CEIG - Spanish Computer Graphics Conference (2012)
Isabel Navazo and Gustavo Patow (Editors)

WebGL-based visualization of voxelized brain models

J. Cruz de la Torre, J. Jiménez Ibáñez, A.M. López Estrella and J. Ruiz de Miras

Computer Science Department, University of Jaén, 23071 - Jaén, Spain

Abstract

This work presents our preliminary results on the interactive construction and visualization of voxelized brain models into Web platforms by using WebGL. Our voxel model is a non-usual level of detail (LOD) representation based on the box-counting algorithm used to calculate the fractal dimension of 3D brain images. Our results to date show a promising solution based on a client-server approach.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Image Generation—

1. Voxel model and solutions developed

The box-counting algorithm [RHO80] consists of counting how many pixels, higher than a threshold, of a 3D image (a stack of 2D MRI slices) are contained in a grid of voxels. Each of these voxels is labeled as black, gray or white depending on how many pixels it contains. This process is repeated for grids with incremental voxel size. So, our voxel model is an special LOD representation where passing from a level to other implies adding one pixel to the voxel size, what is not typical in voxel-based LOD representations. From a 3D image, we want to calculate and display, in a Web browser, the black, gray and white voxels for a voxel size and a threshold interactively selected. Initially, the server stores the 3D image to be represented. The first solution consists of sending to the client the original 3D grid formed by the set of MRI slices, and then all processing is done on the client. The voxels are modelled and rendered in the web browser using WebGL [Kg11]. Our second approach is based in a client-server strategy, where the server calculates the voxels for the LOD and the threshold selected, and sends it to the client. While the client is visualizing the model, other LODs are calculated on the server and transferred and stored in the client using second-plane AJAX calls. Thus, when the user selects another consecutive LOD, there is no request to the server due to this LOD has been received and saved previously.

2. Preliminary results and conclusions

We have tested our two solutions using MRI brain models of size 157 x 157 x 189 in two different scenarios, a LAN and an ADSL network. In each case, the LODs range in voxel size from 1 to 10 pixels (Figure 1 shows two LODs of voxel sizes 5 pixels and 1 pixel respectively). The client-server solution obtains the best times, achieving speedups of 2.37x and 1.76x against the client solution for each scenario. The client-server solution requires more network connections, but the processing is performed in background while the client renders the model. Once the first LOD has been transferred, changing between consecutive LODs is done in interactive frame rates. But if the threshold is updated, the client must wait until the initial LOD for the recalculated 3D grid is provided by the server. Both the computing speed of JavaScript and the size of the original 3D grid transferred make the client solution much slower. Our current work focuses on implementing GPU shaders in WebGL for optimizing the computation of the LODs. We are working in an alternative solution by performing a parallelization via JavaScript threads to exploit the multi-core client CPU.

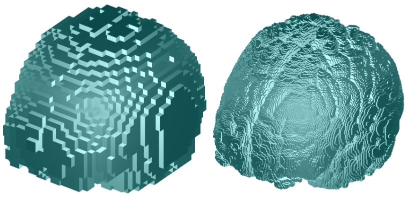

Figure 1: *Two LODs (voxel size 5 pixels and 1 pixel).*

3. Acknowledgements

This work has been partially supported by the University of Jaén, the Caja Rural de Jaén, the Andalusian Government and the European Union (via ERDF funds) through the research projects UJA2009/13/04 and PI10-TIC-5807.

References

[Kg11] KHRONOS-GROUP: WebGL specification v. 1.0, 2011. http://https://www.khronos.org/registry/webgl/specs/1.0. 1

[RHO80] RUSSEL D., HANSON J., OTT E.: Dimension of strange attractors. *Physical Review Letters 45* (1980), 1175–1178. 1

DOI: 10.2312/LocalChapterEvents/CEIG/CEIG12/151-151