A Multiresolution Approach to Large Data Visualization

John Clyne[1]

Abstract

Numerical simulations of turbulent flow, which are fundamental to the study of weather, climate, oceanography, astrophysics, and related disciplines, are yielding data sets with temporal and spatial resolutions that cannot be effectively studied with today's analysis tools. For example, present day volume visualization applications available to NCAR scientists are capable of supporting interactive exploration of 128^3 time varying volumes or perhaps 256^3 static volumes. Numerous groups at NCAR and other institutions, however, are routinely generating time evolving datasets with spatial resolutions on the order of 512^3. At least one group at NCAR has plans for a 1024^3 simulation within the year. This white paper discusses a strategy for addressing interactive visual exploration of these large data based on multiresolution data representation. Preliminary results are presented from a prototype volume rendering tool that has been extended to support interactive direct volume rendering of large data by exploiting multiresolution methods.

Introduction

Current visual analysis tools are not adequate for the data sets generated by the numerical simulations of many of today's turbulence researchers. As the computing power available to researchers has increased, so has the resolution of their data sets. Unfortunately, not all aspects of computing (e.g. networking, I/O, graphics, memory) have scaled at the same rates as floating point performance and consequently analysis tools have not kept pace with the ability to generate larger and larger data sets. Turbulence data sets with spatial volumes on the order of 512^3 made up of hundreds or even thousands of time steps are commonplace at NCAR today. Plans of at least one research group call for a 1024^3 simulation later this year [1]. While efforts at some institutions have produced direct volume rendering [2] and iso-surfacing [3] tools capable of interacting with data sets on this scale, these tools run on specialized computing platforms costing millions of dollars. Isosurfacing and direct volume rendering tools available on affordable computing platforms, and presently used by NCAR researchers, may effectively handle, at best, static 256^3 data sets or 128^3 time varying data.

The large data visualization techniques developed at Utah [3] and LANL [2] are in many ways brute force approaches and not entirely necessary for visual data exploration of these large data. A number of turbulence researchers at NCAR have addressed the large data problem by simply coarsening their data either using simple first order or even zero order filters. The reduced resolution, or *browsing* resolution, data set can than be effectively visualized with current tools. The browsing data set still preserves enough

[1] clyne@ncar.ucar.edu

information to allow exploration of gross features. Once a feature of importance has been located, the data may be either rendered in a batch process at its full resolution, or a sub region of the data at a higher resolution may be extracted from the original data and than fed back into the visualization tool for further visual analysis. Clearly this approach is not ideal.

Multiresolution Data Representation

The coarsening process described above is a crude form of a multiresolution data representation. Though the approach is somewhat laborious, it does enable visual or statistical analysis of higher resolution data, albeit in a highly inefficient manner. Other forms of multiresolution representation that are not so cumbersome have been investigated at NCAR. *Vis5d*, an Earth Sciences visualization tool popular with NCAR researchers dealing with geo-referenced data, was modified to provide decimation of the triangle meshes used to represent isosurfaces. Unlike the manual process described above, changing between browsing and full resolution is performed automatically. The modified version of Vis5D maintains triangle meshes at a user-definable browsing resolution and at their original, full resolution. When the user is interacting with the data (changing view points, zooming, or stepping through time) the browsing resolution is displayed. As soon as the user pauses interaction to think or study a static snapshot, Vis5D refines the image with the full resolution isosurface (See figure 1).

Current triangle decimation methods yield remarkable results. Compression rates of over an order of magnitude were observed with the Vis5D application having little or no visual loss of information. Unfortunately, these triangle decimation methods have primarily arisen from commercial interests in delivering 3D content over the web. Though the reduced mesh permits interactivity and represents an improvement over the manual coarsening method, the algorithms are computationally expensive. For example, reducing a 250k triangle mesh to 25k triangles takes ~15 seconds on an SGI/MIPS 250MHz R10k processor. This was accomplished using the fastest mesh reduction algorithm the author is aware of [4]. Interactive selection of isosurfaces values thus becomes impossible with current decimation methods for any large mesh - precisely the meshes we are interested in reducing.

Furthermore, triangle decimation is not a general solution to the problem of visualizing large scientific data sets. Only visualization techniques that produce triangle meshes as the final stage of the visualization pipeline may benefit. A better place to employ a data reduction is at the beginning of the visualization pipeline; to the original floating point data. Lastly, an ideal data representation would to be truly multiresolution providing data approximations at potentially numerous resolutions, not simply a fixed browsing and full data resolution. The optimal browsing resolution may not be determinable apriori and may vary widely with the visualization technique employed and the computing platforms available.

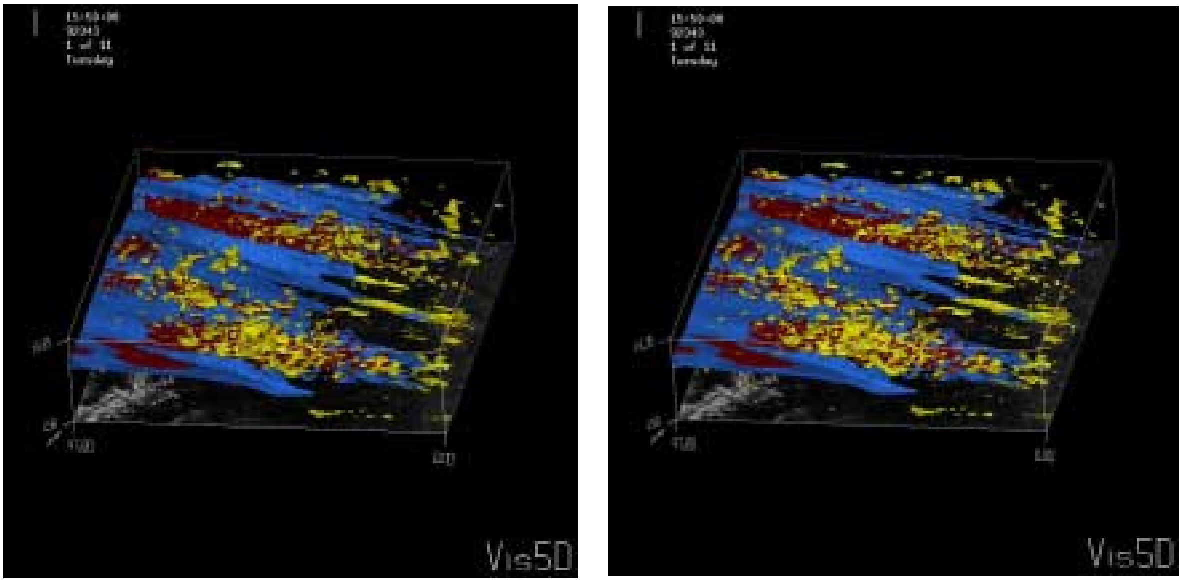

Figure 1 The isosurfaces on the left are the direct output of the Vis5D isosurface generator. On the right, the isosurface meshes for each of the fields shown has been decimated by a factor of 10.

Wavelets

The relatively new theory of wavelets offers a natural mathematical framework for multiresolution data representation. At each successive resolution level the wavelet transformation produces a lower frequency approximation to the previous level along with higher frequency *detail* coefficients representing the loss of information between successive levels. The accuracy of the coarser approximations to the original data is dependent on the nature of the data itself and the choice of wavelet basis function used in the transformation. Exploring optimal wavelet basis functions for various forms of scientific data is an active area or research [5].

Wavelet transforms are invertible. It is possible to have lossless reconstruction of the approximation to the next higher resolution [7]. Transformation to and from wavelet space can be accomplished in linear time and in many instances reconstruction from wavelet space may be accomplished only with additions and subtractions of floating point numbers making the inverse transformation exceptionally fast.

Wavelets have other properties that make them of interest for representing large scientific data sets. The detail coefficients produced at each coarsening represent the loss in information between levels. For the *orthogonal* class of wavelet basis functions the magnitude of the coefficients determines the amount of information lost. Lossy compression schemes can easily exploit this simply by throwing out coefficients whose magnitude falls below some threshold [6]. Additionally, progressive refinement of an approximation is readily performed by simply applying detail coefficients in order from largest to smallest magnitude. Such a scheme was employed by Tao [7] to progressively transmit ocean simulation data. Compression ratios of 50:1 were found to yield acceptable results.

Multiresolution Visualization Tool

In the sections that follow a prototype direct volume rendering application that exploits some of the multiresolution properties of wavelets to permit interactive exploration of large, Cartesian gridded data sets is presented. Wavelet approximations of the full resolution data enable interactivity needed for data browsing while the wavelet detail coefficients enable these coarse approximations to be quickly refined. The implementation is based upon enhancements NCAR's *volsh* volume renderer. The initial enhancements exploit the multiresolution nature of wavelet transforms, but do not take advantage of compression and progressive transmission afforded by the properties of wavelet detail coefficients. However, the design does not preclude their future inclusion.

Volsh Enhancements

Volsh is a portable, direct volume rending environment developed by SCD's Visualization and Enabling Technologies Section. Volsh supports numerous rendering engines including a parallel, software-based implementation of the shear-warp factorization algorithm, MERL's VolumePro, and SGI's Volumizer. The prototype wavelet-based volume renderer supports only SGI's Volumizer API. On a single-pipe Onyx2/IR2 this driver is capable of rendering time-varying 128^3 volumes or static 256^3 volumes at interactive rates (~5Hz). I/O becomes the principle bottleneck for rendering time-varying data at higher resolutions.

Volsh has been extended to support the following new usage scenarios:

1. Reduced resolution, wavelet approximations of the data may be explored during data browsing. I.e. changing viewpoints, zooming, operating cutting planes, stepping through time all occur with a user-selectable browsing resolution. A simple dial allows the user to set the browsing resolution to whatever may be accommodated by a rendering session. A user running on desktop workstation in his office might limit the browsing resolution to 128^3, while a user sitting at the console of a more powerful, multipipe platform might be able to get away with 256^3 or greater resolutions. Once a user stops data interaction (e.g. pauses to look at results on the screen) iterative refinement of the image, up to a user-definable maximum resolution, begins. Changing viewpoints, etc. again immediately causes the resolution to drop down to browsing resolution so that interactively is maintained. In essence, idle computing cycles that occur while a researcher is not interacting with data are used to continually refine the image.

2. For any given time-step, often only a subset of a data volume is of interest. The researcher may zoom in on an area simply by changing the viewpoint or may explicitly subset the data through the use of cutting planes. In these instances large regions of the volume are not in the viewing frustrum and need not be rendered. Consequently the computational requirements for rendering are reduced. The multiresolution tool is aware of this and thus increases the browsing resolution to improve the quality of the image while maintaining roughly the same level of interactivity. I.e. a sample budget is maintained.

Supporting the above usage scenarios requires the following "under-the-hood" modifications:

1. While extracting data at a given resolution from a wavelet encoded dataset is reasonably efficient, it is not fast enough to support interactive retrieval rates for time-varying data. In order to permit interactive exploration of time-varying data, a buffer of most recently accessed time steps is maintained by volsh. The size of buffer is user controllable. Assuming a 128^3 volume with byte-sized voxels, a 500 time-step sequence can be accommodated with only a Gigabyte of memory.

2. Prior to adding support for multiresolution data, the organization of data in volsh's memory was non-blocked. I.e. voxel data was laid out in memory as a single contiguous array with X-axis varying fastest, followed by Y, than Z. For large volumes this organization was not ideal for cache-blocked microprocessor architectures. Furthermore, extracting a data sub region becomes horribly inefficient for sub regions aligned along the Z-axis. Hence, volsh has been adapted to accommodate a blocked data structure where data volumes are comprised of a sequence of smaller sub-blocks.

Data Model

The volume rendering enhancements discussed above assume the existence of a wavelet based multiresolution data model. The term "data model" is used to refer to both the underlying data storage format as well as the operations on the data permitted by the data model's application programmer's interface (API). The data model, like our target application, is a prototype. It provides the base functionality to meet the needs of the rendering application described above as well as serves as a framework for experimentation. While all of the requirements of the model are not yet determined, the following characteristics are available:

Wavelet Basis Functions

The choice of wavelet basis function impacts the run-time performance of the transformations as well as the quality of the approximations of the coarsened functions. There is a trade-off to be made between quality and performance. The data model provides the flexibility to select an appropriate wavelet basis function for a given task by employing the Lifting method for wavelet transformations[8], which allows the user to specify a variety of biorthogonal wavelets.

Data Types

The data model supports wavelet encodings of multi-resolution data in both floating point and integer format at various precisions. Data access methods provide by the API support the conversion between types and precisions with user-provided mappings. For example, when narrowing 32-bit floating point data to unsigned 8-bit integers it is possible to specify the range of values mapped and a means for treating values falling outside that range.

Data Organization

Data are organized in a block structure to support rapid access to data sub regions as well as to facilitate access by applications that are optimized for blocked data. The dimensionality of the blocks is user definable. Wavelet coefficients associated with different transformation levels are stored in separate files. Hence, it is possible to maintain coarser approximations on on-line storage while the more storage-intensive coefficients for the highest resolutions may be maintained off-line if not needed

Data Access

In addition to providing access to the entire volume of data at all resolutions supported by the wavelet encoding, sub regions of data will be accessible at the block level.

Compression and Progressive Transmission

The initial implementation does not support data compression or progressive transmission. However the design does not preclude their future addition.

Performance

The implementation outlined above is capable of browsing a time-varying 1024^3 data set at interactive rates on a modest visualization platform. The user may adapt to the rendering capability of a given platform by simply adjusting the browsing and refinement resolutions with a few simple dials. Tables 1 and 2 below indicate sub region extraction times for an arbitrary 128^3 and 256^3subregionsfrom a 1024^3 volume.

Table 1: Extraction time for arbitrary 128^3 sub region from a 1024^3 volume

Resolution	Extraction Time in Seconds
128x128x128	0.1
256x256x256	0.2
512x512x512	0.4
1024x1024x1024	1.0

Table 2: Extraction time for arbitrary 256^3 sub region from a 1024^3 volume

Resolution	Extraction Time in Seconds
256x256x256	3.1
512x512x512	4.7
1024x1024x1024	6.1

Conclusions

We have described an intelligent approach for addressing large data analysis. Though our prototype system relies on Cartesian data, the methods we have described generalize to other data topologies. Our method is tunable to support whatever computing capability a user has available to them. Lastly, though our prototype application was a volume renderer, other data analysis tools could easily take advantage of these methods.

References

[1] Annick Pouquet, Director of NCAR's Geophysical Turbulence Program, personal communications

[2] PAINTER,J.,MCCORMICK,P.,AND MCPHERSON, A. *Reality monster volume rendering*. In NSF/DOE Workshop on Large Scale Visualization and Data Management (May 1999). Presentations available at ftp://sci2.cs.utah.edu/pub/ldv99/.

[3] S. Parker, M. Parker. Y. Livnat, P.-P. Sloan, C. Hansen, and P. Shirley. *Interactive Ray Tracing for Volume Visualization*. *IEEE Transactions on Visualization and Computer Graphics*, July-September 1999.

[4] Michael Garland and Paul S. Heckbert, "Surface Simplification Using Quadric Error Metrics, Proceedings of SIGGRAPH 97," *Computer Graphics Proceedings, Annual Conference Series, pp. 209-216 (August 1997, Los Angeles, California). Addison Wesley. Edited by Turner Whitted. ISBN 0-89791-896-7*

[5] P. C. Wong and R. D. Bergeron, "Authenticity Analysis of Wavelet Approximations in Visualization," Visualization '95 Proceedings, pp. 184-191, 1995.

[6] E.J. Stollnitz, T.D. DeRose, D.H. Salesin, `*Wavelets for Computer Graphics: A Primer*', Department of Computer Science and Engineering, University of Washington, Seattle, Technical Report 94-09-11

[7] Hai Tao and Robert J. Moorhead , "Progressive transmission of scientific data using biorthogonal wavelet transform,"*Proc. IEEE Visualization 94* , pp. 93-99, Washington, D.C., October 1994

[8] W. Sweldens. The lifting scheme: A custom-design construction of biorthogonal wavelets, *Appl. Comput. Harmon. Anal.* Pp 186-200, 1996