

# HARDWARE-ACCELERATED DISTORTION FOR VOLUME VISUALIZATION IN MEDICINE

Marcelo Cohen  
School of Computing  
University of Leeds  
marcelo@comp.leeds.ac.uk

Ken Brodlić  
School of Computing  
University of Leeds  
kwb@comp.leeds.ac.uk

Nick Phillips  
Leeds General Infirmary  
Nick.Phillips@leedsth.nhs.uk

**Abstract**—Medical imaging applications frequently deal with large datasets, usually three dimensional data extracted from a CT or MRI scanner. Volume visualization techniques provide an effective way to visualize these data but sometimes there is too much to see at once on the display. Thus ideas from the Information Visualization field can be borrowed to help achieve a more effective understanding of that data, particularly the notion of focus and context based on spatial distortion. In order to obtain an acceptable performance, it is possible to use the graphics processor (GPU) to accelerate the distortion itself.

## I. INTRODUCTION

Volume visualization is a well known technique for assisting the diagnosis of a number of diseases, by producing a 3D view of the results of medical scans. For example, in a neurological application one may want to visualize an aneurysm which is a condition that occurs when the tissue of a blood vessel is weak at a certain point and then suffers a dilation. This can be a very dangerous situation and even cause death if the aneurysm ruptures - hence the need to identify this problem as soon as possible. But an aneurysm can show up in a number of sizes, and sometimes can be hard to identify amongst the vessel network of the brain. It is quite important to be able to see this network in its entirety, but sometimes it is just impossible to visualize both the aneurysm and the vessels with enough detail.

Thus this research has been motivated by the need to view brain aneurysms with good quality but at the same time still providing the user with a full view of the surrounding arteries. Ideally this should be done as quickly as possible - so another goal was to provide the surgeon with a tool capable of producing a 3D image nearly in real time. This paper presents the ideas which led to the implementation of such a prototype and also reports some results.

## II. APPROACH

### A. Volume Visualization

Our sample dataset is taken from CTA (CT Angiography), where contrast has been injected into the patient to make the arteries opaque to x-ray. The scanner produces a number of slices, forming a 3D volume of data. We model the data as a “material” whose colour and opacity at any point reflect the value of the data. Modern graphics hardware supports the technique of 3D texture mapping whereby an image is created

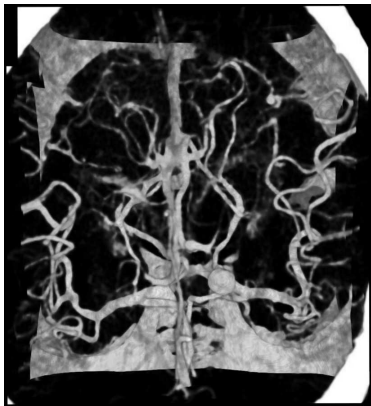
by compositing slices through the volume at right angles to the viewer. By changing the colour and opacity associated with data values, we gain flexibility in the final image - for example, data values corresponding to the presence of arteries can be highlighted.

### B. Focus and Context

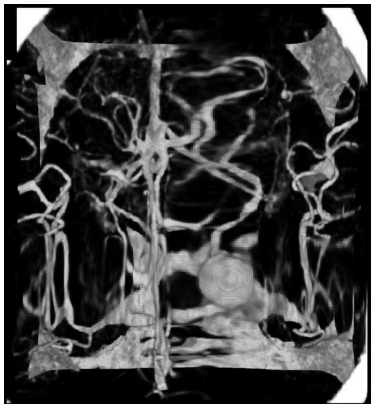
The idea of *focus and context* comes from the field of Information Visualization (InfoVis)[1]: usually it is possible to identify a region of interest in any dataset (which we call the *focus region*). But sometimes the dataset is so large that it cannot be displayed in its entirety on the computer display, therefore one may show just the focus region and hide the remaining. This is the traditional “zoom in” operation, present in a number of implementations. But this actually removes information from the display: everything outside the focus region (the so called *context region*).

Therefore the InfoVis field has introduced a number of approaches to solve this problem - usually through the use of some kind of spatial distortion, like enlarging the focus region and compressing the context. For instance, a widely known method proposed by Spence [1] is the bifocal display. Although these techniques are usually applied to non-numerical data, interesting results have already been obtained with our 3D medical data set [2]: imagine the region around the aneurysm as the *focus* and the network of arteries as the *context*. In a conventional 2D bifocal display, there are nine regions - the focus region and eight neighbouring context regions: left, right, above, below and the four remaining diagonal ones.

Likewise, the same method can be applied to the entire volume, now considering that the focus region will be a cuboid instead of a rectangle. In this way, there will be twenty seven regions within the volume ( $3^3$ ). Results can be seen in figure 1: the aneurysm is magnified while the vessel network remains clearly visible, although slightly compressed. This method produced a good result, but it is far from interactive: as the distortion process itself is computationally intensive, the effectiveness of the system is reduced. Hence there was the need to research on faster methods, such as GPU programming.



(a) normal



(b) magnified

Fig. 1. Bifocal distortion on a 3D dataset

### C. GPU programmability

Modern graphics processors are partially programmable, which means that some fixed sections of the graphics pipeline can be changed by the user. In the past GPUs were programmable only through assembly language, largely restricting their widespread use. But now a number of higher level languages allow simpler and portable programming such as Cg, OpenGL Shading Language and HLSL. GPU programming requires the creation of pieces of code called *shaders*. For example, it is possible to change the opacity of each pixel, or change the way textures are used - this is done through a *fragment shader*. The important thing to understand is that a fragment shader affects **every pixel** being drawn (while active), regardless of its origin.

### III. GPU DISTORTION EXPERIMENTS

We now describe some experiments we have carried out, to investigate the potential of using the GPU to compute the

distortion. The test dataset was 256x256x128 slices, and the experiments were run on a GeForce FX2000 graphics card.

In the work described in II-B, the focus and context effect was generated by a transformation of the dataset into a new 3D volume to which texture mapping was applied (see [2] for details of the transformation). Thus the CPU of the PC was used for the transformation, and as we noted in II-B this is computationally expensive. However, we can intercept the texture coordinates being used to display a pixel by the GPU, and carefully apply the distortion effect on the graphics card, by using a fragment shader to change the texture coordinates of each voxel according to the region (focus or context) that it is in.

This is a significant improvement over the previous method, which required substantial changes to the volume rendering code. Not having to regenerate the textures, the GPU method also allows the distortion parameters to be changed interactively. Finally, another benefit is the possibility of using different distortion methods in the future, just by replacing the fragment shader. We note, however, that the frame rate rapidly decreases as the size of the display window increases (from 50 frames per second on a 100x100 window to less than 5 frames per second on a 500x500 window). This happens because to generate a 3D view, a number of semi-transparent slices must be drawn on top of each other and the fragment shader has to compute the new texture coordinates of every single pixel drawn on screen - which in itself is a costly procedure.

### IV. CONCLUSIONS AND FUTURE WORK

The GPU distortion method is an elegant method, as it removes the complexity from the volume renderer and also allows for a more generic implementation of the distortion itself. Although it is faster than the previous CPU-based method, the performance is still not optimal. To accelerate the process, it is possible to encode the distortion as a secondary texture and just apply it when displaying the volume. This could be achieved by encoding the difference between the original texture coordinates and the distorted ones. We are currently exploring this idea, and more generally, we are developing a generic model of focus and context for volume visualization.

To serve as a testbed for these experiments, a prototype has been developed that allows easy manipulation of distortion parameters, as well as simplified navigation through slices and in 3D space. The prototype still has to be evaluated and we are interested particularly in the user interface issues and in the applicability of the distortion method itself.

This research project is being accomplished as part of a PhD research programme sponsored by the Brazilian Research Agency CAPES.

### REFERENCES

- [1] R. Spence, *Information Visualization*. ACM Press, 1st ed., 2001.
- [2] M. Cohen and K. Brodlie, "Focus and context for volume visualization," in *Theory and Practice of Computer Graphics* (I. C. S. Press, ed.), pp. 32-39, 2004.