangulate using the Delaunay algorithm. The procedure is very similar to the one we used to generate the surface model only in smaller chunks. The result might become blocky because it is bound to the original DEM grid's resolution. Considering that we have to repeat the whole procedure for every polygon it might be more efficient to use the GIS to export the complete map as a georeferenced raster image and drape it on the surface geometry as a texture.

3.3. Profile data

Generally speaking profile data is a set of two-dimensional graphs that represent seismic velocities or electrical resistivity. Those were obtained through 2D inversion of measurement data with the adequate software. These graphs may or may not intersect one another (see Figure 2). Once the profile images are ready to be exported in high resolution they have to be placed in the 3D scene as accurately as possible. We're going to import these profiles as images on planes with transparent background. Next we'll need at least two reference points on the surface geometry and the profile that we have to match together. Point data imported from the GIS as described in the previous subsection might be a good choice for that purpose. First the image has to be scaled accordingly before placing it with respect to the surface geometry. This can become a tedious and repetitive task when sitting at the desktop.

4. IMMERSIVE VISUALIZATION AND INTERACTION WITH GEOPHYSICAL DATA

4.1. Visualizing geophysics in VR

From the hardware perspective we are currently working with the Vive system from HTC [10] mostly because of the precise and responsive tracking system that allows us to track a room of 3.5 by 4.5 meters (in our case) with bi-manual 6dof interaction. On the software side we use the Unreal Engine 4 from

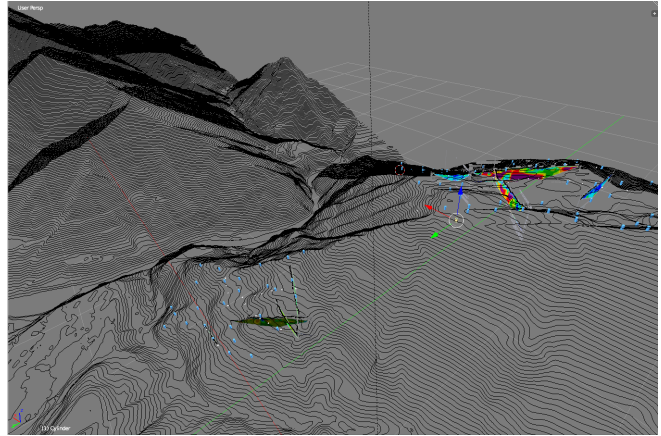


Fig. 2. Intersecting profile data visualized with contour lines as surface representation.

EPIC [3], because it is powerful and has out of the box support for our current VR hardware while the provided functionality also has a layer of abstraction from it.

To setup the scene we built in the 3D modeler it is better to think about the way we want to present the data to the final viewer instead of using a monolithic copy. Considering the frequently overlapping data from different geological measurements or features it is better to partition the scenes objects so that the different parts can be enabled and disabled individually in the final scene. For instance one way to deal with this is to export all the seismic profiles as one object and to create another object with all electric profiles thus bringing more structure to the final scene. With the scene graph's hierarchy being organized around the surface model as backbone we usually attach all other parts of the model to the ground level. Light sources used to emphasize certain features in the scene should also be anchored in a way that makes sure that they move according to what they're supposed to highlight (see Figure 3).

Before we go over to the interaction part of this section we still have settings and adjustments to make to the final scene's materials. Since transparencies don't automatically carry over from the 3D modeling tool. The settings have to be adapted material-wise to achieve the desired effect. Translucency of several overlapping profiles can rapidly become confusing for the viewer with additive transparency settings. Another way to solve this problem is to mask out the profile background using the image's alpha channel and reserve translucency for a see through effect on the surface model thus minimizing overlapping transparencies in the process.

4.2. Interacting with geophysical data in VR

Since version 4.12 the Unreal Engine comes with a built in VR editor mode that lets you use motion controllers to build and tweak the 3D scene with 6dof interaction while wearing

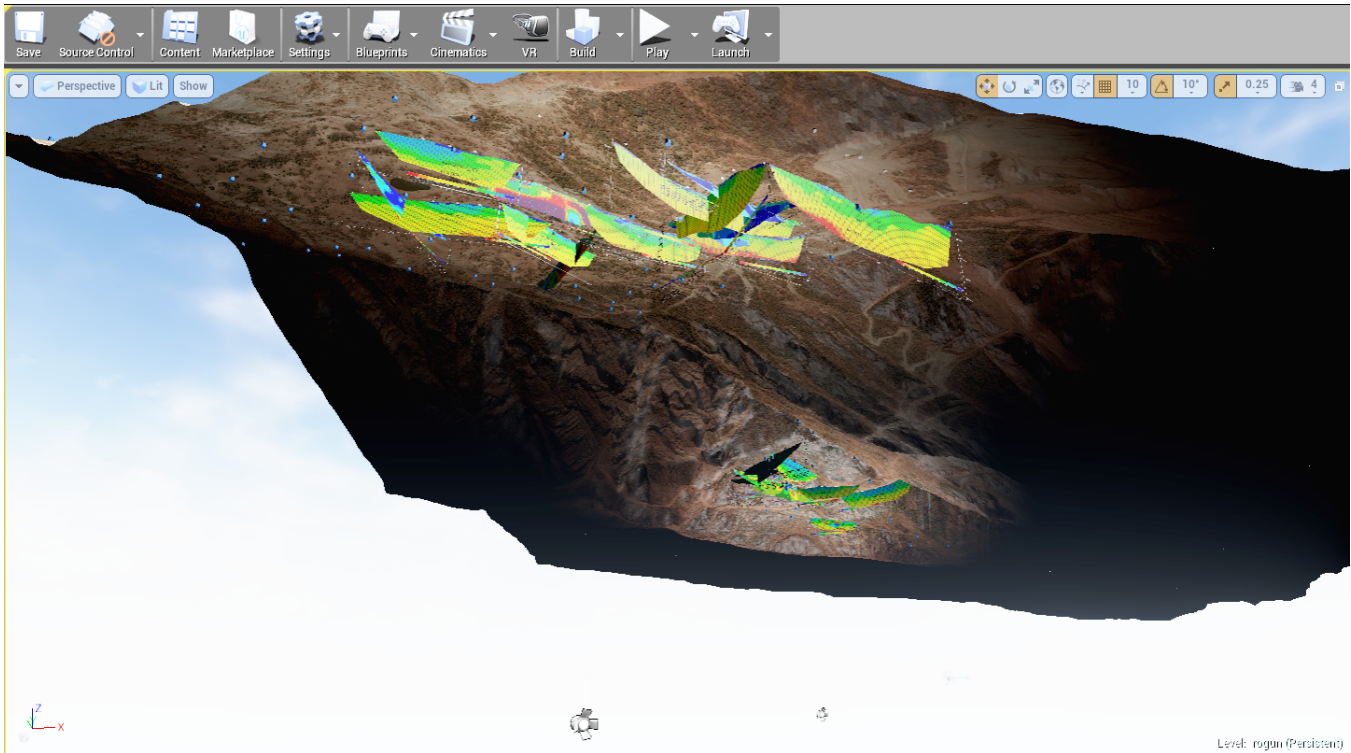


Fig. 3. Subsurface view of geophysical data with the Unreal Engine 4.

the VR helmet.

We tried it with the tedious task of placing the profile data according to the surface model. After you've become familiar with the controls, a profile could theoretically be placed with one move. In practice this requires several adjustments and tweaks. Although its fast, precision is an issue when manipulating far away objects with two picking beams to control the position, orientation and scale in one fell swoop as depicted in Figure 4.

Once you get used to the VR editor it becomes a very useful extension to prepare and inspect the final scene with the geophysical data. The problem that we had when showing our content in the VR editor was that the controls tend to be overwhelming for one time users. The obvious solution to this problem is to implement more simple interaction methods that would allow us to show and inspect our data without the fear of scaring people off or having them break the model every time they aim at and then click something with a motion controller.

We decided to use a simple fly through metaphor where pushing the trigger while holding the motion controller in one direction would move the viewer towards exactly that direction. The trigger button makes the most sense because it is a one dimensional analog input signal that we can use to adjust the movement speed.

5. BENEFITS AND IMPROVEMENTS USING IMMERSIVE VISUALIZATION AND INTERACTION

Before we get to the main part of this section where we compare the desktop and VR visualization of our models we are going to give the reader a brief rundown of the main differences between a HMD based VR system versus systems like a CAVE [4] or other large tiled display systems e.g. power walls.

Besides the obvious price tag difference that we already mentioned in the introduction there are other more practical considerations. On one hand, compared to lightweight tracked filtering glasses used in a CAVE, HMDs are less comfortable to wear. While the technology has certainly been improved weight and wiring are still problematic issues.

On the other hand, HMDs are clearly a more economic solution not only from a financial, but also from the spatial point of view as less room space is required to set up the system. Furthermore, HMD systems require less maintenance and related human resources.

While it is far from perfect at least large scale VR installations give you the possibility to let a few other viewers stand as closely as possible to the user that's benefiting from user centered projection and moving parallax. With HMDs there's no such option the only thing that's left for bystanders is to look at a screen view of the scene, although it could easily be projected to a wall.

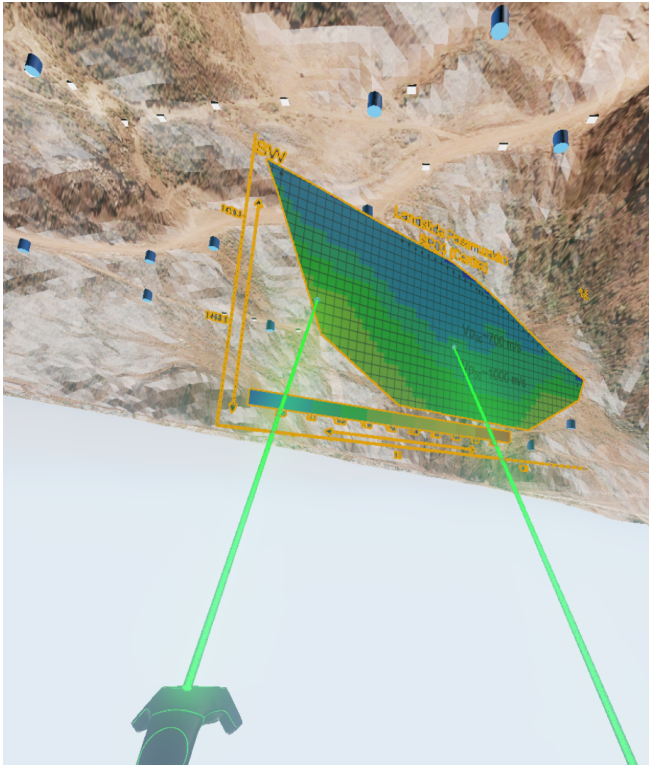


Fig. 4. Placing geophysical profile data using picking beams in VR editor mode.

Our last point is about visual quality. One thing to keep in mind when comparing the field of view of HMD based systems to a system with glasses, is that the visibility field with stereo vision is restricted by the glasses size. The most outstanding argument in favor of the visual quality provided by HMD systems is the implicit edgeless, squared 360 degree panoramic view that is impossible to match even with high end CAVE systems. For people that are subject to motion sickness seeing your own body helps so it is an inconvenience with HMDs.

Finally we focus on the fundamental question about the upgrade from desktop to VR visualization for geophysical data and its pros and cons. Even though the VR hardware has become much cheaper, it is still more expensive than a simple desktop system without a HMD. The high hardware specifications are certainly due to the fact that during rendering a lot of data has to be processed twice until the final images for the left eye and the right eye are ready (at frame rates that don't induce motion sickness). As expected it is only natural that higher frame rates can be achieved with the same data and system but on the monitor. Better depth perception is certainly one of the biggest assets of VR. The added benefit from first and foremost moving parallax and stereo vision are really stunning for new users. Depth perception for simple points and lines on desktop systems is impossible without moving the model. In VR you get depth cues without fur-

ther ado. The only thing you have to do is moving your head around the scene. This makes it easier than with a mouse and keyboard where you first have to learn the key bindings and button mapping before you understand what sort of trackball interaction you are dealing with. Even with a 6dof mouse the desktop can't compete because the device is designed for experts.

Once the model is scaled beyond the size of the tracking space the user needs a way to get around the scene. So VR interaction has to be included and custom tailored to the application, the available hardware and last but not least the different users. We consider this to be the make or break point for VR applications and demonstrations. This is where VR navigation helps if you can keep the interaction simple enough for presentation purposes. While from a developer's perspective it might be tempting to try to use all available degrees of freedom with all buttons the casual user is thank full for less. We found that once it has been tailored to the viewers degree of expertise it is faster to pick up than with basic desktop interaction. But the desktop visualization also has some advantages. One advantage that should not be neglected is that usually people don't get motion sickness just by looking at a monitor. Motion sickness only affects a small percentage of users, nevertheless it is still an issue for VR. A smaller problem is the over all readability of 2D text content in VR. Jitter and blur often hinder the viewer from reading comfortably. Readability can be improved through 3D font extrusion but in return this increases the polygon count. So it is only an option for short texts but then again it is probably a bad idea to read long texts with in VR anyway. To conclude this section we'll briefly touch the topic of sharing the visualization experience with others. When analyzing geophysical data more than one expert is usually required to get the most insight out of the 3D model. This reflects the different kinds of data sets and measurement methods that are involved. A collaborative VR solution would be required to adequately address this problem.

6. CONCLUSION AND OUTLOOK

The improved availability and quality of entry level VR systems should convince hesitant or reluctant potential users to try this technology before deciding to pursue or condemn VR for their application's needs. One thing in particular that we found out is that representing geophysical models with a HMD and only head tracking has become an easy task. Further we learned that user interaction with motion controllers can become complicated to learn. For now we have reduced it to simple scene navigation but we'll certainly add more interaction mechanisms over time. The next thing that we'd like to investigate in that context is shifts of different interaction paradigms based on user perspective and the model to user size ratio. Finally we noticed that distributed and/or co-located real time collaborative VR is necessary. This is

certainly a complicated task that involves network communication and data sharing topics [2]. In our opinion it is the only valid option to circumvent the problems we had trying to share the VR experience with a HMD based system. It is probably safe to assume that this is an issue for disciplines other than geophysics as well.

7. REFERENCES

- [1] James Ahrens, Berk Geveci, Charles Law, CD Hansen, and CR Johnson. 36-paraview: An end-user tool for large-data visualization, 2005.
- [2] Bérenger Arnaud, Guillaume Artignan, Jérôme Cance, Gabriel Delmas, Mountaz Hascoët, and Nancy Rodriguez. Coviz: cooperative visualization to facilitate sense making by groups of users. In *2011 15th International Conference on Information Visualisation*, pages 255–260. IEEE, 2011.
- [3] Epic Games Inc. Maryland corporation 620 Crossroads Blvd. Cary NC 27518. Unreal engine, 2016.
- [4] Carolina Cruz-Neira, Daniel J Sandin, Thomas A Defanti, Robert V Kenyon, and John C Hart. The cave: audio visual experience automatic virtual environment. *Communications of the ACM*, 35(6):64–73, 1992.
- [5] Lance Flavell. *Beginning Blender: Open Source 3D Modeling, Animation, and Game Design*. Apress, 2011.
- [6] Thomas Hilfert and Markus König. Low-cost virtual reality environment for engineering and construction. *Visualization in Engineering*, 4(1):1, 2016.
- [7] Jeffrey Jacobson and Michael Lewis. Game engine virtual reality with caveat. *Computer*, 38(4):79–82, 2005.
- [8] Oliver Kreylos, Gerald W Bawden, and Louise H Kellogg. Immersive visualization and analysis of lidar data. In *International Symposium on Visual Computing*, pages 846–855. Springer, 2008.
- [9] JL Mallet. Gocad: a computer aided design program for geological applications. In *Three-dimensional modeling with geoscientific information systems*, pages 123–141. Springer, 1992.
- [10] HTC America Inc. 13920 SE Eastgate Way Suite 400 Bellevue WA 98005 US. Vive, 2016.
- [11] J Van Aalsburg, MB Yikilmaz, O Kreylos, LH Kellogg, and JB Rundle. New frontiers in fault model visualization and interaction. In *AGU Fall Meeting Abstracts*, volume 1, page 1137, 2009.