International Conference on Artificial Reality and Telexistence
Eurographics Symposium on Virtual Environments (2014)
T. Nojima, D. Reiners, and O. Staadt (Editors)

Adding Interlacing to In-Situ Data Compression for Multi-Resolution Visualization of Large-Scale Scientific Simulations

H. Lehmann, M. Lenk and B. Jung

Virtual Reality and Multimedia Group, Technical University Freiberg, Germany

Abstract

Large-scale simulations in HPC environments produce massive data sets in the order of several terabytes which pose challenges both to data storage and data analysis. A promising approach for reducing the amount of data written out during simulation runs is in-situ compression. However, even compressed data sets are typically still much too large for interactive visual data exploration. We propose an interlacing scheme applicable to in-situ compression in order to write out results of HPC simulations in a compressed multi-resolution format. This data format allows progressive loading of large scientific data sets in high-end immersive displays. Low-resolution views are easily extracted for visualizations on low-end displays such as hand-held devices and web browsers.

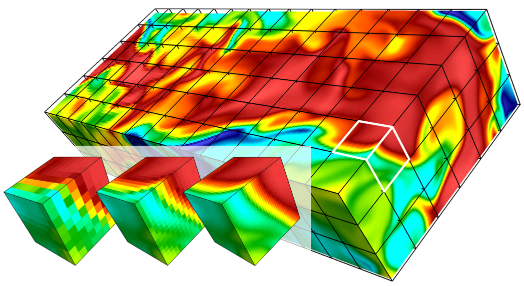

Figure 1: *Top: The voxel grid is subdivided into AMR blocks. Each block is processed with the in-situ compressor in the HPC environment and can be restored at multiple LOR. Bottom: E.g., for three LOR, AMR blocks can be restored at* 8^3, 16^3 *and* 32^3 *voxels (full resolution). Resolution of the data shown increases from left to right.*

Today's HPC maintain a steady increase in FLOPS while lacking improvements in I/O capabilities. Data movement from main memory to mass storage has become a bottleneck that forces scientists to subsample data or write data infrequently defeating the purpose of high-resolution simulations. In-situ approaches reduce the amount of data to be written out benefiting from data residing in main memory. *In-situ compression* supports the reconstruction of the full data set and analysis as post-processing step while reducing data size to ~ 15-20% [LSE*11].

We apply an interlacing scheme to two in-situ block-based compressors - ISABELA (*In-situ Sort-And-B-spline Error-bounded Lossy Abatement*) [LSE*11] and SBD1 (*Set Based Decomposition*) [IKK12] and show the resulting compression rate (CR, *compressed/raw* data size). We describe the integration into the ParaView/VTK toolkit and use it to generate views for different low- and high-end platforms, such as handheld devices, web browsers and immersive VR.

Figure 1 shows the in-situ *Adaptive Multi Resolution* (AMR) compression approach: Each compute process subdivides its local simulation grid into distinct AMR blocks. Each AMR block can be restored at several levels of resolution (LOR). Subdivision of AMR blocks is accomplished by 3D interlacing similar to Adam7 used in 2D *Portable Network Graphics* shown Figure 2 *right*. Voxels inside AMR blocks are reordered, so that lower LOR voxels precede voxels of higher LOR. At the end reordered voxels are grouped in 8^3 compression blocks and fed into an in-situ block-based compressor.

Data easily can be restored at multiple LOR by only decompressing a continuous subset of compression blocks (LOR 1≙100%, 2≙12.5%, 3≙6.125% ...). Refinement doubles the number of voxels in each dimension, so the method

DOI: 10.2312/ve.20141370

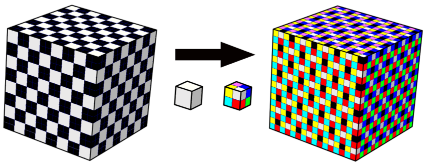

Figure 2: *Refining LOR from* 8^3 *(black/white) voxels to* 16^3 *(black/white+colored) voxels. Black/white voxels and voxels of one color are stored in seperate compression blocks containig* 8^3 *voxels each.*

is comparable to spanning an octree. However, higher LOR completely contain lower LOR voxels, avoiding storage requirements for average values. Progressive loading can be accomplished by successively placing decompressed voxels in an already allocated grid.

The impact of the additional interlacing step on the CR of ISABELA and SDB1 was tested with three different CFD data sets (`melt`, `isot`, `comb`). ISABELA relies on linearizing, sorting, B-spline fit and error correction. SDB1 relies on scalar quantization replacing the original data values with indices referring to a look-up table. ISABELA is run with a temporal compression extension (i.e. every third time step is a reference time step). SDB1 was extended to store one look-up table per AMR block and look-up indices inside compression blocks making it AMR capable. Both compressors are configured for a maximum point-wise relative error of 1%. The proposed scheme provides simple decompression of AMR blocks at multiple LOR, at the expense of lower CR due to reduced spatial coherence of voxels in compression blocks. Table 1 shows the CR with and without AMR. It can be seen, that the additional interlacing yields a moderate increase of CR by $\sim 0.5 \ldots 5.0\%$.

Table 1: *CR with [and without] AMR of block-based in-situ compressors ISABELA and SBD1 for* `double` *precision CFD data: (*`melt`*) 164 time steps Metal Melt Flow in Porous Media (116.76GB,* $288 \times 288 \times 288$*); (*`isot`*) 1024 time steps Isotropic Turbulence [LPW*08] (6GB,* $64 \times 64 \times 64$*); and (*`comb`*) 122 time steps Turbulent Combustion [YCS07] (147.44GB,* $480 \times 704 \times 96$*) with various scalars* $(mixf, vort, y\_oh, chi, hr)$*.*

Variable	ISABELA		SDB1	
`melt`	15.4%	[17.8%]	14.1%	[18.5%]
`isot`	16.6%	[18.2%]	14.0%	[18.0%]
`comb`				
mixf	12.2%	[14.7%]	4.7%	[7.5%]
vort	19.7%	[23.1%]	12.6%	[17.8%]
y_oh	12.4%	[14.8%]	15.2%	[19.4%]
chi	15.7%	[18.9%]	19.4%	[21.1%]
hr	15.4%	[20.3%]	12.3%	[16.5%]

Our AMR compression is interoperable with AMR and Multiblock formats [ZLG11] of the ParaView/VTK toolkit through plugins. Global low- or local high-resolution views for mobile devices, web browsers, desktops and CAVEs as shown in Fiugre 3 can be generated and LOR can be configured spatially through points of interest.

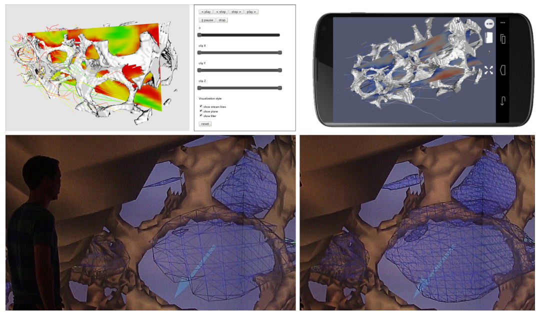

Figure 3: *AMR views of decompressed data. Top left: web browser via X3DOM, Top right: hand-held device via KiwiViewer, Bottom: CAVE via ParaView/VTK.*

In conclusion, in-situ AMR compression is a promising approach for coping with the I/O bottleneck in both HPC and realtime visualization environments. Our interlacing scheme works with block-based in-situ compressors as well as AMR and Multiblock file formats of ParaView/VTK. While inducing only a minimal overhead in terms of compression rate, it enables interactive, progressively refinable visualizations of large-scale scientific simulations.

Acknowledgements This research is supported by the Deutsche Forschungsgemeinschaft (DFG) within SFB 920.

References

[IKK12] Iverson J., Kamath C., Karypis G.: Fast and effective lossy compression algorithms for scientific datasets. In *Proceedings of the 18th international conference on Parallel Processing* (Berlin, Heidelberg, 2012), Euro-Par'12, Springer-Verlag, pp. 843–856. 1

[LPW*08] Li Y., Perlman E., Wan M., Yang Y., Meneveau C., Burns R., Chen S., Szalay A., Eyink G.: A public turbulence database cluster and applications to study lagrangian evolution of velocity increments in turbulence. *Journal of Turbulence 9* (2008). 2

[LSE*11] Lakshminarasimhan S., Shah N., Ethier S., Klasky S., Latham R., Ross R. B., Samatova N. F.: Compressing the incompressible with ISABELA: In-situ reduction of spatio-temporal data. In *Euro-Par (1)* (2011), Jeannot E., Namyst R., Roman J., (Eds.), vol. 6852 of *Lecture Notes in Computer Science*, Springer, pp. 366–379. 1

[YCS07] Yoo C., Chen J., Sankaran R.: Direct numerical simulation of turbulent lifted hydrogen/air jet flames in heated coflow. In *2nd ECCOMAS Thematic Conference on Computational Combustion in Delft, the Netherlands* (2007). 2

[ZLG11] Zagaris G., Law C., Geveci B.: Visualization and analysis of AMR datasets. *Kitware Source 18* (07 2011), 2–6. 2