A Prototype Discovery Environment for Analyzing and Visualizing Terascale Turbulent Fluid Flow Simulations

John Clyne[a] and Mark Rast[b]

[a]Scientific Computing Division, National Center for Atmospheric Research, Boulder CO;
[b]High Altitude Observatory, National Center for Atmospheric Research , Boulder CO

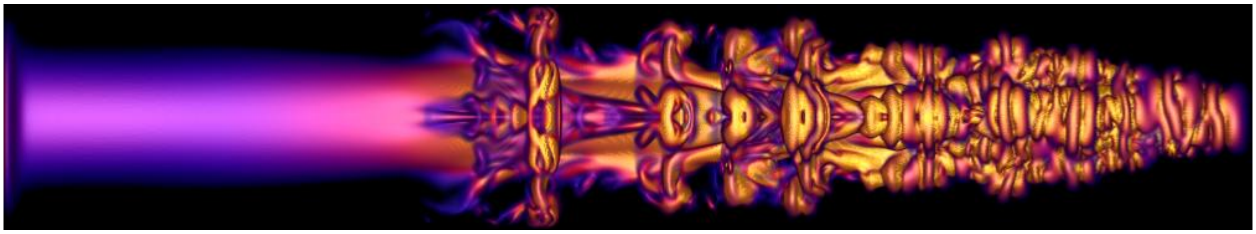

Figure 1. Solar convection is dominated by the formation of thermal downflow plumes in the surface layer. This image displays the enstrophy in a three-dimentional compressible starting plume driven by cooling at the top and descending (left to right) through a highly stratified (increasing density with depth) medium.

ABSTRACT

Scientific visualization is routinely promoted as an indispensable component of the knowledge discovery process in a variety of scientific and engineering disciplines. However, our experiences with visualization at the National Center for Atmospheric Research (NCAR) differ somewhat from those described by many in the visualization community. Visualization at NCAR is used with great success to convey highly complex results to a wide variety of audiences, but the technology only rarely plays an active role in the day-to-day scientific discovery process. We believe that one reason for this is the mismatch between the size of the primary simulation data sets produced and the capabilities of the software and visual computing facilities generally available for their analysis. Here we describe preliminary results of our efforts to facilitate visual as well as non-visual analysis of terascale scientific data sets with the aim of realizing greater scientific return from such large scale computation efforts.

Keywords: analysis, terascale, progressive data access, visualization, turbulence, multiresolution

1. INTRODUCTION

Since the seminal NSF panel report on visualization in scientific computing,[1] visualization has routinely been presented as an essential part of the scientific discovery process. The visualization literature is rich with accounts describing the dependence of domain-scientists on visualization in understanding complex phenomena. While it is certainly true that visualization plays a large and important role in the sciences, we believe the nature of that role commonly differs from that often expounded. Visualization in support of the geosciences at NCAR, and possibly elsewhere, is used primarily in an explanatory role, conveying complex results to a targeted audience only after the analysis that led to those results has been completed using less visual methods. Only rarely is visual interaction with the data used as part of the day-to-day data discovery process. Furthermore, the visualization process is typically carried out by visualization experts with little or no scientific domain knowledge and with widely varying levels of interaction with the domain scientists. The deliverable produced by these visualization support staff on behalf of the scientists is a collection of animations or images that may be used for the purposes of publication or presentation.

Further author information: (Send correspondence to John Clyne)
John Clyne: E-mail: clyne@ncar.ucar.edu
Mark Rast: E-mail: mprast@ncar.ucar.edu

Visualization and Data Analysis 2005, edited by Robert F. Erbacher,
Jonathan C. Roberts, Matti T. Gröhn, Katy Börner, Proc. of SPIE-IS&T
Electronic Imaging, SPIE Vol. 5669 © 2005 SPIE and IS&T · 0277-786X/05/$15

With visualization's well-documented potential to provoke unexpected insights, why has it not been embraced by more scientists as a tool for scientific discovery? We believe the reasons are twofold: 1) inadequate software, and 2) inappropriate facilities. We also believe that with improvement in both these areas greater scientific return from large scale computational efforts is possible.

1.1. Inadequate software

Visualization packages presently available to domain scientists suffer numerous problems that limit their greater acceptance. With little market for domain specific products, today's commercial and freeware offerings tend to target broad needs and lack domain specific capabilities such as the mapping transformations needed by researchers in the geosciences. Ironically, because of the generality of many of these packages they also tend to suffer from *feature creep*; they are often encumbered by features and capabilities of little interest to specific groups. The byproduct of this generality is often unwanted complexity.

Perhaps even more problematic than ease-of-use issues or lack of domain-specific features is the utter absence of numerical capabilities needed for quantitative analysis. It is not enough to simply visualize scientific data to understand it. In-depth analysis requires the application of a host of mathematical, statistical, image, and signal processing functions in order to truly gain insight. While these capabilities may be found in non-visual analysis packages, such as Mathematica^{TM}, Matlab^{TM}, or IDL^{TM}, a close integration of advanced three-dimensional (3D) visualization and quantitative numerical tools is not found. While some analysis packages offer limited 3D visualization algorithms, their performance is often quite poor particularly for very large data volumes.

Quantitative interrogation capability is not the only important feature lacking in today's advanced visualization packages, another is the ability to manipulate the data as it is being analyzed and return the output of the manipulation in a convenient visual form. Without a means to derive and visualize new quantities from raw model outputs, the analysis process is hampered and the visualization tool becomes a secondary accessory to the discovery process. In principle, derived quantities might be generated for an entire data set in a pre-analysis step, but this limits either creativity or interactivity by restricting analysis to anticipated needs or relegating it to a batch mode. Moreover, for large data sets it may be far more efficient to derive new quantities only as needed, rather than compute and store subsequently unused variables in a shotgun approach.

1.2. Inappropriate facilities

Exploration of terascale data sets demands substantial computing resources: high performance, high capacity storage systems, fast networks, powerful computational platforms, and powerful graphics systems. Because of the expense of these systems, combined with the physical distance limitations imposed by high resolution video signals, these facilities are generally made available to users as shared resources offered by way of a visualization lab. The user reserves time in the lab to perform visual computing work as needed.

There are two problems with this scenario. The first is simply one of convenience; the lab may or may not reside in close proximity to the user. It is not uncommon for a single visualization lab to be shared among several campuses. Even when the facility is located close by, the lab is not an office. The second is an even greater hindrance to the use of visualization in data exploration and scientific discovery; the scheduled nature of a shared resource puts it in conflict with the highly responsive needs of analysis. Analysis work cannot be effectively segmented into visual and non-visual tasks. It is not practical to, for example, reserve a visualization lab for Friday afternoon while performing non-visual analysis tasks earlier in the week. To be most effective, it must be possible to access visual and non-visual tools in tandem and spontaneously. Ideally, the results of any analysis procedure performed on a 3D data volume should be seemlessly and interactively returned to the user, visually if desired.

In this application paper we describe preliminary efforts aimed at overcoming the barriers some NCAR scientists face in their use of visualization for data-motivated scientific discovery. We present initial results from a prototype environment developed specifically to meet the analysis needs of researchers studying fluid turbulence. In particular, we discuss the analysis of a very large numerical simulation of a compressible thermal starting plume motivated by studies of solar convection. Our prototype environment offers advanced 3D visualization combined with quantitative analysis and data manipulation capabilities, and is capable of accommodating terascale size data sets. We also discuss our experiences with a remote image delivery technology to address the aforementioned issues with shared visualization labs.

2. RELATED WORK

A number of other authors have presented experiences and discussed challenges associated with the visualization of large earth sciences fluid flow simulations. Clyne et al. describe the virtues of various visualization methods, the benefit of stereo viewing, and the need for interactive visualization in the exploration of a high-resolution quasi-geostrophic (QG) turbulence simulation.[2] Silver and Wang discuss the application of feature tracking to a variety of earth sciences fluid flow data sets, including simulations of rotating, stratified turbulence, also derived from the QG equations of turbulence.[3] Dorch reports on astrophysical magnetohydrodynamic simulations and the exploration of the solar dynamo problem. He notes discoveries made through visualization that would have been difficult otherwise.[4] More recently Bemis et al. discuss experiences with a system designed to visualize and analyze simulated and observed hydrothermal plumes.[5] In particular, the effect of ocean currents and the internal turbulent structure of these plumes are studied. Custom visualization software is developed and used to compare simulated with observed ocean plumes. Erlebacher et al. study the dynamics of mantle convection in the earth's core, again focusing on thermal plumes.[6] Custom modules for the commercial visualization package, Amira^{TM}, are developed to aid in the exploration of the 501^3 resolution mantle convection simulations.

All of these efforts employ novel methods to address the software and facility challenges discussed in Section 1. However, none of them present a comprehensive environment that combines qualitative visual browsing with quantitative interrogation, which is necessary for fully effective data analysis.

3. PROTOTYPE ANALYSIS ENVIRONMENT

Our prototype analysis system is comprised of a combination of key technologies and resources that together provide an environment for effective data exploration. Each of these technologies is discussed in turn in the following subsections.

3.1. Application Software

Our application software targets a specific usage model, aimed at exploring data sets that are time-varying, multi-variate, and possessing very high spatial resolutions. The usage model assumes a repetitive process whereby the user employs visualization primarily as a filter to reduce a vast spatial-temporal volume of data to a manageable size that can then be accommodated by more quantitative tools. In essence, qualitative visualization is used to find features of interest that cannot be easily or efficiently detected by other means. Once a region of interest is identified, quantitative numerical techniques can be applied to the focus region. This progression then repeats. By narrowing the data domain with highly interactive visualization, we can reduce significantly the amount of data that must be processed by other, far more computationally demanding analysis methods.

The software tools we employ consist of three application level components, all layered upon a common, high-performance data model. The application level components include a custom, interactive volume renderer that we have developed specifically to meet the needs of turbulent fluid flow researchers; a widely-used commercial scientific data analysis package, RSI's Interactive Data Language (IDL^{TM})[7]; and the freeware visualization development framework, the Visualization Toolkit (VTK).[8] All three of these technologies are coupled together, albeit loosely in our current prototype, into a single problem solving environment.

Fundamental to the coupling and interaction between software applications is a custom scientific data file format, and associated application programmer interface (API), upon which all applications are layered. Our data representation scheme supports progressive access and efficient sub-region extraction. Through a highly-efficient wavelet-based transformation we are able to deliver data to applications at varying resolutions, permitting the user to make speed/quality tradeoffs throughout the analysis pipeline. Referring back to our targeted analysis model, the scientist can visually browse data sets that are vast in the spatial domain by simply requesting an appropriate approximation level, reducing the region of interest, or a combination of both. Once a feature of interest is detected at low resolution it may be refined and viewed at less interactive rates or operated on by other tools in the environment. Since our multiresolution approach operates on the raw data itself, as opposed to downstream outputs of the visualization pipeline, other non-visual tools may exploit its benefits. A more detailed description of our progressive data access scheme is provided in Section 4.

3.1.1. Multiresolution Data Browser

The Multiresolution Data Browser (MDB) is an interactive, direct volume renderer that provides ”quick-look” capability for scalar data that is large, multi-variate, and time-varying. MDB targets the qualitative, visual exploration of relatively large-scale feature dynamics in a time-varying scalar, flow field. Data is ingested into MDB via the progressive access scheme just described, thus enabling speed/quality trade-offs essential to maintaining interactivity. To further improve interactivity, MDB relies on relatively low image quality, but high-performing, texture-hardware accelerated volume rendering via SGI's Volumizer API.[9]

MDB is in general devoid of capabilities not deemed to be of interest to its target user group, and its graphical user interface (GUI) therefore decidedly simple. Essential features that are present include the following:

- The user may precisely limit the region of interest in both the spatial and temporal domains, significantly reducing the load on the application. Similarly, explicit control over the spatial resolution is afforded by our progressive data access storage format.
- Multiple data variables may be input into MDB and qualitatively compared with lockstep viewing parameters.
- MDB operates directly on floating point data. Quantization to 8 or 16 bit integer quantities required for hardware texture-based volume rendering is performed on the fly at a negligible cost, facilitating exploration of data with high dynamic range.
- To support real-time animations of time-varying data, MDB provides a memory cache, whose size may be controlled by the user, which buffers the most recently viewed data regions. With appropriate selection of the spatial subregion, resolution and the number of time steps, it is possible to smoothly animate through a reasonably long temporal sequence.

Aside from its multiresolution and time-varying data capabilities, MDB is not functionally very different from many other volume rendering applications. What differentiates MDB, and makes this browser of particular value to our goals, is its ability to interact with other applications. By using a very simple communication mechanism MDB may convey session parameters (e.g. current viewpoint, time step, sub-region coordinates, etc.) to other applications, greatly facilitating operation by multiple tools on an identified region of interest. Our much-needed quantitative analysis and data manipulation capabilities can then be provided by tools far more adept at these functions, such as IDL^{TM}. Thus it becomes not only possible, but relatively easy, for multiple applications, offering potentially widely varying functionality, to operate on precisely the same temporal-spatial region of data.

3.1.2. Interactive Data Language

IDL^{TM} is a commercial data analysis and manipulation tool that is widely popular with many domain scientists. The package provides an array-based interpreted language supporting a rich set of numerical functions and plotting routines needed for quantitative analysis. Additionally, many community user libraries with application specific analysis procedures exist. We note that IDL^{TM} also provides a toolkit of rudimentary 3D visualization algorithms. However, the performance of these 3D capabilities is quite poor, particularly for very large data volumes. Key to our application is the extensibility of the language, allowing us to add IDL^{TM} functions for reading and writing our custom format data files.

3.1.3. Visualization Toolkit

VTK is a freeware, object-oriented framework for multi-platform visualization tool development, supporting a wide array of basic and advanced visualization algorithms.[8] As with IDL^{TM}, we have extended VTK's data import modules to include support for our progressive data access scheme. Thus it is possible for many of VTK visualization components to take advantage of multiresolution data. In particular, we have assembled a simple VTK ray-casting application, providing a means to batch compute high-quality volumetric renderings of our data. The ray caster may be invoked directly from MDB to produce a high-fidelity version of the relatively low-quality image presented by MDB, or the ray caster may be invoked as a batch application from the command line for the purpose of producing a complex animation sequence. The images in Figures 1 and 4 were all generated with VTK.

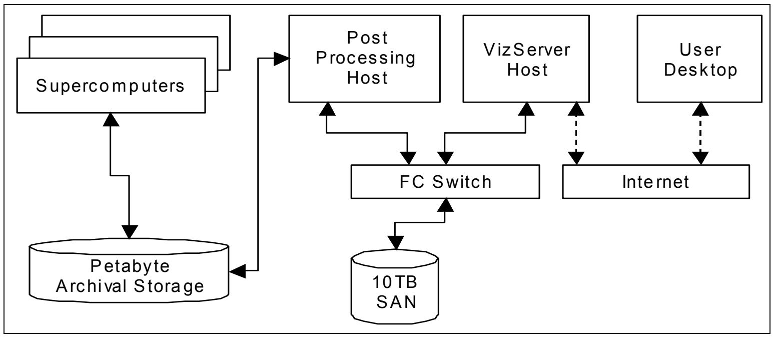

Figure 2. Computing systems diagram.

3.2. Shared file systems

Data sets that are terabytes in size are time consuming to move, and costly to store. A five-terabyte data set, for example, requires a day or more to copy at GigE or FibreChanel speeds and more than a week to transfer over a 100MBit network. Even with recently emerging RAID systems based on inexpensive commodity SATA disk drives, the cost of a system capable of holding this much data may be tens of thousands of dollars. Thus for large data it is highly desirable to keep only a single copy and move the data as little as possible.

Unfortunately, in many computing centers a single computing platform may not be available or appropriate for all visualization, analysis and post-processing tasks. Data replication would seem inevitable. However, the relatively recent advent of Storage Area Networks (SANs), combined with shared filed systems, provides a means for multiple computing platforms to share access to a single storage asset with performance near or identical to that of dedicated, local storage. Integral to our analysis environment, is such a system: a 10TB SAN connected to our visual and post-processing computing platforms.

3.3. Remote Image Delivery

Remote visualization has been an active research area for many years. With the advent of powerful desktops, ubiquitous high-bandwidth networking, and the many advances in video codec design, image-based remote delivery systems are now deployable technologies. Image-based remote rendering has an advantage over geometry based approaches in that the technology is insensitive to the complexity of the scene. Performance is determined only by the resolution of the image being displayed, the available network bandwidth, and the image encoding/decoding capabilities of the server and client. Today's powerful desktops are more than capable of decoding an image stream in real time, making possible the virtual elimination of distance limitations imposed by analog video signals. Thus the power of the visual supercomputer may now be harnessed from the convenience of one's office.

We have deployed a commercial remote visualization technology: SGI's VizServer.[10] The host system is a 10 processor, 4 pipe Onyx3800, dedicated for VizServer use. The machine is attached to our 10 terabyte SAN, providing users with a large working data repository for visualization.

Figure 2 depicts the connectivity of the relevant systems in our computing environment.

4. PROGRESSIVE DATA ACCESS

To help accommodate exceptionally large data sets we have devised our own scientific file format for the storage of floating point data sampled on a regular Cartesian grid. A key feature of our representation scheme is the ability to permit progressive data access in combination with rapid subregion extraction. Complete details of the storage strategy are available elsewhere.[11] Here we briefly summarize the approach and list its most important attributes.

Floating point data stored in our custom file format first undergo a wavelet transformation. Wavelet transformations map functions into a space consisting of an overall coarse approximation of the function together with the detail coefficients permitting the coarsened approximation to be refined at various scales. Thus it becomes possible to progressively access the data. That is, the data may be reconstructed at progressively finer resolutions. Additionally, we block the data before it is written, paying careful attention to optimal IO request sizes, making possible efficient extraction of grid subregions.

The wavelet transformation process is lossless, save for floating point round off errors. Thus unlike many preprocessing strategies aimed at improving performance, multiple copies of the data are not required. The total number of wavelet coefficients are equal to that of the number of samples in the function. Thus no additional space is required by the wavelet representation. Other notable attributes of our data representation strategy include:

- Both the forward and inverse transform are highly efficient. This is an important consideration for very large data sets. An encoding scheme with long preprocessing requirements is not practical for terascale sized data.
- The implementation operates out-of-core for both forward and inverse transforms, permitting extremely large grids to be processed using only a modest memory footprint.
- Data approximations are produced by reconstructing and resampling the original data, not by simply subsampling, leading to higher quality approximations.

Lastly, we note that by supporting multiresolution at the raw data level, as opposed to a downstream stage typical of multiresolution visualization approaches, the benefits of multiresolution become available to all components of the analysis pipeline. Not only does our triangle rendering speed increase, for example, but so does our isosurface extraction speed. Similarly, our non-visual analysis tools benefit.

5. RESULTS

5.1. Process

We have used our prototype analysis environment in the preliminary exploration of a number of numerically modeled astrophysical turbulence data sets with spatial resolutions ranging from order 256^3 to 512^2x2048, and each possessing hundreds of time steps. Each simulation outputs five double-precision field variables: a three-component velocity field, fluid temperature, and density. The end-to-end, compute-to-publish process for each can be summarized briefly as follows: run numerical simulation of the fully-compressible Navier-Stokes equations on a distributed-memory supercomputing cluster; archive results to tape-based mass storage system; copy archived data to SAN described previously; re-assemble individual processor data files into contiguous data sets; transform data to our progressive data access scheme, reducing precision from 64 to 32 bits; visualize and analyze data.

For the largest of the data sets, a simulation of a 3D compressible starting plume, the compute and archive phase required over six months running on a 128-processor supercomputer and generated nearly 10TBs of double precision floating point data. Retrieving the data from tape, with an effective transfer rate of only 5MBs/sec, and re-assembling the processor files, consumed another two months. By comparison, our highly efficient wavelet transformations required only a couple of days to complete. The visualization and analysis process is on-going, using the tools previously describe, from an office located four miles from the visualization/analysis computer site and connected via switched 100Mbit ethernet.

5.2. Overall effectiveness of prototype environment

One measure of the overall effectiveness of the progressive data access scheme implemented is summarized by the timings listed in Table 1. These indicate in seconds the time taken to perform a single analysis procedure or to visualize a single frame of an animation sequence. The visualization schemes tested were those discussed previously, while the analysis, using IDL$^{\text{TM}}$, included a simple scalar array multiplication to evaluate the fluid pressure p from its density ρ and temperature T,

$$p = \rho T, \tag{1}$$

an evaluation of the fluids ionization fraction which involves transcendental function evaluation (exponentiation) via the Saha equation

$$\frac{y^2}{1-y} = \left(\frac{2\pi m_e k}{h^2}\right)^{3/2} \frac{T^{3/2}}{\rho N_{\text{A}}} e^{-\chi_{\text{H}}/kT}, \tag{2}$$

Resolution	p	y	ω^2	mdb	vtk
Full	6.82s	1130s	2330s	143s	178s
1/2	1.36s	44.9s	295s	18.3s	24.6s
1/4	0.19s	5.35s	34.8s	2.20s	4.10s
1/8	0.01s	0.63s	3.04s	0.55s	1.90s

Table 1. Execution time for typical analysis and visualization operations (see text). Full domain size is $504^2 \times 2048$. Visualization results are for time-varying data and include costs for rendering to a 512^2 image, inverse wavelet transformations, and I/O from disk. Analysis operations do not include I/O or transformation costs. All experiments were conducted on SGI systems with 250MHz, and 600MHz processors, respectively. Note: physical memory size (10 GBs) was insufficient for the analysis operations at full resolution and the system was forced to page from disk.

Resolution	p	y	ω^2
Full	0	0	0
1/2	1.09	0.03	85.7
1/4	2.53	0.14	97.3
1/8	3.79	0.65	99.8

Table 2. Error $\varepsilon(\%)$ introduced by grid coarsening in all three directions ($1/2$ resolution is $1/8$ problem size) for calulations used in typical analysis operations (see text).

(where ρ is the fluid density, T its temperature, χ_{H} is the ionization potential, N_{A} is Avogadro's number, and atomic constants m_e, k, and h are the electron mass, Boltzmann's constant and Planck's constant respectively), and the computation of the fluid enstrophy via a second-order finite-difference evaluation of the curl of the fluid velocity

$$\omega^2 = (\nabla \times \mathbf{u})^2 \,. \tag{3}$$

This last operation requires six separate gradient evaluations (two array shifts each) on the full 3D domain.

The timings were done separately for resolutions halved in all directions three times, as provided by our progressive data access scheme, so that the coarsest arrays operated on were 512 times smaller than the full native resolution of the data. They clearly indicate that both visualization and analysis operations occur interactively only on significantly reduced domain sizes or after dramatic resolution reduction. Of course, if the resolution of the simulation output is reduced prior to analysis operations, error is introduced into the calculation. Its magnitude and scaling with resolution depends on the particular operation being performed. We estimate the analysis error as the pointwise normalized maximum-absolute-error (the maximum absolute difference between the result calculated at full resolution and sampled at lower resolution and the result calculated from the lower resolution field variables, normalized by the value of the former at that point). The error measure was chosen in this way to avoid problematic error measures near field values of zero (the maximum absolute error is chosen and then normalized rather than the maximum relative error being used). This error measure ε is quoted in Table 2 and plotted in Figure 3 as a function of the time it takes to perform the analysis operation in units of that needed

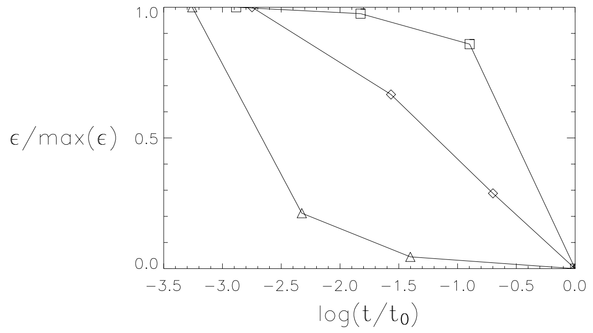

Figure 3. Normalized maximum-absolute error as a function of calculation compute time on successively coarsened grids. Shown for the determination of p, y, and ω^2 (see text) with *diamonds*, *triangles*, and *squares* respectively.

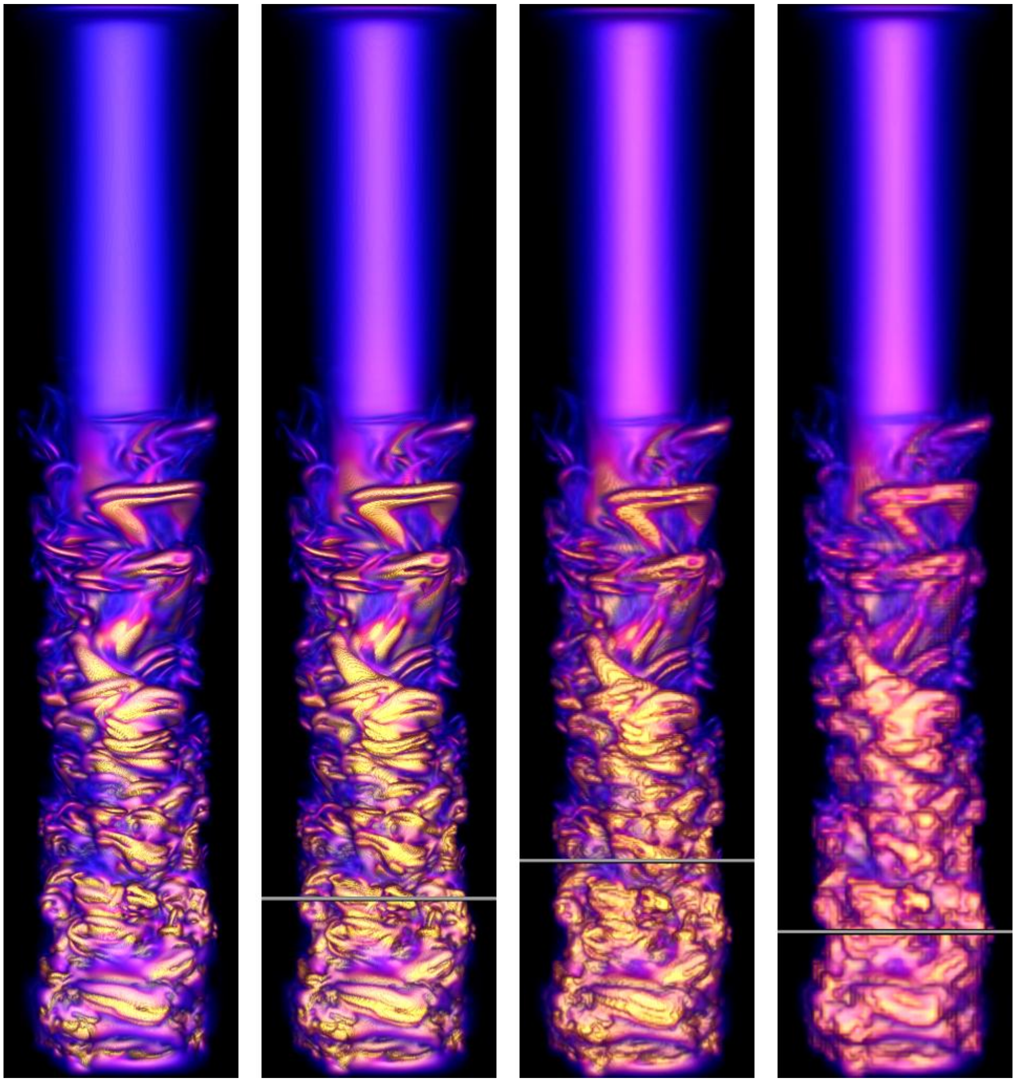

Figure 4. A single 3D compressible downflow plume rendered with VTK at four different resolutions, from left to right: full ($504^2 \times 2048$), 1/2th ($252^2 \times 1024$), 1/4th ($126^2 \times 512$), and 1/8th ($63^2 \times 256$) respectively. The horizontal lines through the coarsened data renderings indicate the location of the plane containing the maximum error (See Figure 5).

for the operation at full resolution t_0. In that figure the errors and timings for p are plotted with diamonds, while those for y and ω^2 are shown with triangles and squares respectively. For both the algebraic operations tested, the error introduced by employing low resolution field variables in the calculations remain quite low even for very large grid size reductions. Consequently, substantial savings in analysis compute time is possible without severe penalty. For gradient evaluations, as in our enstrophy computation, however, the pointwise error grows very quickly to very large values, and for these types of analysis operations significant care must be taken in the use of computations based on low resolution reconstructions of the field variables. The time savings are great, but the error penalty is likewise large.

The error tolerance in data analysis is very sensitive to the application aim. Pointwise maximum errors of a few percent (as found in our algebraic operations on coarse grids) may be tolerable for lowest order quantitative exploration of fluid dynamic force balance or turbulence statistics, while those approaching 100% (as in our coarse grid enstrophy calculation) are probably not. However, visual exploration often has much greater error tolerance. Figure 4 displays a volume rendered image of the thermal plume enstrophy field at four resolutions (each successive image having a factor of two coarser grid in all directions as provided by our wavelet-based progressive data access scheme). Quite a bit of the structure in the plume is preserved as the resolution is reduced, and feature identification (finding locations of enstrophy maxima for example) is quite robust through at least two levels of coarsening. This is further illustrated by Figure 5, which shows grey scale images of the enstrophy in horizontal planes and one-dimensional cuts through the points of maximum error at the three coarsened resolutions. For reference, the location of these sites are indicated by the horizontal bars superimposed on the reduced resolution plume images in Figure 4 The first image of each set in Figure 5 displays the the enstrophy field as calculated from the velocity field variables at full resolution, the second is a coarsened image of the first (lower order wavelet reconstruction), and the third an image of the enstrophy calculated from the lower-order reconstructed velocity fields. The color coded horizontal and vertical cuts through these images, plotted below them, have been normalized by their peak amplitudes. It is clear that while the pointwise error in the coarsened calculations is large (Table 2), the locations of vorticity structures are well represented at all but the lowest resolution.

This is particularly important in light of the enormous reduction in time (by nearly two orders of magnitude) for the twice coarsened calculation. During the course of interactive analysis, many such low resolution calculations can be performed and examined visually to gain insight into the spatial and temporal relations between derived physical properties. The most important of these may have to be subsequently tested and verified at higher resolution, but significant amounts of hypothesis testing, not possible at full resolution, can be achieved via the efficiency benefits of lower resolution. Since analysis proceeds via data inquiry, reduction of the response delay to interactive time scales can greatly facilitate that inquiry by linking a series of questions into a chain, making each subsequent question more relevant to the motivation of the prior one.

6. DISCUSSION

We find that multiresolution data access can efficiently facilitate interactive visualization and analysis of very large data sets. Most quantitative analysis and visual browsing can be performed without undue information loss or error introduction, and with significant savings in time, at reduced resolutions. Full resolution checks of the conclusions can be facilitated efficiently on conveniently extracted subdomains, before final calculations are performed at full resolution on the full domain, if that is ever required. Thus by using MDB to find regions of interest we can greatly reduce the load on analysis tools which are either not implemented as efficiently or are inherently more computationally demanding. Field gradient measures proved the greatest challenge to multiresolution analysis, and further tests employing higher order wavelet and gradient evaluation schemes are planned.

We believe that the integration of our custom volume renderer, MDB, with the other software applications and analysis packages leads to significant efficiency gains in our efforts. Though the current communication mechanism we have implemented between these tools is simplistic and unidirectional it has proven an effective prototype. The coupling of MDB, IDL^{TM}, and VTK provide a powerful desktop system that permits us to visually browse data for exploratory purposes, apply quantitative techniques quickly to smaller or coarsened volumes, derive new insights into the relationships between phyical variables, and finally produce high quality renderings to illustrate these to a broader audience.

These tasks can all be performed in an efficient manner and, augmented with image-based remote delivery, conducted from desktop sites far from storage and compute facilities. In cases where the geographic distance between data and researcher are extreme, this capability may permit data exploration that would be otherwise not be possible. In our own situation, where the separation is only a few miles, remote visualization affords a convience that enhances scientific productivity, by facilitating spontaneous investigation.

Lastly, we cannot underestimate the value of our large, shared storage area network. Without the ability to contain a multi-terabyte data set on a high-bandwidth storage system that may be accessed from multiple compute platforms, our efforts would be gravely hampered.

7. CONCLUSIONS

We have presented our preliminary experiences with the application of a collection of software and hardware technologies employed in the exploration of a terascale data set. Our environment combines custom developed exploratory visualization tools, targeting the needs of the numerically-simulated-turbulence community, with commercial and freeware applications adept at quantitative analysis and high-quality rendering. The foundation of this software environment is a multiresolution

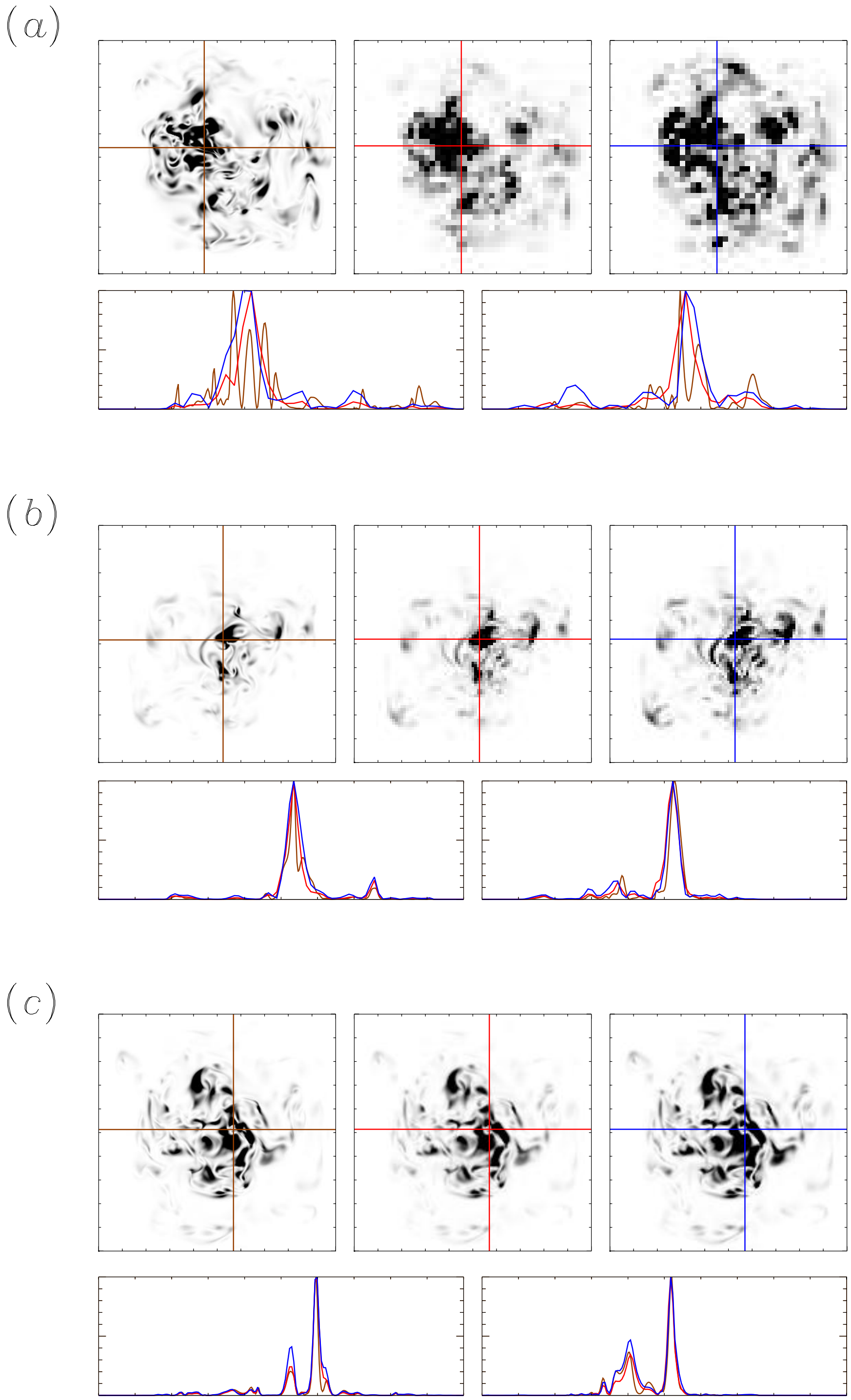

Figure 5. Horizontal enstrophy planes and plots through the sites of maximum error in the reduced resolution 3D domains (locations indicated in Figure 4): 1/8 resolution (1/512 problem size) in (*a*), 1/4 resolution (1/64 problem size) in (*b*), and 1/2 resolution (1/8 problem size) in (*c*). In each group the left most image shows the enstrophy grey-scaled at full resolution, the middle plot a low order wavelet reconstruction of that, and the right most plot the enstrophy as calculated from a low order reconstruction of the velocity fields. Only the right most image represents a calculation with significant computational savings. The color coded plots below the grey-scaled images show the enstrophy along the indicated horizontal (*left* panel) and vertical (*right* panel) cuts (normalized by the maximum value along them).

data representation scheme that provides progressive access, permitting the researcher to make speed/quality trade-offs. Software applications with visual components, strongly dependent on high-performance 3D computer graphics, are made accessible to the scientist via a commercial, remote visualization technology, and are based on a large, high-performance storage area network, capable of housing and quickly delivering, or recording, large data sets to a variety of computing platforms without unnecessary data copying. This trio of applications and technologies – qualitative and quantitative analysis software, remote image delivery, and a capacious SAN – have greatly facilitated the hypothesis-driven, scientific discovery process and substantially increased productivity in our analysis of terascale data.

Our investigation of the 3D plume data set, as well as a number of other large simulations not discussed in this paper, is on-going and much analysis remains to be done. In parallel with the study of these data we also plan further exploration and quantification of our multiresolution methods and their applicability to higher-order analysis operations. We would like to be able to quantifiably state the validity of various operations with approximated data. We also plan to further develop our software applications, seemlessly integrating bidirectional communication and taking them from the prototype stage to widely-usable tools that may be distributed to others in the community. Finally, we hope to expand the range of application to include non-uniformly gridded data and vector field variables.

Acknowledgements

This work is partially supported by NSF ITR-0325934. The authors would also like to acknowledge Joseph Mendoza for keeping our systems running, post-processing an enormous amount of data, and generating countless images and animations.

REFERENCES

1. B. H. McCormick, T. A. Defanti, and M. D. Brown, eds., *Visualization in Scientific Computing*, vol. 6, ACM SIGGRAPH, 1987.
2. J. Clyne, T. Scheitlin, and J. Weiss, “Volume visualizing high-resolution turbulence computations,” *Theoretical and Computational Fluid Dynamics* **11**(3), pp. 195–211, 1998.
3. D. Silver and X. Wang, “Tracking and visualizing turbulent 3d features,” *IEEE Transactions on Visualization and Computer Graphics* **3**(2), pp. 129–141, 1997.
4. S. B. F. Dorch, “Astrophysical mhd simulation and visualization,” *Lecture Notes on Computational Science and Engineering* **13**, pp. 209–220, 2000.
5. K. G. Bemis, D. Silver, P. A. Rona, and C. Feng, “Case study: a methodology for plume visualization with application to real-time acquisition and navigation,” in *Proceedings of the conference on Visualization ’00*, pp. 481–484, 2000.
6. G. Erlebacher, D. A. Yuen, and F. Dubuffet, “Case study: visualization and analysis of high rayleigh number — 3d convection in the earth’s mantle,” in *Proceedings of the conference on Visualization ’02*, pp. 493–496, 2002.
7. S. Alam and C. Schauble, “Elements of idl.” ftp.cs.colorado.edu/pub/HPSC/ElementsOfIDL.ps.Z, 1995.
8. W. J. Schroeder, K. M. Martin, and W. E. Lorensen, “The design and implementation of an object-oriented toolkit for 3d graphics and visualization,” in *Proceedings of the conference on Visualization ’96*, pp. 93–100, 1996.
9. P. Bhaniramka and Y. Demange, “OpenGL volumizer: A toolkit for high quality volume rendering of large data sets,” in *Proceedings of the Volume Visualization and Graphics Symposium ’02*, pp. 45–54, 2002.
10. C. Ohazama, “OpenGL VizServer white paper,” tech. rep., SGI, 1999.
11. J. Clyne, “The multiresolution toolkit: Progressive access for regular gridded data,” in *Proceedings of Visualization, Imaging, and Image Processing ’03*, pp. 152–157, 2003.