Visualization Viewpoints

Editor:
Theresa-Marie Rhyne

In Situ Visualization at Extreme Scale: Challenges and Opportunities

Kwan-Liu Ma, ***University of California, Davis***

The advent of supercomputers gives scientists the power to understand the universe's evolution, predict climate change and variability, design more efficient energy production, and study many other complex scientific phenomena and engineering design problems with numerical simulations. As supercomputers' speed and capacity continue to increase, scientists are beginning to model physical phenomena and chemical processes at an unprecedented level of detail and accuracy, making possible new discoveries and breakthroughs in many areas of study.

The rate of increase in supercomputing power isn't slowing down. For example, the US Department of Energy has made a significant investment through its SciDAC (Scientific Discovery through Advanced Computing; www.scidac.gov) program on computational-science research, scientific-computing software tools, and hardware infrastructure. China, Japan, and some European countries are making similar investments. Petascale computers, which can perform 10^{15} operations per second, have become available, and we're moving toward exascale computing (that is, achieving 10^{18} operations per second).

However, the resulting data from simulations at the petascale level are too large to store and study directly with conventional postprocessing visualization tools. This problem will only become more severe as we reach exascale computing. A plausible, attractive solution involves processing data at simulation time (called *in situ visualization*) to reduce the data that must be transferred over networks and stored and to prepare the data for more cost-effective postprocessing visualization. Here I discuss critical issues in realizing in situ visualization and suggest important research directions.

The Challenge of Large-Scale Simulation

For handling and analyzing petascale-level simulation output, the current practice is to first dump a temporal subset of the data to disk. The researchers then examine the data in an offline postprocessing step using a dedicated visualization machine (see Figure 1). A large-scale simulation typically runs from tens of thousands to hundreds of thousands of time steps, but scientists can't afford to store every time step's output. They usually save only every 300 to 500 time steps, resulting in several terabytes of data. At petascale and exascale levels, the amount of data would be several orders of magnitude greater, making the simulation spend most of the supercomputing time doing I/O and take much longer to finish.

Even when scientists can afford to keep most of the data for analysis, they must transfer the data to a machine with sufficient capacity and processing power to do the desired visualization and analysis. However, the data transfer cost is likely unacceptably high, and the visualization machine would need to be almost as powerful as the supercomputer producing the data. The alternative is keeping and looking at an even smaller temporal subset (and in some cases a spatial subset) of the data, which defeats the purpose of performing the original high-resolution simulation enabled by petascale computing.

Better approaches are to not move the raw data or to minimize the data that must be stored and moved. One strategy is to have both the simulation and visualization calculations run on the same parallel supercomputer so that the data can be shared (see Figure 2). Such simulation-time coprocessing can render images directly or extract features—which are much smaller than the full raw data—to store for later examination. So, reducing both the data storage and transfer cost early in the data analysis pipeline optimizes the overall scientific discovery process.

Such in situ visualization isn't new. However, scientists have seldom adopted this approach, for two reasons. First, most scientists are reluctant to use their supercomputer time for visualization calculations, especially when the calculations are expensive. Second, coupling the parallel simulation code with the visualization code could take

 Published by the IEEE Computer Society

significant effort. In particular, the domain decomposition optimized for the simulation is often unsuitable for parallel visualization, resulting in the need to replicate data to speed up the visualization calculations.

Nevertheless, in situ visualization is clearly a feasible solution for the upcoming extreme-scale data problem presented by scientific supercomputing. It's a particularly desirable solution because at simulation time, all relevant data about the simulated field and any embedded geometry are readily available for the visualization calculations. All such relevant data and information would be prohibitively expensive to collect again and compute during postprocessing.

We can conduct in situ processing to compress data, extract salient features from the data, create a data hierarchy, or build indexing for fast data access. We can conduct in situ visualization to achieve runtime monitoring and even steering of the simulation. Or, on the basis of domain knowledge, we can create snapshot images or an animation characterizing the simulation. Figure 3 displays selected time steps from in situ visualization of a turbulent-combustion simulation running on Oak Ridge National Laboratory's Cray XT5 supercomputer.[1]

In situ visualization presents many new challenges to both simulation and visualization scientists. Before we can realize this approach, we must answer several questions:

- How do simulation and visualization calculations share the same processor, memory space, and domain decomposition?
- What fraction of the supercomputer time should we devote to in situ data processing and visualization? As in situ visualization becomes a necessity rather than an option, scientists must accept visualization as an integral part of the simulation.
- What data-processing tasks and visualization operations are best performed in situ? To what extent does the monitoring scenario stay relevant, and how do we effectively couple monitoring with knowledge-driven data reduction?
- What unique requirements do in situ visualization algorithms have?
- As we store less raw data to disk, what supplemental information should be generated in situ?
- To provide generic support for in situ visualization, how do we abstract out the complexity of the simulation, mesh structures, parallel implementation, and so on?
- Can existing commercial and open source visualization software tools be directly extended to support in situ visualization at extreme scales?

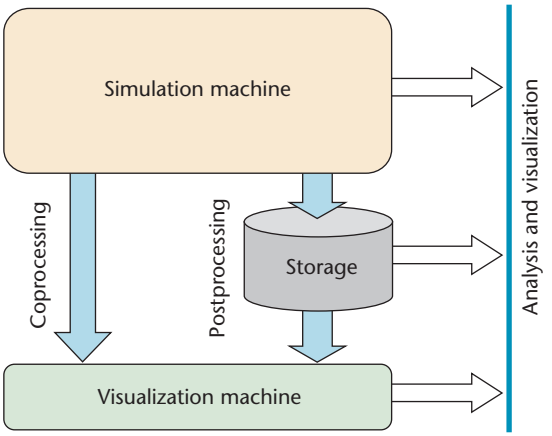

Figure 1. Transferring the entire data generated by a petascale simulation to a storage device for postprocessing visualization or a visualization machine for coprocessing could become a serious bottleneck because I/O would take most of the supercomputing time.

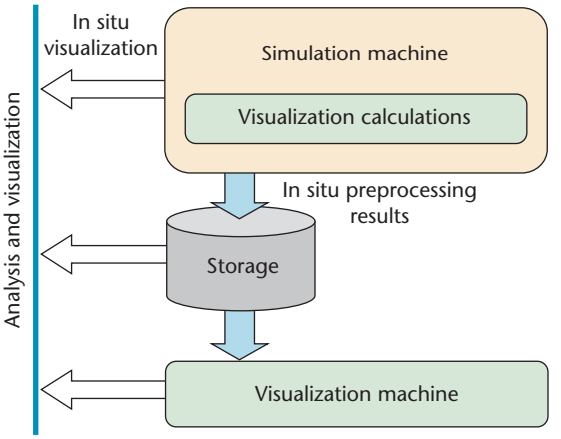

Figure 2. A more feasible approach to data analysis at extreme scales is to reduce and prepare the data in situ for subsequent visualization and data analysis. The reduced data, which typically are several orders of magnitude smaller than the raw data, can greatly lower the following data transfer and storage costs.

To answer these questions and to understand how this new approach affects simulations and subsequent visualization tasks, we must embark upon a new line of research in close collaboration with simulation scientists. The SciDAC program and the US National Science Foundation's cross-cutting programs such as Cyber-Enabled Discovery and Innovation (www.nsf.gov/crssprgm/cdi) and PetaApps (www.nsf.gov/pubs/2007/nsf07559/nsf07559.htm) encourage such research and open many opportunities for visualization researchers. In the following sections, I address two crucial research topics: parallel algorithms and data reduction methods.

Parallel Algorithms

Because all large-scale simulations run on parallel supercomputers, in situ visualization will require

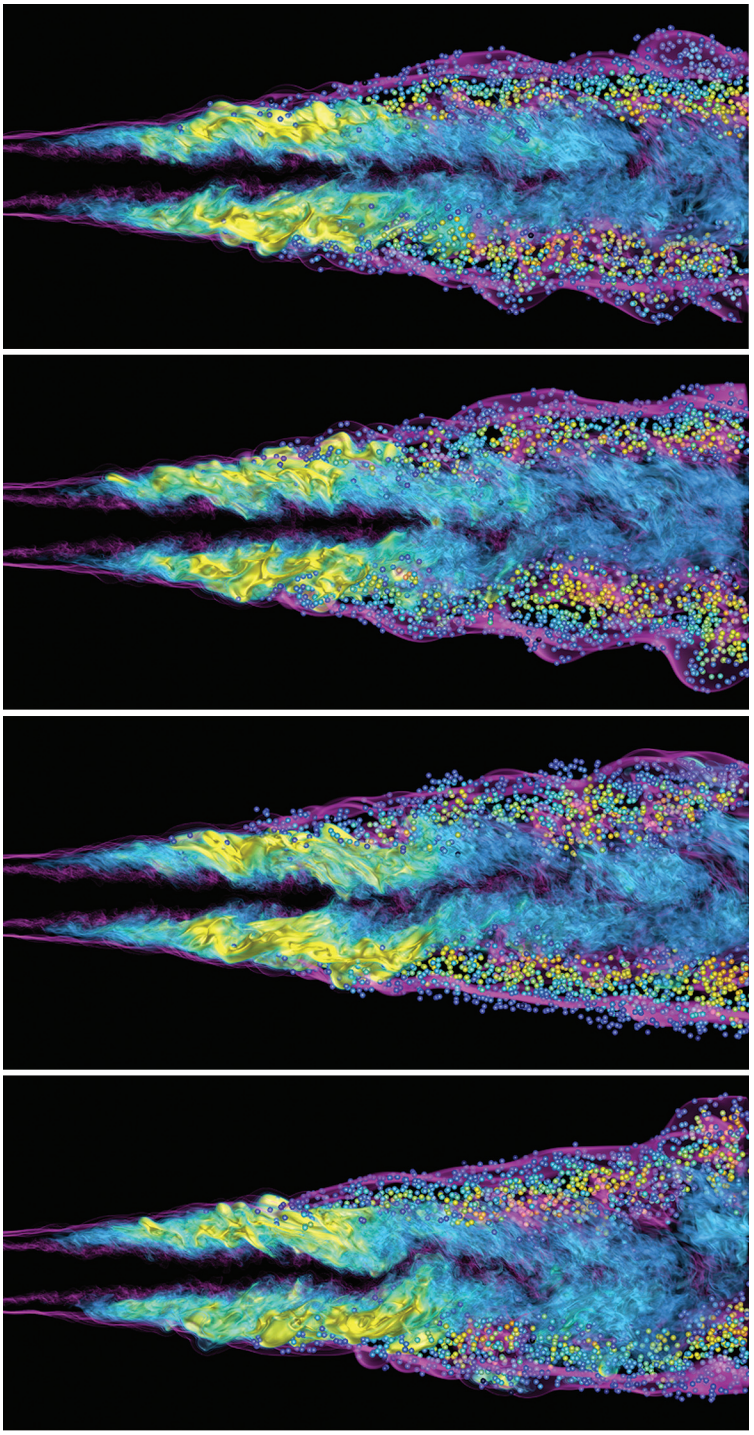

Figure 3. In situ visualization of a large-scale turbulent-combustion simulation running on 6,480 processors of a Cray XT5. Images of four selected time steps show the mixing of Y_HO_2 concentration and particle data colored on the basis of Y_OH. (Hongfeng Yu of Sandia National Laboratories and the SciDAC Institute for Ultrascale Visualization created the image; Jacqueline Chen of Sandia National Laboratories provided the data.)

parallel algorithms for both data reduction and rendering. Owing to the increasing scale of the hardware, simulations, and output data, these algorithms must be highly scalable. Some previously developed parallel visualization algorithms are no longer usable. For example, parallel image-compositing algorithms such as binary swap and SLIC (Scheduled Linear Image Compositing) scale up to only hundreds of processors. Although some recently refined algorithms have shown improved performance using thousands of processors,[2] petascale computing involving several hundred thousands of processors demands new, more scalable algorithms.

Scalable parallel in situ visualization algorithms are challenging to design for four reasons. First, as I alluded to earlier, the visualization algorithms should use the same domain decomposition made by the simulation to avoid data duplication and interprocessor communication as much as possible. However, even if these algorithms use the same domain decomposition, most of them still require duplicating data at partition boundaries. The resulting memory overhead must be minimized. For in situ processing, communicating data as needed rather than replicating data up front likely gives better results.

Second, for computations using tens of thousands of processors, interprocessor communication, if not carefully managed, would become a bottleneck, making the associated visualization solution unacceptable. An effective approach seems to be precomputing an optimal schedule for packing and ordering communication.

Third, the visualization computation should take a small fraction of the overall simulation time. Although sophisticated visualization methods offer visually compelling results, they're often unacceptable because they use too much simulation time. A plausible direction is to use domain knowledge whenever possible to reduce computation by processing only a small subset of the data. On the other hand, as in situ visualization technology becomes more mature and its benefits become substantial, scientists will be willing to trade simulation time for more analysis. In fact, in situ visualization suggests that scientists rethink the process of computational scientific discovery.

Finally, for volume rendering, how do we derive appropriate color and opacity transfer functions to best characterize the modeled phenomena in the data's spatial and temporal domains? Clearly, we must develop an adaptive method without constantly acquiring global information about those domains.

Parallel algorithms are also needed for more demanding visualization tasks. In many applications, it's desirable to visualize and understand the intrinsic interaction between different variables

over time. A simulation might involve tens or even hundreds of variables and chemical species. Such multivariate visualization is thus best done in situ because all relevant data are available. In situ visualization lets scientists visualize the full extent of their data in combination and at a fidelity and density not possible with postprocessing methods.

Figure 4 shows simultaneous visualization of three variables from the turbulent-combustion simulation shown in Figure 3. Figure 5 shows simultaneous pathline visualization and volume rendering of the magnitude of angular momentum from a supernova simulation. This visualization was based on two vector fields: velocity and angular momentum. The corresponding animation lets scientists see how the bundles of counter-rotating flow lines interact with the horizontally moving shock. Such visualizations are costlier because they require six times more data as well as calculations that are difficult to parallelize. The visualization in Figure 3 was created in an offline postprocessing step using a scalable parallel pathline visualization algorithm,[3] but that parallel algorithm isn't suited for in situ visualization. We must develop either a clever way to trace particles in situ or a way to depict vector fields that will scale with the supercomputer's size.

Data Reduction Methods

For petascale simulations, reducing data in situ is appealing because it can benefit all subsequent data movement and processing.

We might achieve data reduction with compression or by extracting features in the data. What exactly is a feature? Different disciplines clearly carry different definitions of features. Generally speaking, a feature is a small subset of the raw data isolated with domain knowledge and represents a particular physical structure, pattern, or event of interest. Examples include vortices, shocks, eddies, and critical points. We can categorize these features and use them to characterize and break down the overall modeled physical phenomenon.

The savings in storage space with feature extraction with or without contextual information can be significant, much greater than with compression-based methods. However, scientists don't always know completely what to extract and track in their data. This makes in situ feature extraction a particularly challenging problem because discarded data can't be retrieved to recover a feature without rerunning the simulation.

Basic visualization algorithms exist for feature identification, extraction, and tracking, which incorporate principles from image processing, computer vision, and machine learning. Feature

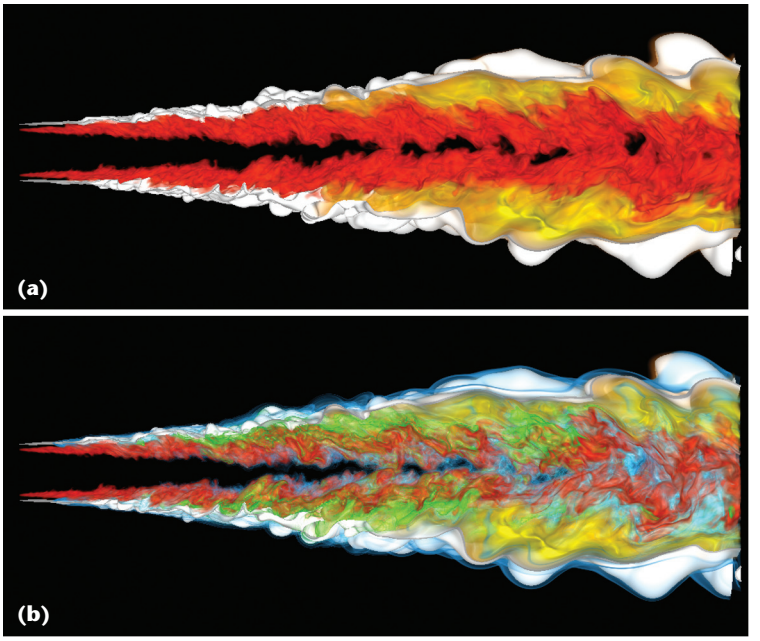

Figure 4. Simultaneous visualization of temperature, mixture fraction, and fuel concentration in isosurfaces from a turbulent-combustion simulation. (a) A temperature and mixture fraction isosurface represents the most active reaction layer. (b) Adding fuel concentration to the picture shows how it relates to temperature change and the reacting flow. (Hongfeng Yu created the image; Jacqueline Chen provided the data.)

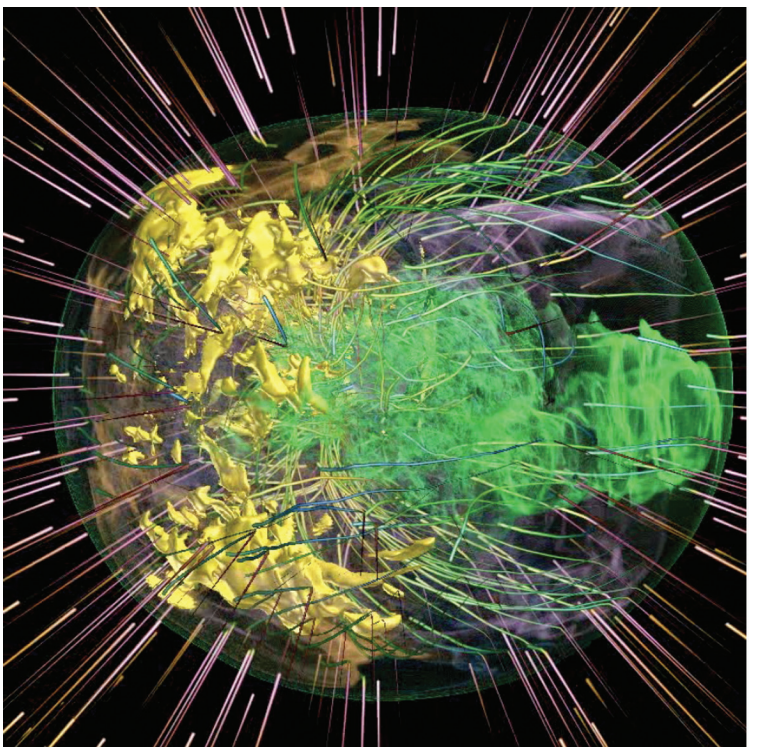

Figure 5. Simultaneous pathline visualization and volume rendering of the magnitude of angular momentum from a supernova simulation. Such visualizations are desirable, but generating them in situ with the simulation requires scalable parallel algorithms that aren't yet available. (Hongfeng Yu created the image; John Blondin at North Carolina State University provided the data.)

extraction and tracking is a common approach to visualizing time-varying flow data.[4] For tracking features through time, they must be correlated across a sequence of time steps. This task is frequently called the *correspondence problem*. Many algorithms perform feature correspondence on the basis of whether the features' regions overlap in adjacent time steps. Others perform feature correspondence on the basis of region attributes such as position, size, shape, orientation, or topology. All these algorithms must make special cases for regions that split or merge over time. They also must be careful with features that move far enough between time steps such that their regions don't overlap in consecutive time regions.

For petascale simulations, reducing data in situ is appealing because it can benefit all subsequent data movement and processing.

A novel, more universally applicable approach is to employ learning-based or evolutionary methods. The basic idea is to capture scientists' domain knowledge through interactive visualization augmented with an intelligent system and then apply that knowledge to feature extraction and tracking. The results should improve as knowledge accumulates over time. However, this approach's robustness hasn't been determined.

Because scientists typically store only one out of every few hundred time steps owing to storage space limitations, the resulting temporal gaps compound the problem of feature tracking. A longer interval between time steps means that a feature must move farther enough to be trackable. So, it's beneficial to perform feature extraction and tracking at simulation time, so that the tracking algorithm can utilize all the time steps that would otherwise be skipped or discarded during postprocessing.

Although scientists need to understand that in situ visualization's benefits come with a certain cost, that cost can be reduced. For example, tracking should use information from the smallest-possible time interval. Consequently, correspondence-based feature-tracking algorithms wouldn't work well because they require calculating regions in each time step independently before performing any tracking. In addition, the storage overhead of these algorithms and those based on machine learning is often too high to make them suitable for in situ processing.

A new prediction-correction algorithm for tracking volumetric features shows great promise.[5] It makes the best guess of the feature region in the next time step, then grows and shrinks the predicted region's border to coherently extract the feature of interest. It uses the data's temporal and spatial coherency to accelerate extraction while implicitly solving the tedious correspondence problem. Most important, this algorithm is straightforward to parallelize. We must develop similar algorithms for visualizing other types of features. In situ feature visualization can guide scientists to set the appropriate level of data reduction, leading to better data throughput for simulations.

If data reduction must come from subsetting or feature extraction, the corresponding information loss must be conveyed to the users. Quality assessment thus is crucial in large-scale data analysis and visualization because in many cases scientists use the reduced version of the data, rather than the original version, for evaluating the simulation and modeled phenomena. To compare the reduced or distorted data's quality, identifying and quantifying the loss of quality is imperative. Most data quality metrics, such as the mean square error and the peak signal-to-noise ratio, require access to the original data. Although computing these metrics is easy, they don't correlate well with perceived-quality measurement.[6] More important, these metrics don't apply to petascale applications simply because the original data are too large to acquire or compare efficiently.

One viable approach is to compute feature information in situ.[7] Then, we can use it as the basis metric for offline quality assessment without needing to access the original data. We can achieve this by computing the distance of feature components of the reduced or distorted data from those of the original data. This will give us an indication of quality loss in relation to the original data. Such feature information also lets us perform cross-comparison of data with different reduction or distortion types.

Finally, to enable interactive data analysis and visualization of the reduced data, efficient organization of the data to facilitate fast access is key. Data subsets must be indexed rather than stored as flat files.[8]

High-performance computing systems running at sustained petaflop speeds are becoming increasingly available for scientists and engineers

to perform petascale and exascale simulations. To possibly understand such extreme-scale simulations and extract meaning from petabytes of simulation output, simulation scientists and visualization researchers must begin working closely together so that a viable solution will be ready. Otherwise, much of the value of petascale and exascale simulations will go unrealized.

In situ visualization is clearly a promising solution for ultrascale simulations. We've seen some success in realizing this solution[1,9] and in ongoing efforts to add support for in situ visualization to open source visualization toolkits such as ParaView and VisIt. However, for others to adopt this approach, we need further research and experimental studies to derive a set of guidelines and usable visualization software components. If this research is successful, it will lead to a new visualization and data-understanding infrastructure, potentially change how scientists work, and accelerate scientific discovery.

References

1. H. Yu et al., *A Study of In Situ Visualization for Petascale Combustion Simulations*, tech. report CSE-2009-9, Dept. of Computer Science, Univ. California, Davis, Apr. 2009.
2. H. Yu, C. Wang, and K.-L. Ma, "Massively Parallel Volume Rendering Using 2-3 Swap Image Compositing," *Proc. 2008 ACM/IEEE Conf. Supercomputing* (SC 08), ACM Press, 2008, article 48.
3. H. Yu, C. Wang, and K.-L. Ma, "Parallel Hierarchical Visualization of Large Time-Varying 3D Vector Fields," *Proc. 2007 ACM/IEEE Conf. Supercomputing* (SC 07), ACM Press, 2007, article 24.
4. F.H. Post et al., "The State of the Art in Flow Visualisation: Feature Extraction and Tracking," *Computer Graphics Forum*, vol. 22, no. 4, 2003, pp. 185–197.
5. C. Muelder and K.-L. Ma, "Interactive Feature Extraction and Tracking by Utilizing Region Coherency," *Proc. 2009 IEEE Pacific Visualization Symp.*, IEEE CS Press, 2009, pp. 17–24.
6. N. Sahasrabudhe et al., "Structured Spatial Domain Image and Data Comparison Metrics," *Proc. 10th IEEE Visualization Conf.* (VIS 99), IEEE CS Press, 1999, pp. 97–104.
7. C. Wang, H. Yu, and K.-L. Ma, "Importance-Driven Time-Varying Data Visualization," *IEEE Trans. Visualization and Computer Graphics*, vol. 14, no. 6, 2008, pp. 1547–1554.
8. M. Glatter et al., "Visualizing Temporal Patterns in Large Multivariate Data Using Textual Pattern Matching," *IEEE Trans. Visualization and Computer Graphics*, vol. 14, no. 6, 2008, pp. 1467–1474.
9. T. Tu et al., "From Mesh Generation to Scientific Visualization: An End-to-End Approach to Parallel Supercomputing," *Proc. 2006 ACM/IEEE Conf. Supercomputing* (SC 06), IEEE CS Press, 2006, p. 12.

***Kwan-Liu Ma** is a professor of computer science at the University of California, Davis. He directs the US Department of Energy SciDAC (Scientific Discovery through Advanced Computing) Institute for Ultrascale Visualization. Contact him at ma@cs.ucdavis.edu.*

Contact department editor Theresa-Marie Rhyne at theresamarierhyne@gmail.com.