On the Design of a Real-Time Volume Rendering Engine

J. Smit, H.J. Wessels, A. van der Horst and M.J. Bentum

ABSTRACT
An architecture for a Real-Time Volume Rendering Engine is given capable of computing 750x750x512 samples from a 3D dataset at a rate of 25 images per second. The RT-VRE uses for this purpose 64 dedicated rendering chips, cooperating with 16 RISC-processors. An plane interpolator circuit and a composition circuit, both capable to operate at very high speeds, have been designed for a 1.6 micron VLSI process. The interpolator is now back from production. It has been tested an complied with our specifications.

1.1 Introduction

The visualization of high resolution images acquired from 3D datasets in real-time is extremely important in every day applications. This is especially true for medical applications where large 3D datasets are acquired in a relatively short time using scanners based on X-ray imaging (CT equipment), Magnetic Resonance Imaging (MRI equipment), Ultrasound imaging etc.

The diagnostic value of the various scanners can be greatly increased if the 3D data can be interpreted and viewed in real-time. This assumes the classification of the dataset, as well as the final visualization of it. It should be observed however that the classification needs to be performed only once. The visualization on the contrary, should be performed at a rate of at least 25 images per second, at such high resolutions as 750* 750*512 points in display space. The real-time speed of 25 images per second is highly useful as only a fraction of the relevant data may be visible at an arbitrary initial setting of visualization parameters, due to the hidden surface removal in the composition algorithm. The subsequent process of the selection of optimal observation parameters, such as tissue opacity, visualization angle, lightsource position etc. takes an intolerable amount of time on current workstation implementations of 3D visualization software with single frame interaction rates in the order of minutes to seconds.

High speed, full resolution imaging gives the additional advantage that measurement data, which normally vanish in noise like 1D structures from small blood-vessels, can be observed by slowly rotating a 3D scene with optimal visualization parameters, as the eye is very sensitive in the recognition of correlated 'noisy' paths in 3D scenes.

The real-time visualization engine described in this paper is based on volume rendering, i.e. it computes lightintensity values along rays, using the composition formula :

$$\begin{array}{rcl} alpha & : & \alpha_{int}(I,J,K).(1-\alpha(I.J)) \\ \alpha(I.J) & := & \alpha(I,J)+alpha \\ C(I,J) & := & C(I,J)+C_{int}(I,J,K).alpha \end{array} \qquad (1.1)$$

Volume visualization has the advantage of being the most complete visualization method [Levo88],[Levo90a],[Levo90b]. The fact that there is no need to extract a specific surface has as advantage that no true interpretation of the 3D data is needed before actual visualization, thereby avoiding the problems with the partial volume effect. The reader is referred to [KaBa90] for a survey about the various techniques for 3D visualization.

1.2 Volume Visualization Hardware

Many volume rendering implementations have been realized on general purpose computers, or on general purpose computers combined with computer graphics hardware. [Levo88] discusses the performance of the volume rendering algorithm. Most of the systems designed so far degrade the image quality in order to obtain real-time performance. For instance the implementations discussed in [KaBa90] are not designed for the volume rendering algorithm. These machines are not capable of rendering semi transparent surfaces and produce images of inferior quality in real-time. The "Voxel Processor Prototype" for instance [GoRe87] is capable of rendering approximately 20 images/sec. However, this machine too does not perform volume rendering with the composition formula 1.2. Instead an alternative approach is used in which no subsampling to true display coordinates is used. The image is generated just by addressing the voxels in a back-to-front order, overwriting the hidden voxels, without a composition step. This way of rendering is more like binary voxel rendering, resulting in images with an arbitrary discreteness, in which individual voxels became visible like 'sugar cubes'. A sub real-time, true volume visualization running on the fast general purpose graphics engine, Pixel-Planes 5, is described in [Levo89]. The hardware of the Pixel-Planes 5 implementation [FuPo89], which includes a 640 MByte/s ring network and dedicated RAM with built-in graphics primitives, is so considerable that its realization can be expected to be expensive. The usage of low resolution images during image rotations was needed in this approach to bring real-time volume visualization within reach.

1.3 The design of a Real-Time Volume Rendering Engine (RT-VRE)

Extensive studies of the volume visualization algorithm reveal that the percentages of time spent in the straightforward implementation of the volume rendering algorithm are divide as follows :

$$\begin{aligned} Interpolation &: 91.34\% \\ Composition &: 5.37\% \\ Geometry &: 3.29\% \end{aligned} \tag{1.2}$$

This gives an indication that a speed-up for the interpolation algorithm is most wanted. It can be shown however that the task of generation of 750*750 images, sampled at 512 depth positions, at a rate of 25 images per second is equivalent to the desire to construct a real-time volume rendering engine capable of executing 600 Giga operations per second, using the straightforward algorithm. A study about the amount of power required to execute this straightforward implementation of the algorithm shows that between 10 and 20 kilo-watts are required to implement the algorithm at the given performance level, provided that it could be realized with chips of the current generation, even if dedicated ASICs are used at strategic places.

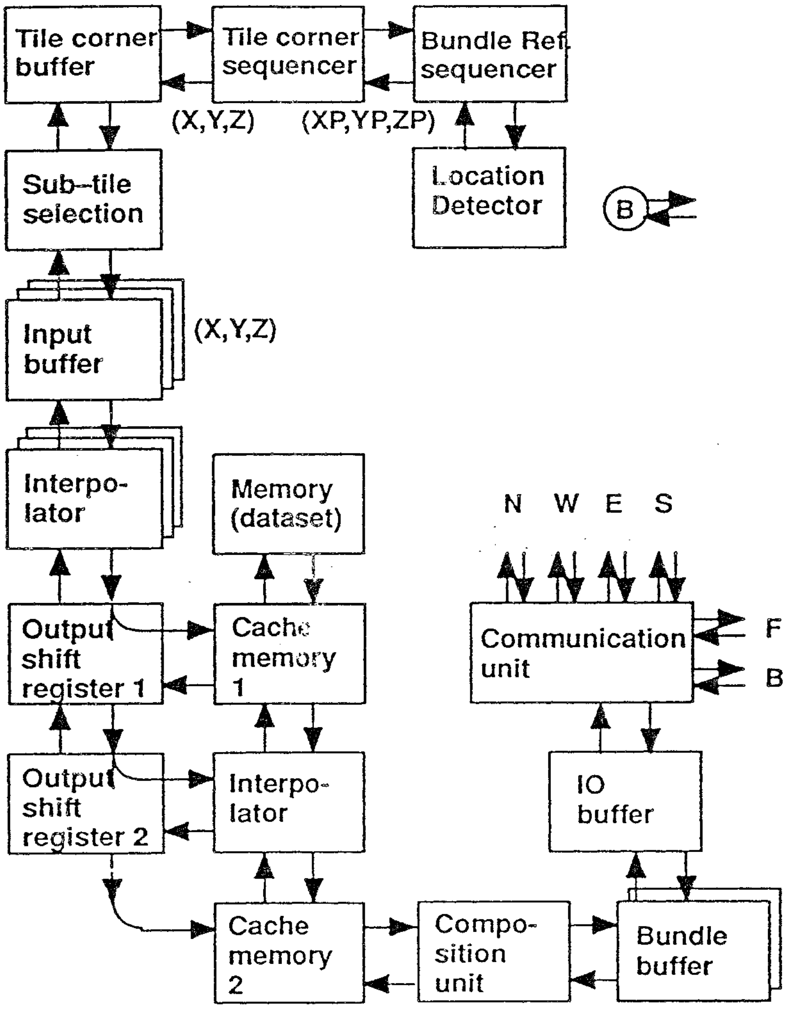

FIGURE 1.1. The VRE Processing Element

The outcome of this study motivated us to start a chip-design, with minimal power dissipation and maximal performance as main objectives, resulting in various novel VLSI building blocks for the visualization task, combining compact layouts, extremely low power dissipation, and unprecedented computational capabilities. Figure 1 shows a block diagram of the typical embodiment of an RT-VRE chip.

The engine processes complete bundels of 3D image data, resulting in 2D patches of display data. The starting point for such a bundle is sent to the VRE Processing Element though a broadcast interface (B) in the form of an origin and a set of increments, used to sequence through the 3D space along the rays. The bundle reference point sequencer just computes a single point in 3D space, which is a reference point for a cut-plane of the bundle. This plane is subdivided into smaller units, called tiles, which are sequenced by the tile corner sequencer. This sequencer puts the calculated addresses in a dedicated tile corner buffer. The sub-tile selection unit selects four tile corner points from the tile corner buffer and loads their X,Y,Z components in parallel into the plane-interpolator for the calculation of local addresses within the bundle plane. The calculated values are shifted out of the output registers of the interpolator, while triggering a cache memory connected to the main memory which contains the voxel values in the form of opacity and colors as well as the tissue types (i.e. the dataset). Any value missing in the cache is loaded from the dataset. The cache memory #1 is of a very special construction. It will cache 8 tuples of voxel data elements, which can be loaded in parallel into the interpolator #2. The output shift register #2 is used to allow some time for the interpolator #2 to calculate all the values of opacity and color within a given plane using a plane interpolator. Repetition of this process gives full tri-linear interpolation within the region of interest. The results of this step are stored in cache memory #2. The addresses produced by the output shift register #2 are used to select the desired values of opacity and color from the cache memory #2, which are fed into the composition unit, which operates on all points within the bundle.

A good feeling for performance level of the ASIC can be obtained if one takes into consideration that a bit addition takes 2 ns. in the 1.6 micron process used. The ASIC executes nevertheless at a 100 MHz composition rate. The DRAM bandwidth is fully saturated, using 40 ns cycles whenever possible and 150 ns cycles when new rows should be selected.

The RT-VRE architecture is capable to calculate composition operations at 100 Mega operations per second, using 4 composition units each measuring 1x2 mm in the 1.6 micron VLSI process. A total of 64 VRE-ASICs are needed to calculate 750*750 images at a rate of 25 images per second, using 512 samples in the depth of a 256*256*256 dataset. Prototypes of the plane interpolator and the 4-way composition unit are currently being processed. An 1 micron process will reduce the power dissipated by each ASIC to about 1 Watt.

The 64 RT-VRE processing elements (PE's) are part of an inhomogeneous multiprocessing network, as shown in figure 2. A total of 4 PE's is connected, together with a general purpose RISC processor, using bi-directional bus-couplers to a local interprocessor bus. Sixteen of such unit build-up the complete RT-VRE, using a regular 2D interconnect pattern. VRAM memories are used at this level to provide high bandwidth, unattended interprocessor communication.

The use of general purpose RISC processors makes the overall RT-VRE design very flexible, as compute intensive operations, like MRI classifications, can be performed on the same hardware with excellent speed.

Figure 3 shows one of the 16 boards to be used in the final RT-VRE prototype. The overall design will dissipate as little as 370 Watt, 160 Watt for the RAMs, 128 Watt for the 64 ASICs, 32 Watt for the RISC processors and 50 Watt for the service processors and peropherals. One ASIC co-processing unit, comprises:

1. One VRE-ASIC
2. A set of bidirectional bus-buffers
3. Either 4 pieces of 256Kx4 DRAM, or one 256Kx16 DRAM

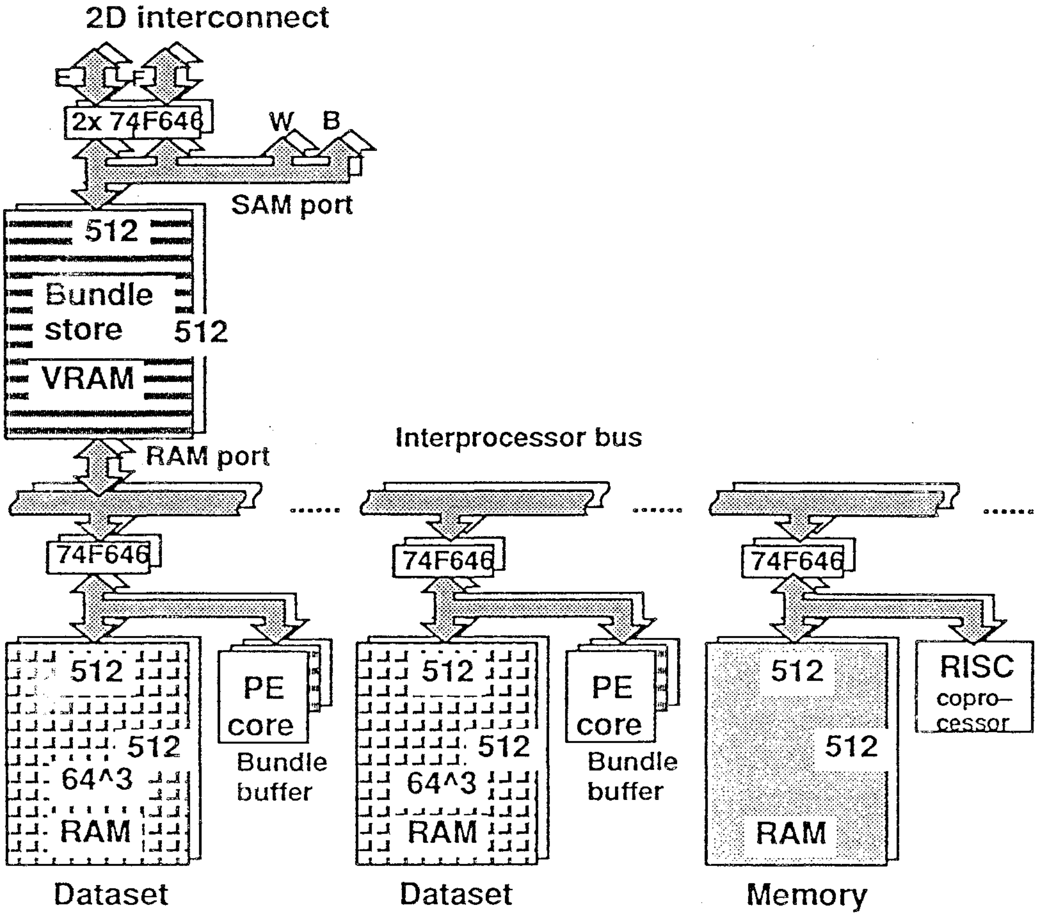

FIGURE 1.2. The local VRE interconnect structure

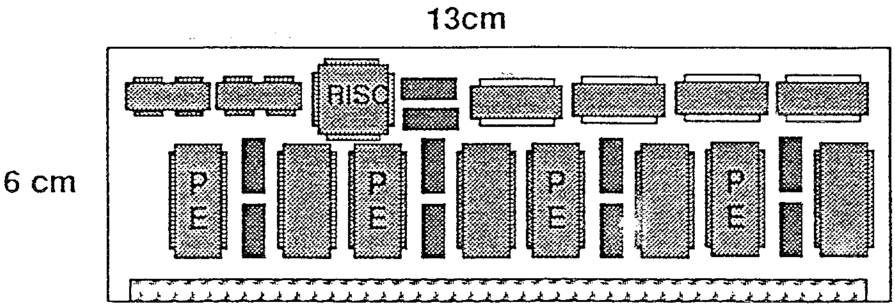

FIGURE 1.3. Board level realization of the VRE

1.4 Additional Functionality

The ASICs provide additional functionality to the RT-VRE through the inclusion of support for other visualization tasks, like:

- Enhanced composition, giving a mix of normal compositions and maximum (minimum) intensity projection.
- Cast-shadow calculation.
- Fast classification.

The inclusion of additional DRAM makes it possible to scroll through larger datasets as those indicated. The system can be software reconfigured from a 256x256x256 dataset resolution, to a 512x512x64 dataset resolution.

1.5 Conclusions

Visualization, of 3D (medical) datasets sampled at a resolution of 750x750x512, is not economical with current CPU technologies and general purpose techniques, due to execessive power dissipation. We have shown a solution for the 3D visualisation problem using dedicated ASICs. The new architecture described in this paper requires as little as 370 Watt. This makes a Real-Time 3D Visualisation Workstation a feasible unit which can be used at arbitrary (clinical)locations.

Acknowledgements:

The final realization of the Plane Interpolator has been done by Maarten de Mönnink and René Oogink. The plane interpolator chip was tested by Hans Snijders. The Composition Unit was realized by Ronald Peer, Waldo Hazeleger and Edwin aan de Stegge. The chips were processed by Eurochip and the Dutch Pico organization. The realization of the chips was a cooperative effeord between the Hogeschool Enschede and the University of Twente.

1.6 References

[FuPo89] H. Fuchs, J. Poulton, J. Eyles, T. Greer, J. Goldfeather, D. Ellsworth, S. Molnar, G. Turk, B. Tebbs, L. Israel, "Pixel Planes 5: A heterogeneous Multiprocessor Graphics System using Processor-Enhanched Memories," in Computer Graphics, Vol. 23, No. 3, July 1989, pp. 79-88.

[GoRe87] S.M. Goldwasser, R.A. Reynolds, "Real-Time Display and Manipulation of 3D Medical Objects: The Voxel Processor Architecture," in Computer Vision, Graphics, and Image Processing, Vol. 39, 1987, pp. 1-27.

[KaBa90] A. Kaufman, R. Bakalash, D. Cohen, R. Yagel, "A Survey of Architectures for Volume Rendering," in IEEE Engineering in Medicine and Biology, December 1990, pp. 18-23.

[Levo88] M. Levoy, "Display of Surfaces from Volume Data," in IEEE Computer Graphics and Applications, May 1988, pp. 29-37.

[Levo89] M. Levoy, "Design for a Real-Time High-Quality Volume Rendering Workstation." in Proceedings of the Chapel Hill Workshop on Volume Visualization, May 1989, pp. 85-92.

[Levo90a] M. Levoy, "Volume Rendering by Adaptive Refinement," in The Visual Computer, Vol. 6, No. 1, 1990, pp. 2-7.

[Levo90b] M. Levoy, "Efficient Ray Tracing of Volume Data," in ACM Transactions on Graphics, Vol. 9, No. 3, July 1990, pp. 245-261.