Direct Processing of Compressed Volume Data

Diploma Thesis • Thomas Höllt

Computer Graphics Working Group, Institute for Computational Visualistics
University of Koblenz-Landau, Campus Koblenz, Germany

UNIVERSITÄT KOBLENZ · LANDAU

COMPUTERVISUALISTIK

Motivation

The advancing research and development of **image acqusition** in medical imaging (CT, MRI) as well as increasing accuracy in **simulations** leads not only to higher visual quality and more detailed datasets but also a to a **vast increase in the amount of data**.

Today, graphics hardware (GPU) is used for many **visualization** and **processing** tasks[1].

Thanks to the tremendous growth of raw computing power, the gap between traditional **bottlenecks** (limited amount of directly accessible memory, low memory bandwidth between CPU and GPU) and the high computation performance is increasing.

During this thesis, these problems should be addressed by extending the volume processing and rendering framework **"Cascada"** focusing on the following aspects:

- Compression of volume data
- Direct visualization of the compressed data on the GPU
- Processing the data directly in the compressed representation on (CPU and GPU)

Compression

- **Haar wavelet transform**[2] is fast to compute and permits relatively easy access to random elements
- Compute **averages** over 2x2x2 Voxel Blocks
- Store **differences** in addition to the averages to preserve detail
- Most of the differences will be zero or at least very small
- Store only differences bigger than a certain threshold (zero for lossless) to achieve compression

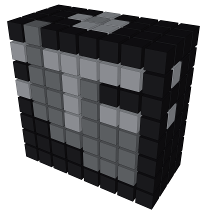

Fig. 1 A Volume Dataset

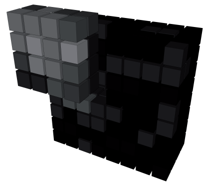

Fig. 2 Wavelet Decomposition

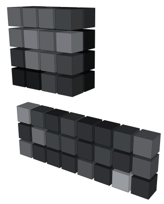

Fig. 3 Zero-coefficients discarded

Visualization & Processing

- Visualization[3] requires the lookup of the average value plus seven coefficients for every fragment and needs seven extra additions compared to non-compressed visualization
- Point operations work either directly on the averages and / or coefficients or require at worst composition and decomposition of the current 2x2x2 block.
  When processing the volume blockwise (instead of per voxel) this means only 14 additional add operations per eight voxels
- Local operations need composition of the required neighborhood (up to 152 additional fetches and 432 additional add operations per eight voxels)
- Operations requiring wavelet composition demand a complete reordering of the coefficients.

Results

Operation	CPU time [ms]	GPU time [ms]	CPU compressed @ 1:2 time [ms]	CPU compressed @ 1:4 time [ms]	GPU compressed @ 1:2 time [ms]	GPU compressed @ 1:4 time [ms]
Invert	855	28.4	151	108.1	283.9	169.3
Scale brightness	872	27.6	151	101.4	280.5	167.1
Binary threshold	561	37.9	1850	1794	3396	2408

Processing times of a 512 x 512 x 128 voxel dataset on an eight core intel Xeon system at 2.8Ghz and a GeForce 8800GT with 512MB of VRAM

- Especially the CPU modes for compressed data benefit from the fewer operations due to discarded coefficients, when no decomposition is needed.
- If composition is needed, however, performance will drop significantly
- GPU modes for compressed data are slow due to big overhead needed for preparing texture data.

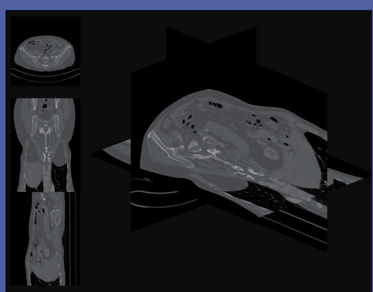

Fig 4 Visualization of a 512x512x256 dataset directly from compressed data in Cascada. Compression rate is 1:4. Rendering at interactive framerates

- For the future, the main focus is on a more complete set of operations including filtering, and more advanced render-modes such as DVR

References

[1] Owens J.D., Luebke D., Govindaraju N., Harris M., Krueger J., Lefohn A.E., Purcell T.: *A Survey of General-Purpose Computation on Graphics Hardware*, Computer Graphics Forum, 2007; 26(1); pp. 80-113

[2] Mulcahy C.: *Image Compression Using the Haar Wavelet Transform*, Spelman Science and Mathematics Journal, 1997; 1(1); pp. 22–31 .

[3] Schneider J., Westermann R.: *Compression domain volume rendering*, VIS '03: Proceedings of the 14th IEEE Visualization, 2003, p. 39

cascada