Proceedings of the Sixth IASTED International Conference
Visualization, Imaging, and Image Processing
August 28–30, 2006, Palma de Mallorca, Spain
ISBN Hardcopy: 0-88986-598-1 / CD: 0-88986-600-7

SEMI-AUTOMATIC DETECTION OF VEGETATIONS IN DIGITAL SATELLITE IMAGES FOR BUILDING 3D TERRAINS

Rafael Moreira Savelli
Roberto de Beauclair Seixas
Instituto Nacional de Matemática Pura e Aplicada – IMPA
Laboratório de Visualização e Computação Gráfica - VISGRAF
Estrada Dona Castorina, 110 – Rio de Janeiro, RJ – 22460-320
Brazil
{savelli,rbs}@impa.br

ABSTRACT
This work presents a method to construct featured 3D terrains using information gathered from their satellites images. This data is used as keys in an image-database search. These searches use wavelets for minimize disk space and improve searching speed. In order to compose the elements with trees or bushes, for example, we use the common billboard concepts embedded with transparent technique named alpha-channel.

KEY WORDS
Computer vision, image processing, object recognition and terrain visualization

1. Introduction

Virtual environments and 3D visualizations have been contributing in many ways to humans. From education and training to space exploration, three-dimensional information in many cases is essential to understand the whole problem. For example, could you imagine how an exploration to Mars would be without a real 3D visualization of its surface? How could you design a path for your remote robot without seeing the surface?

So, in certain circumstances, the more detail you have in some 3D visualization the more chance you have to understand your problem. And thus, you can treat it in the most proper way.

This work presents a method to considerably improve a virtual environment by semi-automatic detecting some of its vegetation on a typically Brazil terrain. The method also uses a satellite image of the given terrain for extracting the vegetation's geo-referential positions.

2. Terrain Modeling

For our case study the analyzed terrains have many features such as mountains, rivers, lagoons, vegetations and etc. So, for representing features like mountains and valleys a hypsometric layer was added to the 2D terrain. When you add the hypsometric layer, a 3D wireframe is constructed. For considering features like rivers and lagoons for example, a 2D texture is applied. The result achieved is a 3D terrain showing only elevations and other primitive elements. The figure 1 shows how to compose the terrain [1].

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 [2,3].

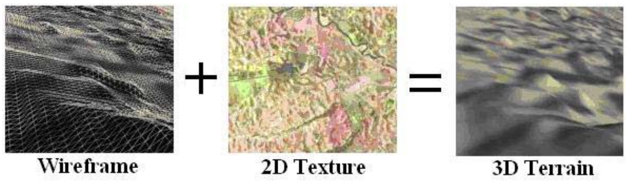


Figure 1: The 3D Terrain Composition

3. Billboard

A billboard is a simple texture-mapped geometry that increases realism and performance when compared to the traditional intricate geometry. In addition, 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 [4].
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 2 shows a schematic typical billboard.

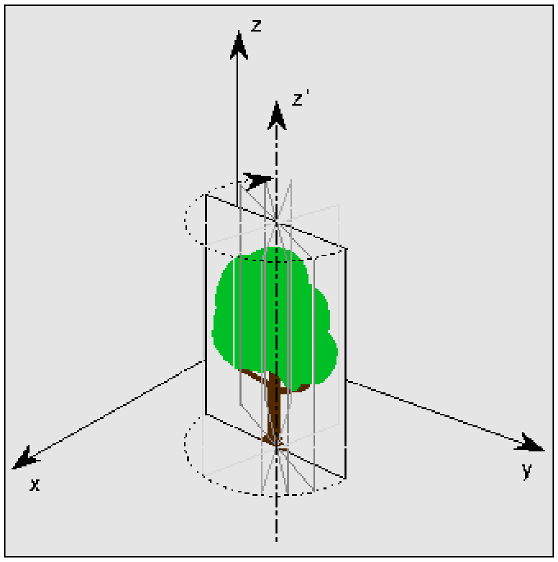


Figure 2: A Billboard and Its Axis

To increase realism, a fourth component, named *alpha-channel*, was added to the *rgb* image. We add this component by giving the *rgb* original image and it's negative to a simple Phyton program which creates and attaches the alpha-channel component to its original image. The resulting *rgba* was transparent near the borders and opaque in the object, increasing the idea to be inserted into context [5].

4. Digital Satellite Image

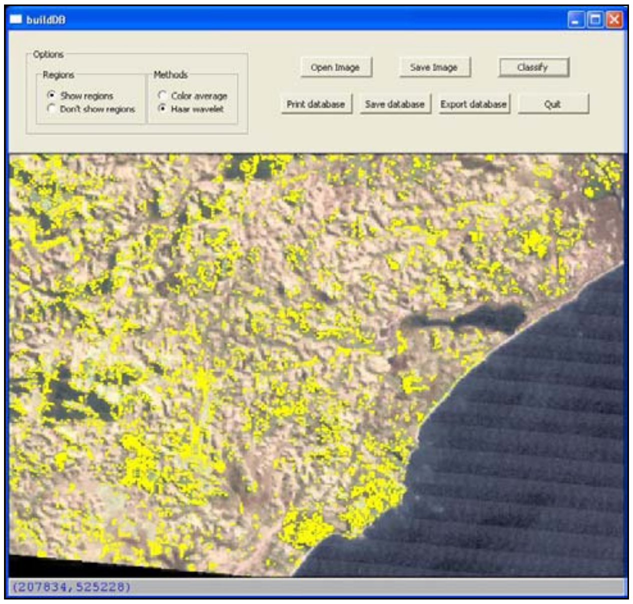
A digital satellite image is a digital picture of the earth taken by and sent from an earth-orbital satellite. This work uses this tool to read all digital image pixels trying to get the current position geo-referential in the terrain where elements are positioned.

We implemented a pre-classification application named "buildDB" (shown in figure 3). This application implements the algorithm *split-and-merge* [6] which searches for homogeneous regions by splitting the image recursively in four equal parts. When a given *threshold* is reached, we say that the entire square is homogeneous. In figure 3, a typical result of a *split-and-merge* algorithm in action can be observed. This application interface was made with IUP (Portable User Interface) [7], CD (Canvas Draw) [8] and IM (Imaging Tool) [9]. All these libraries were created by the TeCGraf/PUC-Rio laboratory [10].

Both approaches can be tested as a threshold for the *split-and-merge* stop criteria: Simple *average* and *Haar-wavelet*. The first one stops splitting when every color component RGB satisfies:

$$\textbf{average} = \textbf{sum} / \textbf{box\_size},$$

where **sum** means the sum of all pixels in the box of size **box_size** (where box_size means width x height of a given region). So, when the average achieves the given threshold, the splitting algorithm stops and this regions is



Regions: [✓ Show regions] Don't show regions

Methods: Color average [✓ Haar wavelet]

(207834,525228)

Figure 3: Region split-and-merge on Macaé City, RJ. There are 803713 Regions Found

considered homogeneous.

The process is similar to the *Haar-wavelet* and its algorithm is given section 6.

With the output image provided by the *split-and-merge* algorithm, we detect many different regions with the same property. Those regions were cataloged in an image-database with a single billboard associated to it.

After several tests, we conclude that the *split-and-merge* algorithm works very well for most satellite digital image we have, specially the ones with resolution equal or above the scale 1:25.000.

5. Database

From the *split-and-merge* algorithm, we know that many regions will be classified as being homogeneous. Those homogeneous regions are stored in a database file. When the visualization application is loaded, it reads from this database all information related to this billboard, such as size and position.

As we can see, the whole process needs basically to access only one database. To be more specific, a single table. This table must store all image cells and its corresponding billboard.

6. Wavelet

In order to identify an element from the satellite image we must perform a pattern recognition algorithm. But as we know, in the design of pattern recognition systems, a potentially large set of data must be computed. So, after the *split-and-merge* algorithm, we must take each generated region and compare with those previously stored in *database*. Those pre-stored images are generated by applying *wavelet* algorithm to them.

Wavelets are useful mathematical tools with a large sort of applications. One of these applications is related to signal and image processing. In the image processing segment, wavelets often help saving disk space by compressing large images. And this is exactly what we needed.

Nowadays we have many variations of wavelets such as *Daublet*, *Gabor*, *Laplacian*, *Haar* and others. We opt for *Haar-wavelets* for two reasons: first, because of its simple implementation method and second because it works very well for our tests.

The *Haar-wavelet* algorithm is given as following [11]:

Imagine that you have a RGB image with dimensions 4x4 only. All RGB components are represented by vectors with values from zero to 255. Just for illustration, imagine the Red component has the following values:

RED:

0	1	2	3
254	137	56	13

So, for each pair in the vector, evaluate the average between them and put the result in another new vector. So,

$$\left\lfloor \frac{RED[0]+RED[1]}{2} \right\rfloor = \left\lfloor \frac{254+137}{2} \right\rfloor = 195 \text{ and}$$
$$\left\lfloor \frac{RED[2]+RED[3]}{2} \right\rfloor = \left\lfloor \frac{56+13}{2} \right\rfloor = 34$$

With these operations, the new vector comes to:

NEW:

0	1	2	3
195	34		

The next step is to fill the white spaces in the vector *NEW*. This is simple by evaluating the following differences:

$$NEW[2] = RED[0] - NEW[0] = 254 - 195 = 59$$
$$\text{and } NEW[3] = RED[2] - NEW[1] = 56 - 34 = 22$$

Resulting,

NEW:

0	1	2	3
195	34	59	22

In this point of the *haar-algorithm*, the vector NEW was filled and we finished one iteration. Now, all averages and differences must be applied to elements in the *NEW* vector and just for those of the first-half (index 0 and 1). So, the next iteration becomes:

NEW:

0	1	2	3
114	81	59	22

For this simple example wavelet algorithm is now finished.

For storing in database, the process must be applied to all components RGB. After this, the image is ready to be stored in the database.

7. Joining Information

The main idea involved in this work is simplified as described in figure 4:

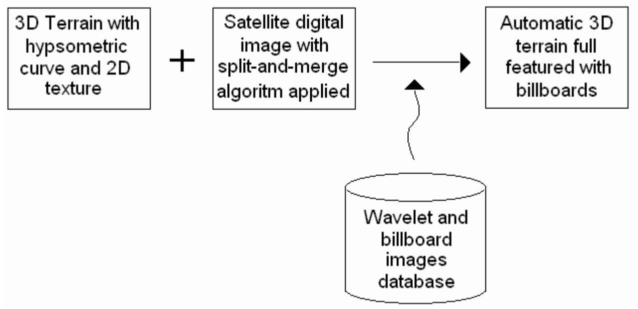
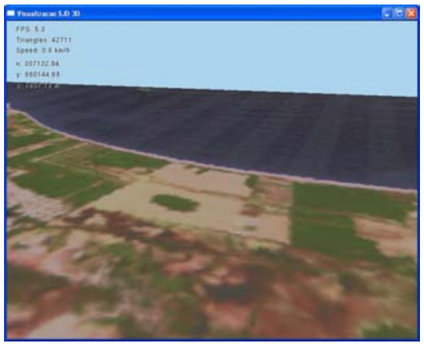

3D Terrain with hypsometric curve and 2D texture
+
Satellite digital image with split-and-merge algoritm applied
Automatic 3D terrain full featured with billboards
Wavelet and billboard images database

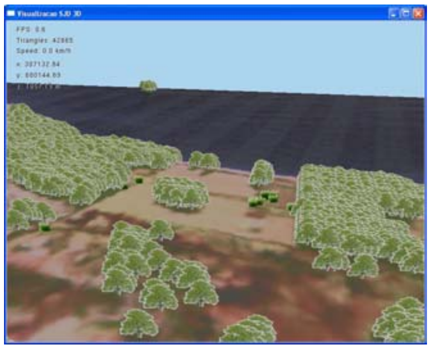
Figure 4: The Whole Process for Automatic Building Featured Terrains

With a certain terrain and its satellite image, you can add features by putting all detected elements in a database. So that the visualization application can consult from this data base and place all billboards in the right positions.

8. Case Study

The methods and algorithms were applied basically in two digital satellite image of a typical tropical Brazilian terrain. These images can be downloaded at Embrapa's site [12]. The final results are shown on the following. In figure 5 we present Itaoca, ES. And in figure 6 we present Macaé, RJ. Both regions from left to right represent: the empty terrain, the terrain with average applied and the terrain with wavelet applied.


(a) Empty


(b) Average Method

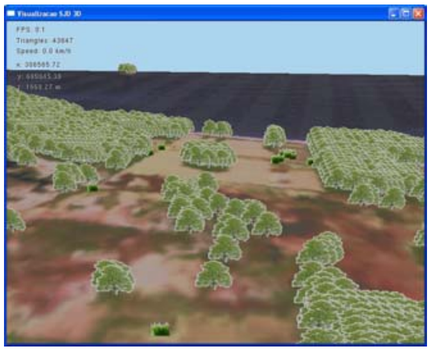
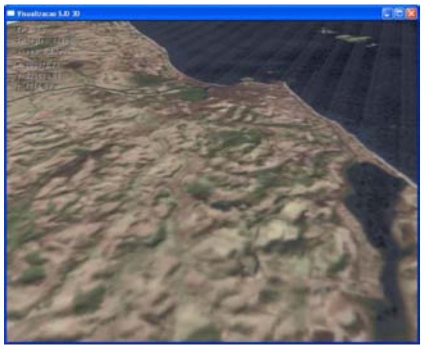
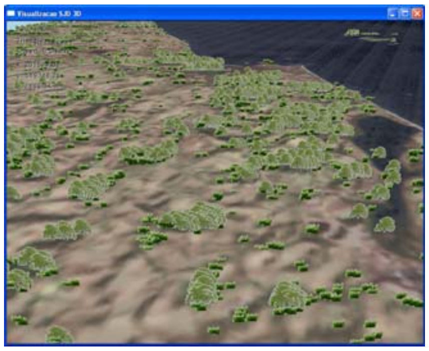

(c) Wavelet Method

Figure 5: Itaoca, ES Terrain


(a) Empty


(b) Average Method

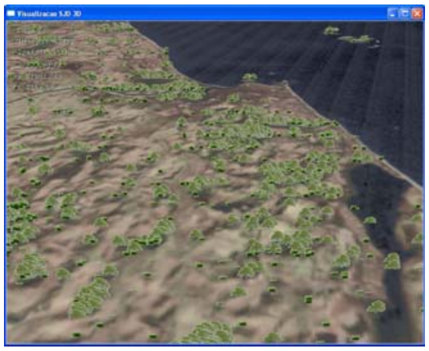
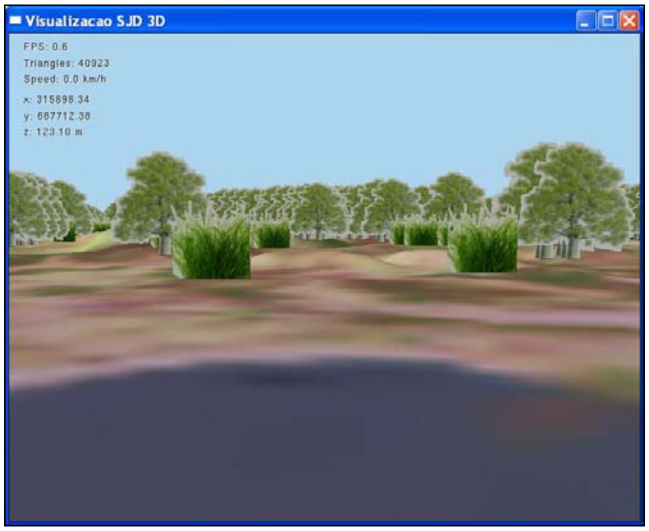

(c) Wavelet Method

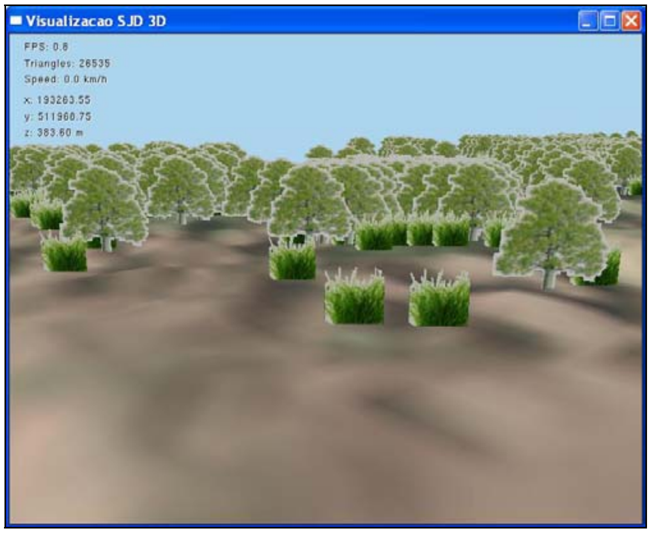
Figure 6: Macaé, RJ Terrain

Now, a closer view from both Itaoca and Macaé regions are shown in figures 7 and 8:



FPS: 0.6
Triangles: 40923
Speed: 0.0 km/h
x: 315898.34
y: 687712.38
z: 123.10 m

Figure 7: A Closer View from Itaoca Billboards



FPS: 0.8
Triangles: 26535
Speed: 0.0 km/h
x: 193263.55
y: 511960.75
z: 383.60 m

Figure 8: A Closer View from Macaé Billboards

When looking carefully at figure 5(a) we note that most green areas are covered with billboards in figure 5(b) and 5(c). Specially when we consider the isolated green portion in the middle of the 5(a) figure. This demonstrates how precise the algorithms can be, by placing billboards in the right positions.

9. Conclusion

In classification process, wavelets show that they really work well for recognizing vegetation in digital satellite image. In general, it could find and detect objects slightly better then the average method. In addiction, the right positioned billboards increased realism helping to interpret real situations.

10. Future Work

As we saw in this paper, billboards really increase realisms in terrains visualization. But their intrinsic characteristics also imply bringing other problems. For example, from an orthogonal terrain view, no trees or vegetations are seen from above because billboards are flat and can not be rotated in their x and y axis (otherwise, we would see a bended tree). So, basically, billboards are good but also limited to low viewing azimuths.

Another problem is relative to billboard image quality. If you look at a single billboard from sufficiently far away, the resolution of its image can be poor and probably will not affect the visualization. On the other hand, when you come closer enough to this billboard a better image quality is needed. Otherwise, users may view this element pixelated. A good approach to solve this problem would be considering multi-spectral and multi-resolution images. In this case, we must develop a full view-dependent visualization. This means that users will see objects depending on theirs own perspectives given by the application camera.

References

[1] E. Poyart, P. Frederick, R.B. Seixas, M. Gattass, Simple Real-Time Flight Over Arbitrary-Sized Terrains, pre-print submitted to Workshop Brasileiro de GeoInformática, 2002.

[2] P. Lindstrom, V. Pascucci, Visualization of Large Terrains Made Easy, Proceedings of IEEE Visualization, San Diego, California, October, 2001, 363-370.

[3] P. Lindstrom, V. Pascucci, Terrain Simplification Simplified: A General Framework for View-Dependent Out-of-Core Visualization, IEEE Transactions on Visualization and Computer Graphics, May 8, 2002.

[4] Advanced Graphics Programming Techniques Using OpenGL, SIGGRAPH `98 Course.

[5] M. Woo, J. Neider, T. Davis, D. Shreiner, *OpenGL Programming Guide: The Official Guide to Learning OpenGL* (Addison-Wesley Professional; 3rd edition, August 6, 1999).

[6] I. Pitas, Digital Image Processing Algorithms and Applications (Wiley-Interscience: 1° edition , February 4, 2000, 282–297).

[7] IUP, http://www.tecgraf.puc-rio.br/iup (last access: 01/07/2006).

[8] CD, http://www.tecgraf.puc-rio.br/cd (last access: 01/07/2006).

[9] IM, http://www.tecgraf.puc-rio.br/im (last access: 01/07/2006).

[10] TeCGraf/PUC-Rio, http://www.tecgraf.puc-rio.br (last access: 01/07/2006).

[11] G. Beylkin, R. Coifman, and V. Rokhlin, Fast wavelet transforms and numerical algorithms I. Communications on Pure and Applied Mathematics, 44:141–183, 1991.

[12] Brazilian Digital Satellite Images: http://www.cdbrasil.cnpm.embrapa.br/ (last access: 01/07/2006).