Vol. 74 | No. 7/1 | Jul 2018
DOI: 10.21506/j.ponte.2018.7.3
PONTE Florence, Italy
International Journal of Sciences and Research

NAVIGATION SPHERE: OPTIMIZING VIRTUAL SPHERE FOR TERRAINS ANALYSES

Ph.D. Franklin Hernandez-Castro
Instituto Tecnológico de Costa Rica - Hochschule für Gestaltung Schwäbisch Gmünd
Costa Rica
franhernandez@itcr.ac.cr

Ph.D. Jorge Monge-Fallas
Instituto Tecnológico de Costa Rica
Costa Rica
jomonge@itcr.ac.cr

Abstract

It is commonly recognized that navigating in three-dimensional spaces can be a complicated task, primarily when working with "objects" that have natural orientations such as terrains. To resolve this problem, we limited user navigation to two types of rotations. This strategy prevents the user from becoming “lost” while “flying” through the environment and feel trapped by the tool. Also, based on the fact that there is always a center of interest, user navigation is limited to the sphere surface whit its center located in this interest area. This paper presents an overview of how this 3D navigation problem is resituated in the context of terrain explorations. Our goal is to help the user navigate 3D terrains by adapting the conventional Virtual Sphere navigation approach to the particular case of "terrain" explorations. We argue that the traditional schemes are too generic for some specific 3D uses, and we propose exploration constraints that are better suited to these particular tasks. We already used this approach, with optimal results, in different contexts, like hierarchical visualization, earthquakes visualization, earthquake animation and 3D wind visualization in Costa Rica.

Keywords

3D object exploration, view movement, user interface, interactive graphics, 3D graphics, rotation control, virtual trackball.

PONTE
Florence, Italy
International Journal of Sciences and Research

1. Introduction

It is commonly recognized that the use of a 2D mouse to analyze 3D geometry, such as terrains, is challenging within visualization environments. The problem is grounded in the inability of 2D input devices to execute 3D commands. The conventional, more common technique, involves the use of a virtual trackball with variations like Virtual Sphere or Sphere View. In these approaches, the point of view (camera) rolls across the surface of a sphere and the object that is the subject of the observation is positioned at the central point of this sphere (Castro, Montero y Fallas, 2009).

Modifying θ or ϕ by dragging or the arrow keys, the user can rotate the camera perspective around the surface (See Figure 1).

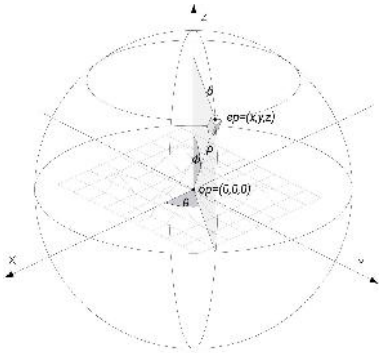

Figure 1: Typical configuration of Sphere View technique.

Where the center of the sphere is the center of the system: $o = (0,0,0)$
and the camera position is given by $c = (x, y, z)$ with $x = \rho\sin(\theta) \cdot \cos(\phi)$, $y = \rho\sin(\theta) \cdot \sin(\phi)$ y $z = \rho\cos(\phi)$ with $]0, m[$ and $m \in R^{+}$.
This technique is advantageous in many cases; however, when it is applied to terrains and objects, it exhibits some problems. For example, it makes no sense to rotate a terrain on the y-axes because the surface of the Earth always remains horizontal in the user mental model. We propose, in these cases, that the rotation possibilities are constrained to increase the ease of analysis.

2. Related Work

Although several methods of navigating using a seven degree-of-freedom view model have been proposed and evaluated, (Cho et al., 2017; McClymont et al., 2011), these have been designed to “fly” around as opposed to analyzing a particular object (or terrain). As such, they are not suitable for the evaluation of a terrain. Other techniques, such as those that involve the use of a virtual trackball, are much better for this task.

The first formal use of a virtual trackball was described by Chen et al. (1988). Another version of this concept is the Shoemake’s implementation (Shoemake, 1992) that extended the field of

the virtual trackball to the full screen. Later, these, and other approaches, were analyzed by Henriksen et al. (2004) in some depth. However, none of them took into account the particular case of terrains (or common 3D objects).

Furthermore, the techniques described above incorporate various navigation approaches. For example, Task Dependent Navigations involve techniques by which the point of view (POV) is automatically located. This method is often used within the field of medical visualization (Mindek et al. 2017), where the POVs are updated with the expectation that the users are given. In this context, Kohlmann introduced deformed viewing spheres to define the viewpoints (Kohlmann et al. 2007). However, in our case, it is impossible to automatically determine the POVs because, in our case, they involve the user performing the analysis on a real-time basis.

3. Methodology

We developed a simple interaction process to define each desired rotation and we follow the approach proposed in (Monge-Fallas & Hernandez-Castro, 2016). We assumed that the user wanted to rotate a terrain in two different ways (See Figure 2).

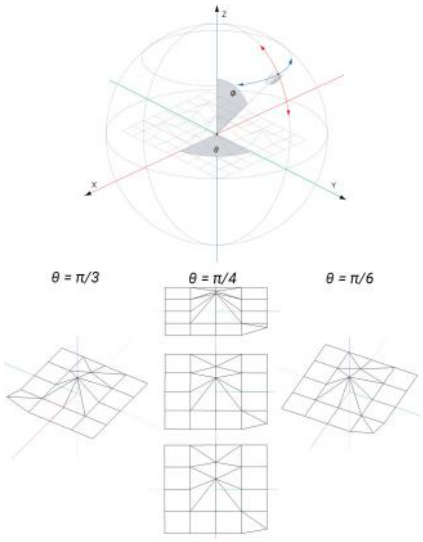

Figure 2: Rotating axes (only x and Z) proposed for terrains.

In the first case (rotating Z-axes), camera's position is calculated by changing the values θ and ϕ , as follows:

$$x = \rho\sin(\theta_0 + \theta) \cdot \cos(\phi_0)$$

$$y = \rho\sin(\theta_0 + \theta) \cdot \sin(\phi_0)$$

$$z = \rho\cos(\phi_0) \qquad \text{with } 0 \leq \theta \leq 2\pi.$$

On X-axes instead, we use:

$$x = \rho\sin(\theta_0) \cdot \cos(\phi_0 + \phi)$$

PONTE
Florence, Italy
International Journal of Sciences and Research

$y = \rho\sin(\theta_0) \cdot \sin(\phi_0 + \phi)$

$z = \rho\cos(\phi_0 + \phi)$ with $0 \leq \phi \leq \frac{\pi}{2}$.

In the case of terrains, we suggested that the user didn't want to rotate the Y-axes, so we limited the value ϕ, with $0 \leq \phi \leq \pi$. (See figure 3)

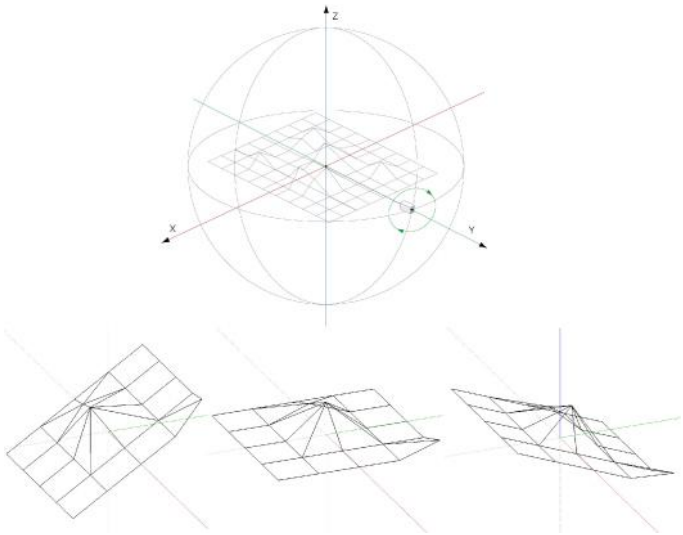

Figure 3: Rotating axes (Y) not recommended for terrains.

Because of this, we constrained the rotation possibilities on only two axes, as shown in Figure 2 (Z & X), and incorporated a zoom and pan effect (See Figure 4).

For the zoom effect, we modify ρ-value. If ρ decreases, we have a zoom-in effect and if ρ increases, we have a zoom-out.

For pan effect, we modify *k* and *h*:

$x_t = h + \rho\sin(\theta_0) \cdot \cos(\phi_0)$

$y_t = k + \rho\sin(\theta_0) \cdot \sin(\phi_0)$

$z_t = \rho\cos(\phi_0)$ with $k, h \in R$

so we get the new coordinates $pc_t = (x_t, t_t, z_t)$ and $op_t = (h, k, 0)$.

PONTE
Florence, Italy
International Journal of Sciences and Research

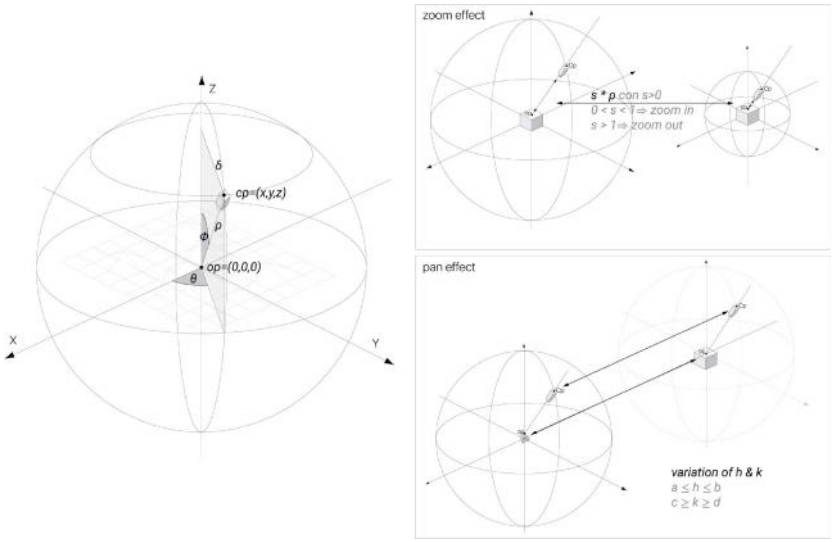

Figure 4: Implementing the Navigation Sphere.

To implement our version of the sphere view: "The Navigation Sphere," we used cylindrical coordinates.

The camera performs two rotations. The vertical circumference is always of the same diameter and the diameter of the horizontal circumference changes according to the latitude of the sphere. The entire system can be decreased or increased, providing the ability for the user to zoom in and out. In addition, the user can pan across the surface by translating the sphere center, making navigation easier and limitless.

The variables employed in this method are as follows:

- c: $= (\rho_0, \theta_0, \phi_0)$: camera position
- o $= (0,0,0)$: camera "look-at point"
- ρ: sphere radius
- □: parallel radius at camera position
- θ meridian angle where the camera is located
- ϕ parallel angle where the camera is located

Where, camera position in rectangular coordinates (c: $= (x, y, z)$) is calculated by:

$x = \rho\sin(\theta_0 + \theta) \cdot \cos(\phi_0)$

$y = \rho\sin(\theta_0 + \theta) \cdot \sin(\phi_0)$

$z = \rho\cos(\phi_0)$ also $\delta = \rho \cdot \sin(\phi)$

We use the mouse (or arrow keys) to move up and down and left and right, and the scroll for zooming. In addition, the option+arrow-keys are used for the pan effect.

As such:

- By pressing the up and down keys or moving the mouse up and down, we increase or decrease the θ angle:

$$\theta = (\theta_0 \pm \Delta\theta)m \quad 2\pi$$

- By pressing the left and right keys or moving the mouse left and right, we increase or decrease the ϕ angle:

$$\phi = (\phi_0 \pm \Delta\phi)m$$

- By scrolling, we increases or decrees the ρ:

$$\rho = (\rho_0 \pm \Delta\rho)$$

Using these, we can move the camera around the object.

We also use other limitations; for example, ρ can't be negative because this would mean that the camera had passed through the object and this wouldn't make sense for terrains. If the user wants to see the object from the opposite side, it should be rotated. This approach is more intuitive and transparent for the user.

The "upside" of the camera always has to be updated with the line between $c - o$. We employed OpenGL, C++ or Java to achieve this and could define it in each language. For example, in the case of C++ (openFrameworks):

- myCamera.setPosition(c);
- lookAtVector.set(o);
- upVector.set(0, 0, 1);
- myCamera.setPosition.lookAt(lookAtVector, upVector);

As a result of this implementation, we can see the figure 5.

Figure 5: Different points of views of Costa Rica map using the Navigation Sphere.

4. Evaluation

The Navigation Sphere has commonly been employed in several visualizations (from biodiversity to wind and earthquakes, figure 6) for many years. We tested the use of the Navigation Sphere to explore terrains and common objects with many scientists. Most of them commented that the Navigation Sphere was far more intuitive than the conventional approach to analyzing terrains and 3D objects. The navigation limits were not detected by the user in most of the cases, and they found this method allowed them to intuitively and naturally observe a real 3D object.

PONTE
Florence, Italy
International Journal of Sciences and Research

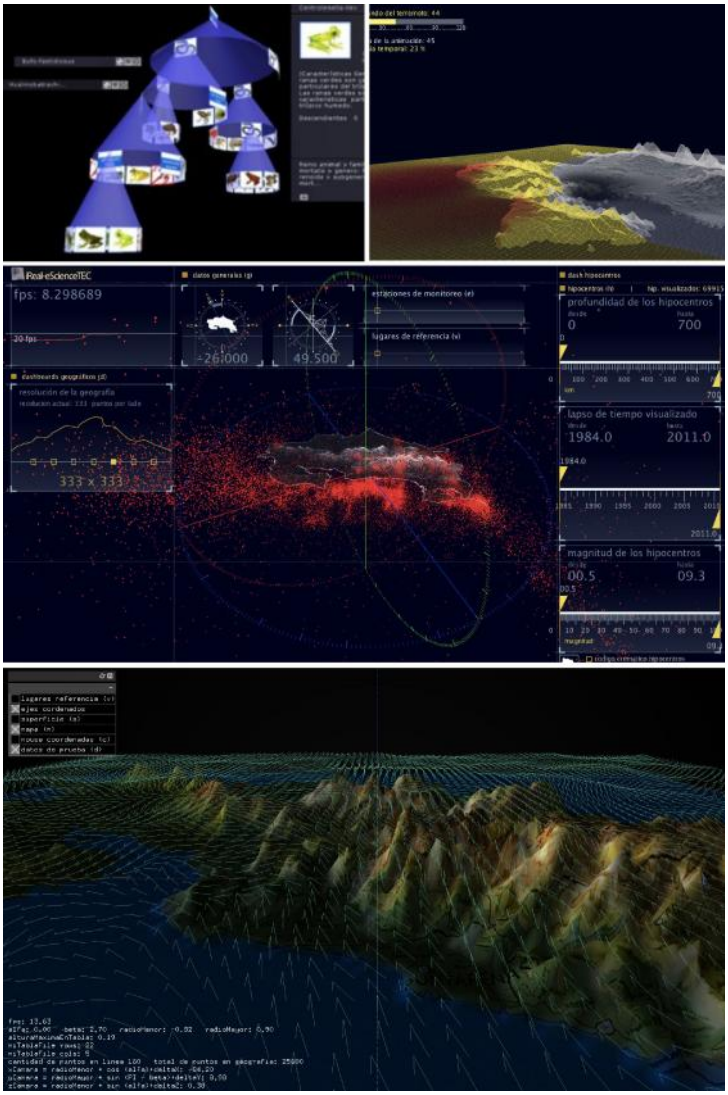

Figure 6: Different implementations of the Navigation Sphere from iReal (a research group of the Technological Institute of Costa Rica). From top to bottom: Biovisualizador 2007 (taxonomic visualization), Virtual Nicola 2016 (3D animation of an Earthquake), Plinius 2016 (3D hypocenters visualization) and SkyGraph 2017 (3D visualization of wind speed).

Additionally, we conducted a user study in which we compared the Navigation Sphere with the conventional approach. Scientists highlighted the need for dashboards that showed the rotation angles θ and ϕ. We implemented these, as shown in Figure 7.

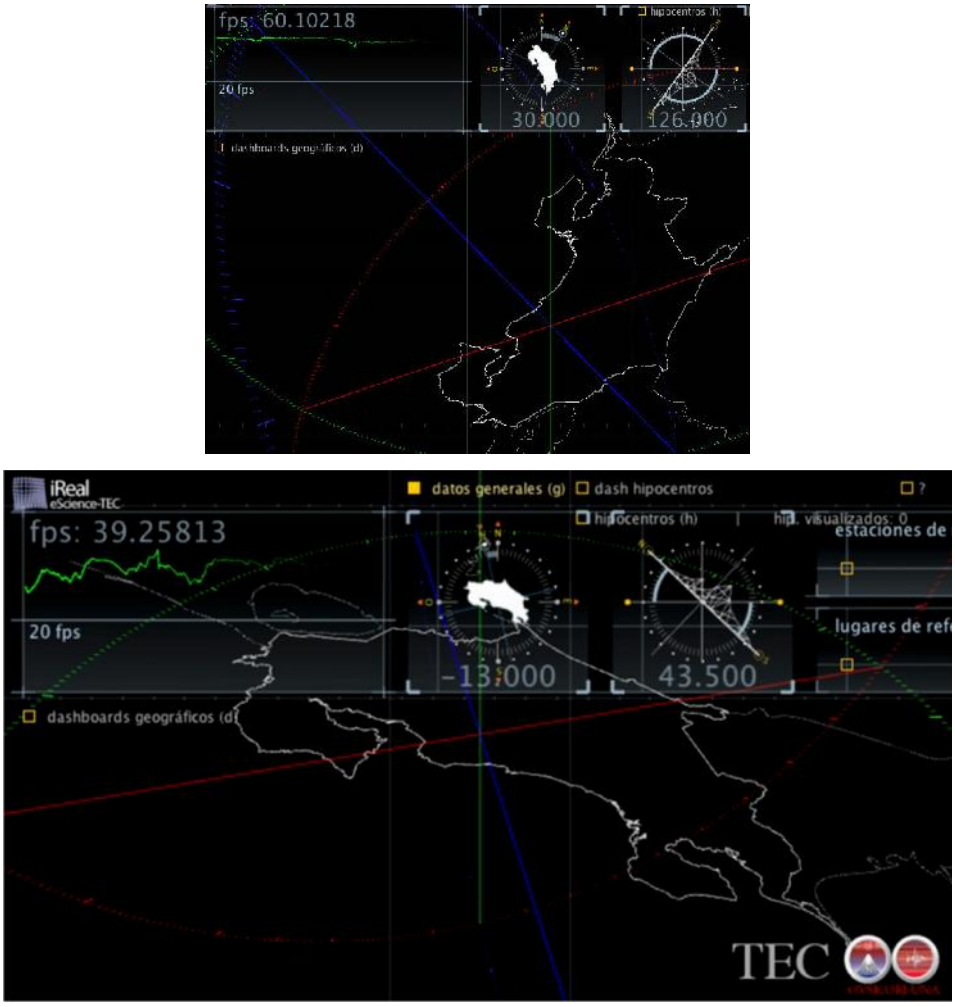

Figure 7: Dashboards showing θ and ϕ angles of the particular point of view.

The subjects were also asked post-use qualitative evaluation questions, including those that sought their insights into how the Navigation Sphere compared to the conventional approach and which of two methods they found to be more intuitive. Most of the participants (87%) commented that the Navigation Sphere was far more intuitive than the conventional approach.

These usability experiments suggested that the proposed user interface is quite intuitive for users to utilize with no training required.

We already used this approach, with optimal results, in different contexts, like hierarchical visualization(Castro, Montero & Fallas, 2009) , earthquakes visualization(Monge-Fallas & Hernandez-Castro, 2018), earthquake animation, 3D wind visualization in Costa Rica and other (Hernández-Castro & Monge-Fallas, 2016).

5. Conclusions & future work

In this paper, we described an intuitive interface that was designed to analyze efficiently and effectively terrains and 3D objects. This interface reduces the time required to examine and explore such kind of objects with natural orientations. The traditional navigation approach does allow users to "fly around" objects, and this confuses the user. Our Navigation Sphere approach limited the navigation options in a natural way that the user found quite easy to use without feeling limited.

Going forward, we (the iReal research group of the Technological Institute of Costa Rica) plan to visualize several 3D databases like Wind Speeds and Coast Line Shifting using this navigation approach.

We also plan to conduct usability studies that include identifying methods by which we can represent, analyze, and observe this kind of object in order to identify better and more intuitive navigation tools.

Acknowledgements

We wish to thank the Volcanological and Seismological Observatory of Costa Rica at National University, OVSICORI (http://www.ovsicori.una.ac.cr), the National Biodiversity Institute of Costa Rica (InBio), and The National Institute of Aqueducts and Sewers (AyA) for providing the databases that were used in our project. We would also like to thank the scientists who provided assistance as consultants, testers, and advisers.

References

Castro, F. H., Montero, E. M., & Fallas, J. M. (2009). Biovisualizador: Visualizando los anfibios de Costa Rica. Tecnología en Marcha, 22(1), 15-23.

Chen, M., Mountford, S. J., & Sellen, A. (1988). A study in interactive 3-D rotation using 2-D control devices. ACM SIGGRAPH Computer Graphics, 22(4), 121-129.

Cho, I., Li, J., & Wartell, Z. (2017). Multi-Scale 7DOF View Adjustment. IEEE Transactions on Visualization and Computer Graphics.

Dos Santos, C. R., Gros, P., Abel, P., Loisel, D., Trichaud, N., & Paris, J. P. (2000). Metaphor-aware 3d navigation. In Information Visualization, 2000. InfoVis 2000. IEEE Symposium on (pp. 155-165). IEEE.

Forghani, M., Vasev, P., Averbukh, V., & RAS, I.(2017) Three-dimensional visualization for philogentic tree. Scientific Visualization, 9(4), 59-66.

Hernández-Castro, F., & Monge-Fallas, J. (2016). Visualizador 3D de la geografía de Costa Rica**.** *Revista Tecnología en Marcha***,** *29*(8), 77-85.

Hernández-Castro, F., Monge-Fallas, J. (2016). What for: classification of visual paradigms. PONTE: International Scientific Researches Journal, 72(7), 46-64.

Henriksen, K., Sporring, J., & Hornbæk, K. (2004). Virtual trackballs revisited. IEEE Transactions on Visualization and Computer Graphics, 10(2), 206-216.

Khan, A., Komalo, B., Stam, J., Fitzmaurice, G., & Kurtenbach, G. (2005, April). Hovercam: interactive 3d navigation for proximal object inspection. In Proceedings of the 2005 symposium on Interactive 3D graphics and games (pp. 73-80). ACM.

Kohlmann, P., Bruckner, S., Kanitsar, A., & Gröller, E. (2007). LiveSync: Deformed viewing spheres for knowledge-based navigation. IEEE Transactions on Visualization and Computer Graphics, 13(6), 1544-1551.

McClymont, J., Shuralyov, D., & Stuerzlinger, W. (2011, September). Comparison of 3D navigation interfaces. In Virtual Environments Human-Computer Interfaces and Measurement Systems (VECIMS), 2011 IEEE International Conference on (pp. 1-6). IEEE.

Mantoro, T., Ayu, M. A., Azziz, U., Muhic, M., AbdulBagi, M., & Abubakar, A. (2016, October). VisUN-3D: User navigation with visualized 3D maps for mobile users. In Informatics and Computing (ICIC), International Conference on (pp. 377-382). IEEE.

Mindek, P., Mistelbauer, G., Gröller, E., & Bruckner, S. (2017). Data-Sensitive Visual Navigation. Computers & Graphics.

Monge-Fallas, J. & Hernandez-Castro, F. An Intuitive 3D Interface for Defining Seismic Profiles by Plinius. PONTE: International Scientific Researches Journal, 74(4), 2018.

Punpongsanon, P., Guy, E., Iwai, D., Sato, K., & Boubekeur, T. (2017). Extended LazyNav: Virtual 3D ground navigation for large displays and head-mounted displays. IEEE transactions on visualization and computer graphics, 23(8), 1952-1963.

Shoemake, K. (1992, September). ARCBALL: a user interface for specifying three-dimensional orientation using a mouse. In Graphics Interface (Vol. 92, pp. 151-156).

Takahashi, S., Fujishiro, I., Takeshima, Y., & Nishita, T. (2005, October). A feature-driven approach to locating optimal viewpoints for volume visualization. In Visualization, 2005. VIS 05. IEEE (pp. 495-502). IEEE.

Theng, Y. L. (1999, January). " Lost in hyperspace" problem revisited and framework for building digital libraries. In COLLOQUIUM DIGEST-IEE (pp. 1-1). IEE; 1999.

Vázquez, P. P., Feixas, M., Sbert, M., & Heidrich, W. (2001, November). Viewpoint selection using viewpoint entropy. In VMV (Vol. 1, pp. 273-280).

Letter:

Our manuscript is just a contribution in 3D navigation for terrains or 3D common objects. It is the conclusion of several years of test with different visualization in 3D space.

We note that is much better for the user if we limited the navigation possibilities, the limits were not detected by the user in most of the cases, and they found this method allowed them to intuitively and naturally observe a real 3D object.

Perhaps it can be useful for other researches in this fields.

Bullets

- *Several methods of 3D navigating, such as those that involve the use of a virtual trackball have been designed to "fly" around as opposed to analyzing a particular object or terrain.*
- *We propose, that the rotation possibilities are constrained to increase the ease of analysis.*
- *We suggested that the user didn't want to rotate the y-axes and constrained the rotation possibilities on only two axes and incorporated a zoom and pan effect.*