DIGITALLY RECONSTRUCTED RADIOGRAPHS USING THE GRAPHICS HARDWARE

D. Ruijters[1,3], B. M. ter Haar Romeny[2], P. Suetens[3]

[1]Philips Medical Systems, X-Ray Predevelopment, the Netherlands

[2]Technische Universiteit Eindhoven, Biomedical Engineering, Image Analysis and Interpretation, the Netherlands

[3]Katholieke Universiteit Leuven, Medical Image Computing, ESAT/Radiologie, Belgium

Abstract

A Digitally Reconstructed Radiograph (DRR) is an X-ray -like representation of a CT dataset. Since the Hounsfield values in a CT dataset are obtained using X-ray radiation, it is possible to reconstruct such a DRR, from any arbitrary viewing angle, including angles that were not included in the original CT scans. DRRs are being used in many intensity based 2D-3D registration algorithms, as well as in the planning phase of radiotherapy. We present an approach that is both very fast and accurate, harvesting the parallel processing power of today's mainstream graphics hardware. Especially we use the high precision z-buffer as intermediate buffer, in order to produce accurate results.

1 Introduction

The intensities on the X-ray image are the result of the X-ray absorption along the paths of the rays:

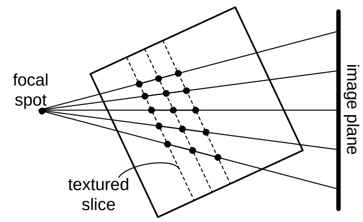
$$I = I_0 \cdot e^{-\int \mu(x)dx} \qquad (1)$$

Whereby I and I_0 are the output and input X-ray intensity and $\mu(x)$ expresses the attenuation coefficient along the path of the ray. A DRR is based on casting virtual rays through the CT volume, and integrating the encountered Hounsfield values, which represent the X-ray opacity per volume element.

The evaluation of a line integral, along the path of a given ray through the discrete grid of the CT dataset, can be expressed as:

$$L = \sum_i \sum_j \sum_k l(i,j,k) \cdot p(i,j,k) \qquad (2)$$

Whereby $p(i, j, k)$ is the intensity of the voxel at index i, j, k, and $l(i, j, k)$ is the contribution of that particular voxel to the given ray. $l(i, j, k)$ is determined by the path of the ray and the interpolation scheme. The brute force execution of equation 2 would be rather inefficient, since for most interpolation schemes, such as tri-linear interpolation (which we use), $l(i, j, k)$ is zero for the vast majority of the voxels.



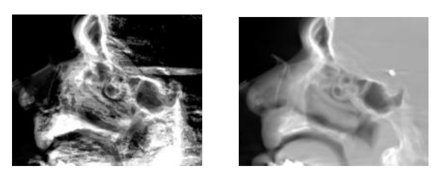
2 Method

We employ the Graphics Processing Unit (GPU) to accelerate the calculation of the DRR. Our method is based on the usage of textured slices, an approach that has been developed for Volume Rendering (VR) applications [1]. There are several complicating factors, compared to VR. The most important are:

- There is no straight-forward way to implement a summation (see equation 2) on the GPU.
- The accuracy of the buffer that holds the intermediate results has to be sufficient.

We overcame these issues by using a pixel shader that performs the summation, and writes its result to the 32-bit z-buffer, rather than the frame buffer.

3 Results



The left image shows a DRR that was obtained by using the standard 8-bit frame buffer, while in the right image the same DRR was rendered, using the 32-bit z-buffer.

The generation of a DRR, using our method, on a 512^2 image grid of a 32 MB CT dataset takes 29 ms on an nVidia QuadroFX 3500 graphics card with 256 MB on board memory.

References

[1] Ruijters, D., and Vilanova, A., "Optimizing GPU Volume Rendering," Journal of WSCG'06, 14(1-3), 9–16, 2006.