Advanced View-dependent Techniques for Digital Terrains Visualization

Rafael Moreira Savelli
Roberto de Beauclair Seixas
The National Institute of Pure and Applied Mathematics - IMPA
Visgraf Laboratory
Estrada Dona Castorina, 110 - Rio de Janeiro, Brazil
{savelli,rbs}@impa.br

Abstract

This work presents some methods to improve visualization of a digital featured terrain. We used the well-known billboard structure and applied approaches such as transparency technique and some view-dependent control like complex object culling and multi-resolution imaging. This work also builds a billboard hierarchy gathering into groups those ones with similar properties. In run-time, the application maps and identifies the correct billboards categories for each scene, getting the most from realism in digital terrains.

1. Introduction

Virtual environments and 3D visualizations have been contributing in many ways to humans. From education to training, three-dimensional information in many cases is essential to understand the whole problem. However, the terrain area and thus the space problem tend to become larger and larger in order to fulfill humans specifications. The point is that the more large your terrain is the more objects must be redrawn in every frame and thus the less frame rate your visualization tend to be.

In order to balance this frame per second rate, this work uses some adaptive methods without compromising the whole visualization. The idea is quite simple and consists in cutting out cpu time processing avoiding unnecessary rendering.

This work is based on previous results where terrain texture are retrieved from digital satellite image [7]. Moreover, this application also uses texture for retrieving objects positions, more specifically, billboards positions and place them in the geo-referential digital terrain. It consists in a semi-automatic process for featuring terrains based on its own satellite image.

2. Background

Previously, many works about digital terrains have been written and some of them made great contributions to this work, specially those which deal with textures and objects. One of them is the work [8] where authors builded terrains with three-dimension modeled objects. In our case these objects models could not be implemented because we have much more objects to handle and modeling is known as an expensive solution.

A second interesting work is described in [2] where authors deals with terrain textures and some important interactive problems.

Finally, the work [7] showed how to get the most from satellites images and how to manage texturing information making terrain visualization much more realistic.

3. Terrain and Billboards Modeling

Algorithms for interactive visualization of terrains are very complex. Due to such complexity and importance, in the past decade this subject has received great attention by researchers on Computer Graphics. As a consequence, a number of strategies have been developed. Among the most successful strategies, one can highlight recent works by Lindstrom and Pascucci [6, 4] both applied to this work.

Basically, the fundamental elements for building a 3D terrain in our case was only a combination of triangulated wireframe with a geo-referentioned texture applied to that wireframe. Having these two elements combined we can figure out many geographic features like mountains and valleys. Yet from a given 2D colored texture, features like rivers and lagoons also become evident. The result achieved is a 3D terrain showing only elevations and other primitive terrain proprieties [1].

Another important consideration include billboarding technique for representing other kind of object such as vegetation. By using this technique, we can have trees, grass, and other kind of plants in the digital terrain.

A billboard is a simple texture-mapped geometry that increases realism and performance when compared to the traditional intricate geometry. More than that, billboarding is a technique in which complex objects are drawn with simple planar texture mapped geometry and the geometry is always transformed to face the viewer. The transformation typically consists of a rotation among the z axis to orient the object towards the viewer and a translation to place the object in the correct position [9]. For the case of the tree, an object with roughly cylindrical symmetry, an axial rotation is used to rotate the geometry for the tree, typically a quadrilateral, about the axis running parallel to the tree trunk. The Figure 1 shows a schematic typical billboard.

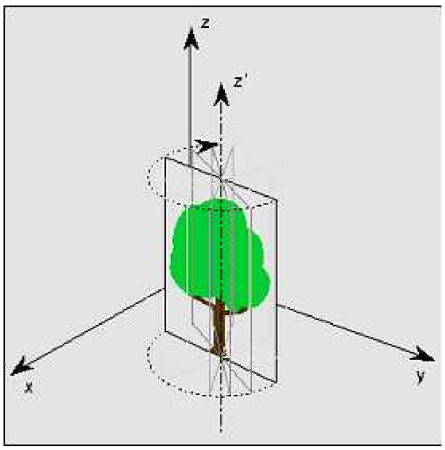

Figure 1. Billboard concept.

To increase realism, a fourth component, named alpha-channel, was added to the rgb billboard image. This fourth component is responsible for ruling how much opaque or how much transparent the texture is in a given pixel. The general resulting rgba image texture tend to be transparent near borders and opaque in the object, increasing the idea to be inserted into scene context [5]. As an example the Figure 2 shows a group of this preprocessed billboards.

We also tried the common static orthogonal billboards for our scene objects. This orthogonal billboard composition is a simpler approach which consists of placing not one but two billboards per object. These billboards must form a cross-shape with orthogonal angle between them. An example can be seen in Figure 3. The idea was to avoid every billboard rotation but results do not attended to our expectative. In practice, we observed that a single billboard rotation cost less than rendering two static billboards. As we can see in table 1, the result was quite the same for both strategies when a reduced amount of object is used. However, when too many objects must be drawn the results for orthogonal billboard get worst and becomes a poor solution when compared to traditional billboard. This table 1 values were obtained from the visualization of *Itaoca* terrain running on a intel Pentium 4 1.6 Ghz machine with 512 MB of RAM memory and 256 MB GeForce FX5200 graphic card.

Number	*Frame per second*	
of billboards:	Traditional:	Orthogonal:
50	40.0	40.0
100	40.0	40.0
200	33.3	25.0
300	28.6	20.0
400	25.0	16.7

Table 1. Performance on *Itaoca* terrain

Figure 2. A traditional billboard example.

4. View-dependent Techniques

As we know, large terrains also have many object all around their surface. Depending on terrain size the need to redraw every billboard per frame become intractable. For that reason, some tricky and smart artifices are needed in order to deliver more cpu time for rendering without damaging the entire terrain scene visualization. Another way is to use these artifices just to improve the visualization quality with less cpu effort.

That is what we call view-dependent technique. We use some adaptive methods depending on the observer scene

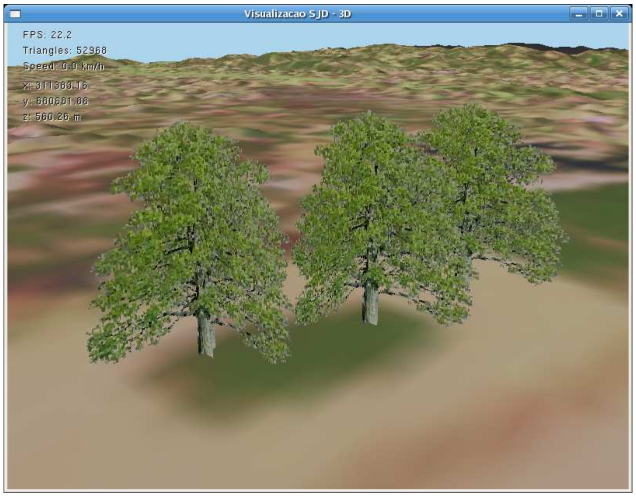

Figure 3. An orthogonal billboard example.

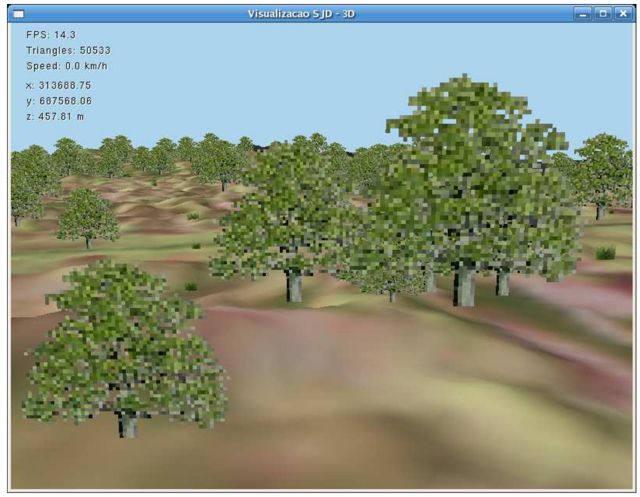

Figure 4. Random height to improve realism.

position, showing him only what is indispensable for a good visualization at his point of view.

For implementation we use C and C++ programming language in order to facilities related to memory access. Another important programming language used was LUA [3] which helps us to load billboards attributes.

On the following, we present the artifices used in this work as well as partial results gathered.

4.1. Random Height

This technique consists in improving the visualization quality only by scaling randomly different billboards height. The main idea of this approach is to make effort on leaving the whole visualization with some natural aspects. We all know that real vegetations are not the same height and thus the application can not have trees, bushes and grass replicas with the same height too. This trick is quite cheap to implement because small variances in billboard height doesn't mean necessarily to change the billboard texture resolution. Figure 4 shows how some billboards looks up in the terrain scene.

4.2. Billboard Clustering

Another important consideration is the fact that very distant vegetation tend to appear to observer not many populated when compared to closer vegetation. We used this natural eye effect in our favor by adding this concept digitally to our case study terrains. The observed results were very satisfactory and it can be seen in Figure 5.

In that picture, note that vegetation goes until a maximum distance from the observer. After that line of view (top of figure), the application consider that it can be no longer seen by the observer and thus, discard any type of vegetation. From that point to the observer position, the amount of billboard increases proportionally and become more and more dense until reaches the observer (at the bottom of the figure) where we it has a maximal billboard density.

Figure 5. Billboard clustering.

4.3. Multiresolution Billboards

The last effort to release cpu time is the fact that we can deal with multiple texture images with different resolution depending on distance between a given billboard and an observer. As you can see in Figure 7, we presents two examples of textures used in visualizations. The right tree in the figure shows a poor texture with only 64x64 pixels resolution. Yet the left one shows the same tree now with 256x256 pixels resolution.

The main idea involved in this approach is to save processing by decreasing billboard texture resolution while billboards get far away from observer. For our experiments, we used three kind of resolutions uniformly distributed along the observer sight. We use 256x256 pixels texture for close enough billboards and only 64x64 for those ones far away from observer. Figure 6 illustrates texture resolution arrangement as a function of distance where $d3 > d2 > d1$.

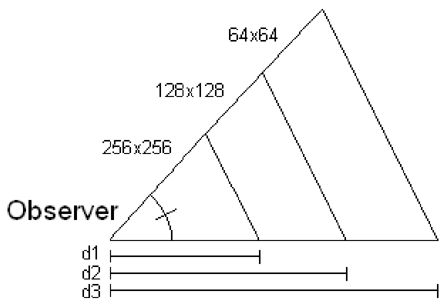

Figure 6. Billboard texture resolution according to distance.

Figure 7. Two sample billboard resolution.

5. Combined Results

Every single view-dependent technique contributes in a way for reducing cpu time. However, each technique separately does not make substantial differences in the visualization rendering because their benefits are modest and also limited. But, knowing how to combine these techniques in order to make real differences in visualization quality and render was the key to this work success.

We started to combine view dependent techniques by adding the billboard clustering with multiresolution billboards. This was quite easy because once divided the observer view into areas as shown in Figure 6, defining a percentage clustering for each area was simple. For instance, we draw in 256x256 area 100% of total existing billboards. On the 128x128 area, only 80% are drawn. Finally, in 64x64 area less than 50% are effectively drawn. After distance d3, no billboard are drawn.

After that, the random height technique was added to all areas. The final result was a great realism scene with relative low frame rate. Figures 8, 9 and 10 show combined view dependent techniques.

6. Future Works

One of the most obvious future works is to research new techniques of view-dependent for improving even more the quality of complexities digital terrains.

Another work could be an improvement the existing view-dependent techniques by making them even better. For example, when billboard is far away from observer it tends to be suppressed because your level of density can achieve zero, removing it from terrain. Instead, we could study the possibility of doing a fade out on those objects. The expected effect would be an object disappearing just a little at a time until is completely vanished from terrain.

Another good extension would be the ability to recognize objects like buildings and roads, most common in urban satellite pictures.

7. Conclusions

The first conclusion of this work is the fact that traditional billboards works better than orthogonal ones. By experiments, we could find a relation between the behavior of both techniques with the visualization frame rate. In addiction, working with just a few amount of objects make us understand that both strategies are good.

The second conclusion refers to the great importance that view dependent strategies are to recently 3d visualizations. The study of each strategy in separately is important to understanding it however, great results can only achieved when combined with other view dependent strategies. At

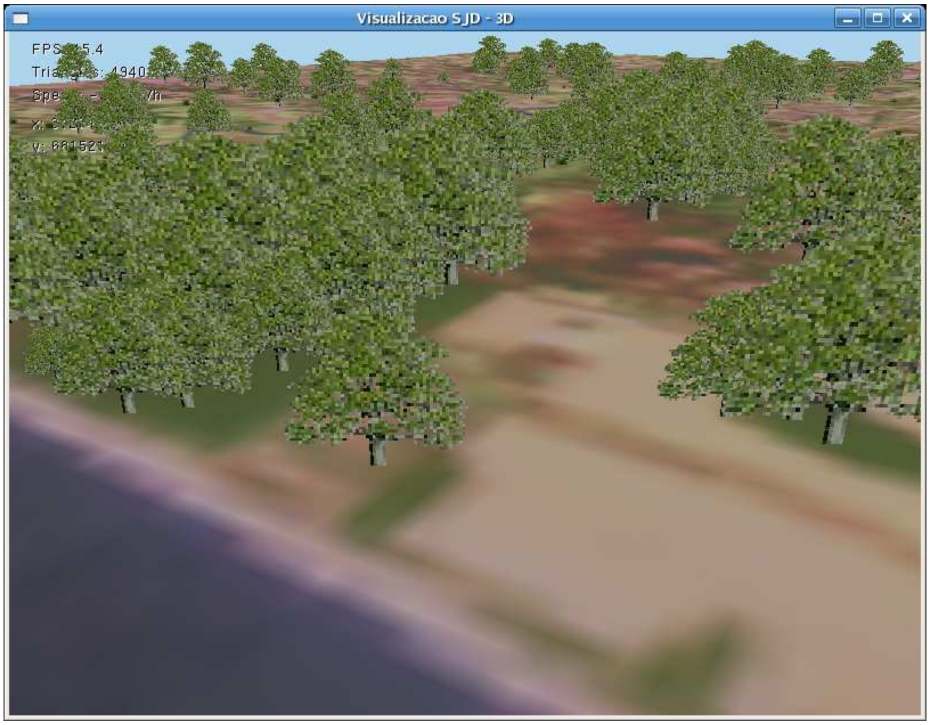

Figure 8. Results for combined view dependent techniques.

this work we combined three view dependent techniques and the final results observed were very satisfactory.

In a near future, terrains tend to become even larger and dealing with such amount of data will require a better special care. Computer graphics cards, processors and memory are still being improved. However, not only hardware make differences but also the combination of hardware and software. In our opinion, research on view dependent area is extremely important in order to support larger projects that is just about to arise.

8. Bibliography

References

[1] R. B. S. E. Poyart, Paula Frederick. Simple real-time flight over arbitrary-sized terrains. *Workshop Brasileiro de GeoInformtica*, 2002.

[2] P. Frederick. Visualizao interativa tridimensional de modelos de terreno com textura. *SIBGRAPI*, pages 341–342, 1996.

[3] R. Ierusalimschy. *Programming in Lua*. Lua.Org, 2006.

[4] V. P. Lindstrom P. Terrain simplification simplified: A general framework for view-dependent out-of-core visualization. *IEEE Transactions on Visualization and Computer Graphics*, May 2002.

[5] T. D. M. Woo, J. Neider. *OpenGL Programming Guide: The Official Guide to Learning OpenGL*. Addison-Wesley Professional, 1999.

[6] V. P. P. Lindstrom. Visualization of large terrains made easy. *Proceedings of IEEE Visualization*, pages 363–370, October 2001.

[7] R. Savelli, R.M.; Seixas. Semi-automatic detection of vegetations in digital satellite images for building 3d terrains. *The Sixth Iasted Internacional Conference On Visualization Imaging And Image Processing*, (0531-046):494–498, August 2006.

[8] C. P. Szenberg F., Gattass M. An algorithm for the visualization of a terrain with objects. *SIBGRAPI*, 1997.

[9] D. B. Tom McReynolds. Advanced graphics programming techniques using opengl. *SIGGRAPH*, 1998.

Figure 9. Results for combined view dependent techniques.

Figure 10. Results for combined view dependent techniques.