

Advanced Visualization Technology for Terascale Particle Accelerator Simulations

Kwan-Liu Ma* Greg Schussman Brett Wilson

University of California at Davis

Kwok Ko

Stanford Linear Accelerator Center

Ji Qiang Robert Ryne

Lawrence Berkeley National Laboratory

Abstract

This paper presents two new hardware-assisted rendering techniques developed for interactive visualization of the terascale data generated from numerical modeling of next-generation accelerator designs. The first technique, based on a hybrid rendering approach, makes possible interactive exploration of large-scale particle data from particle beam dynamics modeling. The second technique, based on a compact texture-enhanced representation, exploits the advanced features of commodity graphics cards to achieve perceptually effective visualization of the very dense and complex electromagnetic fields produced from the modeling of reflection and transmission properties of open structures in an accelerator design. Because of the collaborative nature of the overall accelerator modeling project, the visualization technology developed is for both desktop and remote visualization settings. We have tested the techniques using both time-varying particle data sets containing up to one billion particles per time step and electromagnetic field data sets with millions of mesh elements.

Keywords: hardware-assisted techniques, high-performance computing, particle accelerators, perception, point-based rendering, scientific visualization, field lines, texture mapping, time-varying data, vector field visualization, visual cues, volume rendering

1 Introduction

Particle accelerators have helped enable some of the most remarkable discoveries of the 20th century. They have also led to substantial advances in applied science and technology, many of which greatly benefit society. Accelerator-based systems have now been proposed to address problems of international importance related to energy and the environment. Given the importance of particle accelerators, it is imperative that the most advanced high-performance computing tools be brought to bear on their design, optimization,

technology development, and operation. The SciDAC accelerator modeling project (<http://scidac.neresc.gov/accelerator/>) is a national research and development effort whose primary objective is to establish a comprehensive terascale simulation environment needed to solve the most challenging problems in 21st century accelerator science and technology. This simulation environment will enable accelerator physicists and engineers across the country to work together using a common set of scalable, portable, interoperable software to solve the most challenging problems in accelerator design, analysis, and optimization. Consequently, a critical component of this project is the development of adequate visualization technology which will allow project members to better interpret, verify, and communicate with others the terabytes of simulation data routinely generated.

The three-dimensional, nonlinear, multi-scale, many-body aspects characteristic of future accelerator design problems drive the essential requirement for terascale simulation [3]. Examples include finding the eigenmodes in extremely large and complex 3D electromagnetic structures, predicting the halos of high-intensity beams, and simulating the particle and field dynamics in plasma-based accelerators. The resulting simulation data present unprecedented visualization challenges in terms of both size and complexity. This paper presents our initial success in addressing the challenges of visualizing high-intensity particle beam data and complex 3D electromagnetic field data.

All the simulation codes currently run on parallel computers operated at NERSC/LBL, ACL/LANL, and SLAC. Beam dynamics simulations [10, 11] with up to 500 million particles have been performed, a number that is approaching the needed resolution in the calculation of the beam density to predict beam halo in next-generation accelerator designs. Electromagnetic calculations for the design of complex beamline components and systems have achieved the required accuracy for modeling the cavities in the accelerator structure [16]. Each simulation run can generate terabytes of data.

In addition to the large-data problem which is being addressed by high-performance computing [4, 5, 6, 9], two fundamental visualization problems must be solved. One is the problem of visualizing very dense point data (i.e. particles),

* {ma,schussma,wilson}@cs.ucdavis.edu

kwok@slac.stanford.edu

{JQiang,RDRyne}@lbl.gov

and the other is visualizing very dense line data (i.e. electric and magnetic field lines). We have developed a novel hybrid rendering approach addressing the particle density problem, and a scalable texture-enhanced representation for better illustrating otherwise hidden field properties. While these two problems are unique to accelerator physics data, the techniques we introduce here are applicable to any applications concerned with the visualization of particle data and field line data. We have also exploited the features of the new generation of commodity graphics cards like the *nVidia GeForce* series to accelerate the rendering.

2 Particle Beam Data

Particle accelerator simulations model a large number of charged particles as they move through the accelerator and respond to various forces [11]. The resulting data sets consist of hundreds of millions to billions of particles for each time step, making it impossible to render in real-time, or even to fit into the memory of most PCs. One approach is to convert the particles to volumetric data representing point density, and use texture-mapping hardware to render to the screen [8]. However, the size of volumes that can be efficiently visualized in this manner are limited by the amount of available texture memory, as well as the fill rate of the available hardware. In addition, high-resolution representations present challenges with regard to the available network bandwidth, disk space, and the time required to process the data. As a result, even though this approach does give good interactivity and compact data size, many fine details can be lost, especially in the very low-density areas where scientists are most interested. We have developed a hybrid data storage and rendering method which allows scientists to visualize and explore the data at interactive rates while maintaining much of the important detail of the original data.

2.1 Hybrid Rendering

Our hybrid technique leverages the speed of texture-based hardware volume rendering to represent large features, and the flexibility of point-based rendering to represent fine details. The foundation of the hybrid method is the use of low-resolution volume rendering in the areas of low interest/detail, and the use of point-based methods to enhance or replace areas of high interest/detail. Thus, storage, transfer, and rendering resources are put to more efficient use than with volumetric or particle rendering alone.

The interactivity offered by the hybrid method makes choosing viewing parameters and transfer functions for subsequent higher-quality rendering an easy job, and the storage savings mean that the data can be more efficiently transferred from the computer where it was generated to a remote computer on a scientist’s desk thousands of miles away.

We tested this approach on data obtained from several large-scale beam dynamics simulations. Each particle in

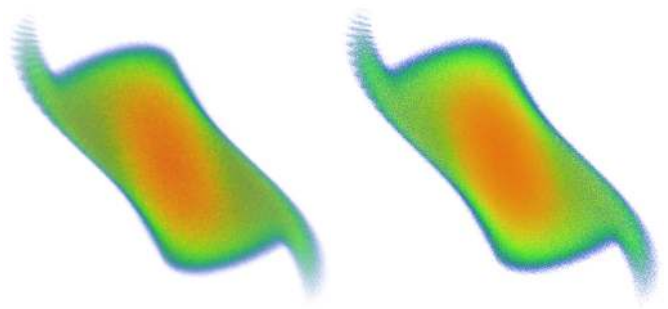


Figure 1: Comparison of a volume rendering (left) and a mixed volume/point rendering (right) of the phase plot (x, P_x, y) of frame 170. The volume rendering has a resolution of 256^3 . The mixed rendering with a volumetric resolution of 64^3 and 2 million points provides more detail than the volume rendering while displaying at a much higher frame rates.

these simulations consists of spatial coordinates (x, y, z) and momenta (p_x, p_y, p_z) in double-precision. The primary simulation, consisting of 100 million particles, requires 5GB of storage per time step. An additional data set, the initial time step of a billion point simulation, requires 48 GB of storage. These sizes make data impractical to move, and impossible for most computer to handle.

Figure 1 shows a comparison of a standard volumetric rendering, and a mixed point and volumetric rendering of the same object. The mixed rendering is able to more clearly resolve the horizontal stratifications in the right arm, and also reveals thin horizontal stratifications in the left arm not visible in the volume rendering from this angle. Note that the bands near the edges are part of the data, not rendering artifacts.

Images for four different distributions including (x, y, z) , (x, p_x, y) , (x, p_x, z) , and (p_x, p_y, p_z) of the data at time step 180 are displayed in Figure 2. The simulation corresponds to an intense beam propagating in a magnetic quadrupole channel. The beam propagates in the z -direction, with focusing provided in the transverse (x and y) directions.

2.2 Point Selection Criteria

In order to construct a hybrid representation, we must decide how to classify points (i.e. particles) as being rendered directly or simulated via volume rendering. For this data set, the most detailed and important area to visualize is the very low-density beam halo [10]. This area poses additional problems for volumetric representation because it is thousands of times less dense than the beam core, making it hard to compute the precise density for a given region, and to assign that density one of a limited number of palette values.

Therefore, we choose points in areas of low density to render directly, and the remaining areas of high density are rendered using fast low-resolution volume rendering. This

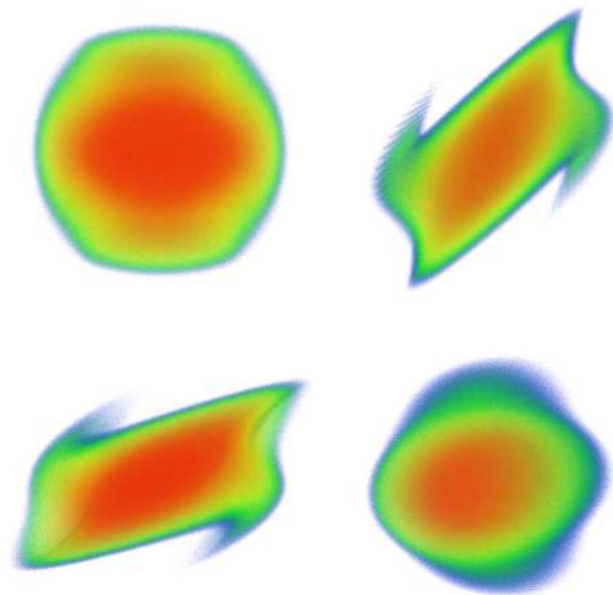


Figure 2: Selected distributions for time step 180. Top: (x, y, z) and (x, P_x, y) . Bottom: (x, P_x, z) and (P_x, P_y, P_z) .

allows the fine detail of the beam halo to be accurately represented at the full data resolution, while maintaining interactivity by reducing the amount of data transferred and rendered.

2.3 Preprocessing

The hybrid representation of the data is computed on the same parallel supercomputer at NERSC/LBL that generated the original simulation: an IBM SP RS/6000 with 2,944 processors. Preprocessing consists of two steps: partitioning and extraction. Partitioning is a one-time process that adds structure to the originally unstructured particle data. Extraction is a fast process that quickly extracts a hybrid representation with given parameters from the partitioned data.

The partitioning program organizes the unstructured point data into an octree. It is provided a time-step number, a plot type (since there are six parameters per point, there are a variety of 3-D plots that can be generated), and a maximal subdivision level. It then reads in all the points and inserts them into an octree. The levels of subdivision of the octree are limited by the maximal subdivision level, which prevents the octree from becoming impractically large. This octree is written out to disk in two parts: one part contains all the particles of the simulation, the other contains the octree nodes themselves. In the particle files, particles in the same octree node are grouped together, and the groups are sorted in order of increasing density. Each node in the octree then contains an offset into the particle file and the the number of particles in its group.

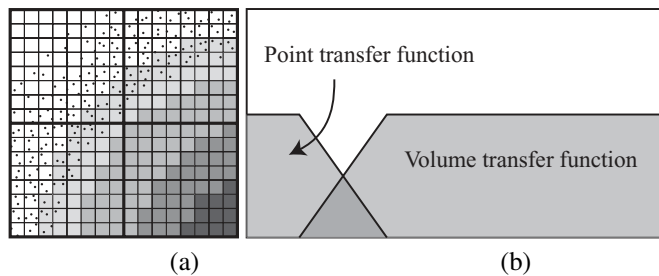


Figure 3: A hybrid data representation. (a) An image is created by classifying each octree node as belonging to a volume-rendered region or a point-rendered region, depending upon the transfer function for each region (the regions can overlap, as in this example). The combination of the two regions defines the output image. (b) The relationship between the two transfer functions. The two transfer functions can be edited together or separately.

The partitioning program takes about 7 minutes per time step for the 100 million particle simulation. Since it is primarily I/O bound, processing time scales linearly as the number of points increases. If the data exceeds the amount of memory available on one node of the supercomputer, it can also be run on multiple nodes: the volume is divided up between nodes and particles are assigned to the corresponding node once they are read from disk. Since the partitioned representation contains all the data present in the original representation, it is possible (although not yet implemented) to discard the original data and convert between different plot type partitionings.

The extraction program converts the partitioned data into the hybrid representation. It is given a partitioned frame and a threshold density. Particles in octree nodes below the threshold density are stored in the hybrid representation. All other points (those in the higher-density regions) are discarded (see Figure 3). To accomplish this, the extraction program reads in the octree and determines which nodes should contain stored points. Since the particle file is sorted in order of increasing density, all particles required for any hybrid representation are in a contiguous block at the beginning of the file. This portion of the particle data is just copied to the output; no computation is necessary for the particles, and discarded particles are never read from disk.

The threshold density parameter provided to the extraction program allows the user to balance file size and visual accuracy for a given application. A high threshold value will yield large file sizes but larger areas of the rendering can be drawn using the more accurate point-rendering method. A low threshold value will yield smaller file sizes appropriate for viewing multiple frames simultaneously, or quickly transferring over a network, at the expense of having a thinner halo region representable by points. Because the extraction process is fast, different hybrid representations can be created and discarded as needed.

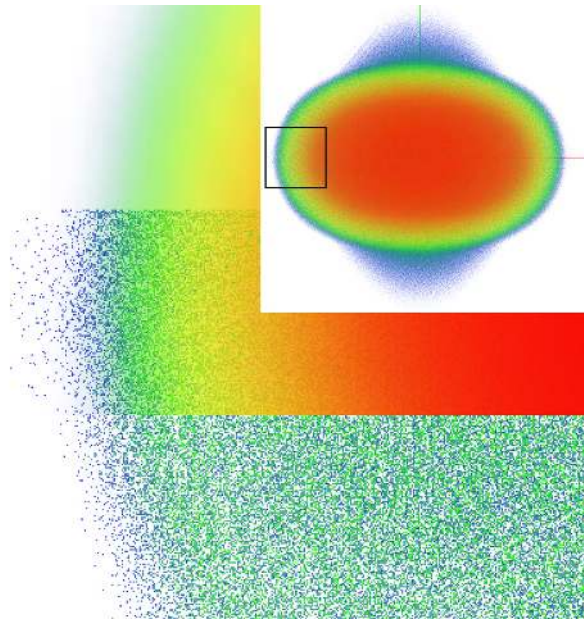


Figure 4: Portions of a hybrid rendering on a sphere-like (x, y, z) distribution showing (top) the volume-rendered portion, (middle) the combined hybrid rendering, and (bottom) the point-rendered portion alone. The front half of the sphere has been clipped; the points obscured by the volume rendering are on the far side. The points shown here are completely opaque so they are more visible.

2.4 Viewing

A separate view program with an interactive transfer function editor is used on a desktop PC to visualize the partitioned data generated by the parallel computer. The volume transfer function (see Figure 3 (b)) maps point density to color and opacity for the volume-rendered portion of the image. Typically, a step function is used to map low-density regions to 0 (fully transparent) and higher density regions to some low constant so that one can see inside the volume. The program also allows a ramp to transition between the high and low values, so the artificial boundary of the volume-rendered region is less visible.

The point transfer function (see Figure 3 (b)) maps density to number of points rendered for the point-rendered portion of the image. Below a certain threshold density, the data is rendered as points; above that threshold, no points are drawn. Intermediate values are mapped to the fraction of points drawn. When the transfer function's value is at 0.75 for some density, for example, it means that three out of every four points are drawn for areas of that density. This allows the user to see fewer points if too many points are obscuring important features, or to make rendering faster. It also allows a smooth transition between point-rendered portions of the image and non-point-rendered portions. Figure 4 displays parts of a hybrid rendering.

By default, the two transfer functions are inverses of each

other. Changing one results in an equal and opposite change in the other. This way, the user can change the boundary between the volume- and the point-rendered regions of the image (up until the boundary specified during preprocessing, beyond which no points are available).

2.5 Results

The hybrid beam rendering method is effective for a variety of simulation configurations and visualization requirements. The user can tailor the hybrid output to range from large, very accurate representations to small, less accurate representations (that still preserve as much interesting data as possible).

The hybrid method can produce very compact representations, allowing multiple time steps to fit into memory. Reasonably high-quality pictures can be made with hybrid data smaller than 100MB, so a high-end PC is capable of holding around 10 time steps in memory at once. The previewing program allows the user to step through frames using the keyboard. If a frame is already in memory, it can be displayed instantaneously: the volume texture and display lists are already loaded into video memory, or can be quickly swapped in by the display driver. If a frame is not in memory, it is loaded from disk, a process that takes around 10 seconds for a 100MB time step. This allows very efficient exploration of the beam's evolution over time; if the step size is small enough, individual particles can be seen moving between frames.

Figure 5 displays selected frames from a simulation over 350 time steps for the (x, y, z) distribution of the data. All frames use the same view looking down z , the beam's axis. The quadrupole magnets are alternately focusing and defocusing in the x and y directions, resulting in the four-fold symmetry seen in the figure. At this resolution, each time step is about 500MB, allowing only two to fit into memory at once. However, hybrid frames are often smaller; these use a conservative point density threshold.

In addition to scaling in the time dimension, the hybrid algorithm also scales well in terms of simulation sizes. Because the output data size does not necessarily depend on the input data size, large simulations approaching 1 billion particles can be reduced to the same size hybrid representation as the smaller simulations. The large simulation's point-based halo region will be thinner than the smaller simulation, but that has little effect on the quality of the resulting image: regardless of the simulation size, points at the high-density halo cutoff region are typically so dense that they visually merge into a volume anyway.

One important effect that occurs in larger simulations is that the octree must be subdivided more finely where there is a high gradient. This occurs both in very large simulations and in smaller simulations with very focused beams. If a higher level of subdivision is not used, the outline of the lowest level octree nodes will be visible at the boundary of the halo region. For low gradients, a shallower depth of octree

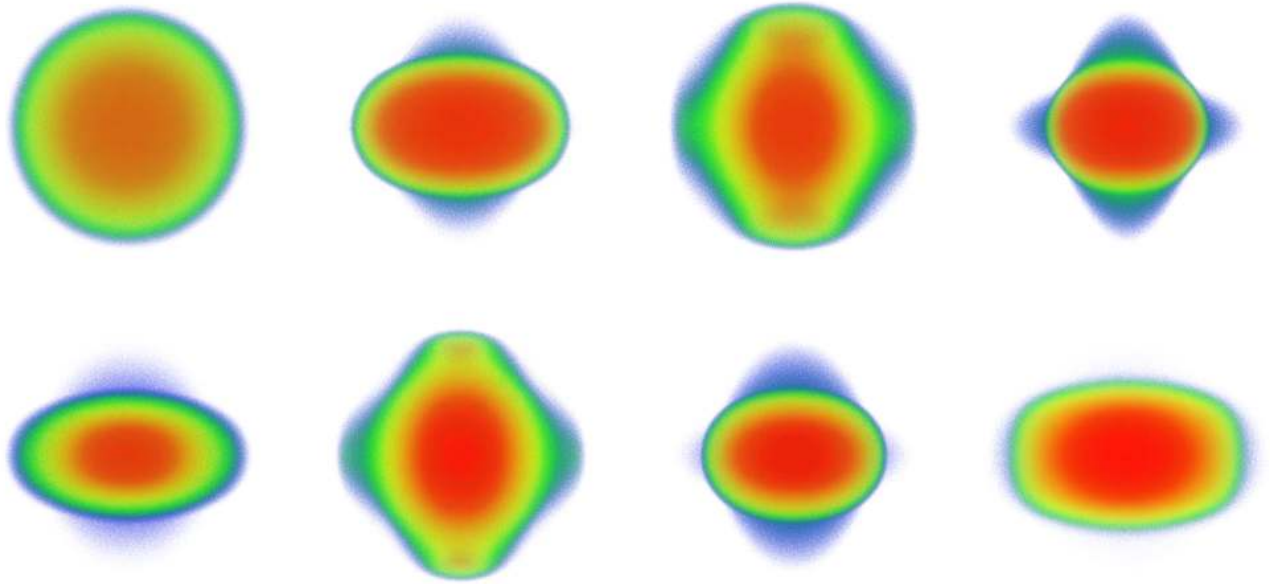


Figure 5: Selected time steps from an simulation over 350 time steps for the x-y-z distribution of the data. Top: frames 1, 50, 100, 150. Bottom: frames 200, 250, 300, 350.

subdivision can be used without introducing significant artifacts, saving valuable space.

For visualizing the particle beam data, volume rendering lacks the spatial resolution and the dynamic range to resolve regions with very low density, areas which may be of significant interest to researchers. Point-based rendering alone lacks the interactive speed and the ability to run on a desktop workstation that the hybrid approach provides. Furthermore, point-based rendering for low-density areas provides more room for feature enhancements. Because points are drawn dynamically, they could be drawn (in terms of color or opacity) based on some dynamically calculated property that the scientist is interested in, such as temperature or emittance. Volume-based rendering, because it is limited to pre-calculated data, cannot allow dynamic changes like these.

3 Electric and Magnetic Field Data

The other simulation code we are working with is based on a parallel time domain electromagnetic field solver using unstructured hexahedral meshes. This code models the reflection and transmission properties of open structures in an accelerator design [16]. To achieve the needed accuracy, the simulations must not proceed faster than electromagnetic information could physically flow through mesh elements. To satisfy the *Courant Condition*, simulating 100 nanoseconds in the real world requires millions of time steps. The parallel simulation code is scalable in terms of both the the number of mesh elements and the number of particles. Each run of the simulation on the SLAC's 32-node PC cluster can produce

terabytes of data.

A scalable solution is required for visualizing such large and complex electromagnetic fields. The main challenge is concerned with displaying a dense collection of intertwined lines in a way that shows clear spatial relationships between them, with unambiguous global or local detail. We believe that interactivity is the key to insightful visualization, so our approach is based on a compact representation for field lines coupled with hardware-assisted perceptually effective rendering. The result is highly interactive visualization facilitating understanding of both the global and local structures of the electromagnetic fields.

3.1 A Compact Representation for Field Lines

The problem of drawing lines to show the structure of vector fields has been studied extensively. Work has also been done to use alternative representation of lines like tubes and ribbons to improve perception of their structures or additional physical properties of the data. We have developed a flexible and scalable representation which we call self-orienting surfaces for illustrating field lines [12]. This representation using hardware texturing can conveniently display the field properties as lines, tubes, or ribbons.

Each self-orienting surface is a triangle strip which is constructed from a sequence of points along a curve, an associated sequence of tangent vectors, and a viewing position. The triangle strip always orients toward the observer which makes aligning a texture to the strip easy. For example, the tube-like appearance is made possible by using hardware

accelerated bump mapping. Compared to polygonal tubes, self-orienting triangle stripes are much more compact, resulting in significant saving in storage and rendering.

Self-orienting triangle strips cooperate well with texturing. Because the strips orient themselves in a view-dependent way, the texture coordinates for moving across the strip become view-independent. Difficulties that can occur with polygonal tubes are avoided. Figure 6 (a), (b), and (c) show conventional line drawing, illuminated field lines, and streamtubes, respectively, for illustrating both the electric field and magnetic field inside a 3-cell linear accelerator structure. As shown in Figure 6 (d), the self-orienting triangle strips rendered with hardware bump mapping give similar visual effect while using only a very small number of triangles, about five to six times less than a typical streamtube representation would require.

As shown in Figure 6 (e), using a wider version of the self-orienting surfaces it is possible to give the impression of the field density by only rendering a small number of self-orienting surfaces, with line density textured according to local field strength. The reduction in the number of lines that must be traced and plotted can help maintain a desirable level of interactivity.

3.2 Seeding Strategy and Incremental Visualization

A key task in field line visualization is the selection of seed points for streamline integration. Much work has been done [2, 7, 14] for providing aesthetically pleasing streamlines through careful selection of seed points. The emphasis is generally on producing a visually uniform density of streamlines in the final image. Our approach is to select seeds so that the local density anywhere in the final distribution of field lines is approximately proportional to the local magnitude of the underlying field. When this approach is applied to electromagnetic fields, the resulting image is intuitive for physicists, because the densities of electric and magnetic flux lines are proportional to the corresponding field strength.

The implementation of our seeding strategy consists in computing a desired average number of field lines to pass through each element of the mesh. This is the average field intensity from at the element's vertices multiplied by the volume of the element. These numbers are then scaled so that the sum over all elements is equal to total maximum number of field lines to pre-integrate. The algorithm consists of selecting the element which most needs an additional field line, picking a random seed point within that element, and integrating the field line from there. During integration, as each new element is visited, that element's desired number of field lines is decremented. This selection and integration process repeats until the total desired number of field lines for the entire mesh has been obtained. By keeping track of how many field lines already pass through the elements, dis-

proportionately high densities of field lines are avoided. By always choosing the element that most needs an additional field line, the images that result from rendering the first n field lines are always nearly correct in showing field line density proportional to the magnitude of the underlying field.

This incremental approach addresses the challenge in presenting extremely dense collections of field lines. Although transparency effects also help address the challenge, they are only useful up to moderate field line density. At extreme densities, transparency effects result in images qualitatively similar to those produced by direct volume rendering. Simple direct volume rendering suffers from ambiguity in that perspective depth cues and lighting cues are missing, and because different combinations of thickness, opacity and coloration assigned to the data can composite to produce the same color and intensity in a final image. An interactive animation of our incremental approach avoids these ambiguities. By sweeping from a minimum to a maximum number of field lines, one gets a compelling sense of the structure and magnitude of the fields being built up. It is clear where the strong regions are, because sparse lines appear there first, and these lines have good perspective and lighting depth cues. As more field lines are added, the strong regions become more dense and the less strong regions start to fill in. Figure 7 shows selected images from such a sequence, with the lines corresponding to the highest magnitude field regions being loaded first. From there, progressively weaker field lines are loaded in. In each image, the density of field lines is approximately proportional to the magnitude of the underlying field. In this way, each image attempts to be the most accurate representation of field magnitude possible given the number of field lines used. The set of field lines in each image in the sequence is a superset of those field lines in the preceding image.

3.3 Perceptually Effective Visualization

In order to better understand a large number of intertwined field lines, perceptual issues cannot be neglected. Proper use of illumination, haloing, transparency and other visual cues can help clarify spatial relationships and reveal hidden information. We have incorporated perceptually effective enhancement methods into the self-orienting surface representation to increase the information level and clarity of the picture.

3.3.1 Illumination

While the illuminated field lines technique [13] can help determine the shading of a field line, it is less effective to accurately interpret the spatial relationships between similarly oriented adjacent or overlapping lines, as pointed out in [1]. In particular, thin lines could look artificial because the texture does not vary sideways across the width of the lines. Our illuminated triangle stripes offer not only improved visual clarity, comparable to the volume LIC approach in [1],

but also the critical interactivity needed for efficient data exploration. Figure 6 (f) demonstrates the effect of enhanced lighting. The enhanced lighting is hardware accelerated, and carries no significant performance penalty over a single light source.

3.3.2 Haloing

Adding halos can clarify the spatial relationships between overlapping lines. Our self-orienting surfaces representation is superior to the illuminated field lines with halos. The illuminated lines images do not provide a perspective depth cue, whereas the self-orienting surfaces do. At medium depth, a cross section of the haloed lines appears as one or two black pixels on either side of a few illuminated pixels. There is an abrupt transition from the black region to the illuminated region. This can be thought of as an approximation for Phong illumination of a tube with a headlight. The diffuse and specular components remain at the middle of the cross section because that is where surface normal, viewing, and light vectors all align. Assuming a small or non-existing ambient contribution, the cross section edges are dark because the surface normal is orthogonal to the viewing and lighting vectors. Our self-orienting surfaces use texture to effectively capture the same surface normal vectors that a polygonal tube would have, so for self-orienting surfaces the lighting appears exact.

At first glance, comparison of the two techniques at medium depth shows little difference. However, at near depth self-orienting surfaces look better. The perspective widening of the self-orienting surfaces provide a significant depth cue. If the widths of the haloed lines are scaled up to match, the sharp transition from black halo to illuminated region becomes very apparent. What was a reasonable approximation at several pixels wide becomes noticeably incorrect when scaled up. In contrast, self-orienting surfaces show even more clearly the Phong illumination model at work, providing a smooth and very convincing cross section.

3.3.3 Transparency

For very dense line data, as displayed in Figure 6 (g), it can be difficult to unambiguously perceive the details in a region in the interior of the 3D field. Surrounding lines, when sufficiently dense, can occlude the interior structures. One approach is to “cut away” the data which is not in the region of interest. While effective as shown in Figure 6 (h), in other cases this could take away the global context for the current region of interest. The other approach is to leave the region of interest opaque while using transparency to de-emphasize the remaining data. As a result, as shown in Figure 6 (i), the interior structures can remain clear, and the global context is not lost. Transparency in complex scenes requires back-to-front compositing for a correct image. Depth sorting is not practical for very large data. Our approach can be coupled with the order-independent transparency technique

supported on the nVidia GeForce 3 but would require disabling bump mapping and finer tessellation of self-orienting surfaces.

3.4 Results

Figure 8 shows images of four selected time steps from the simulation of the 3-cell structure. The ability to animate field lines in the temporal domain is particularly valuable. For example, from these four images, scientists can examine and verify the propagation of the RF waves. Storing the pre-computed field lines rather than the raw data can significantly cut down the data storage and transfer requirements making interactive interrogation of the time-varying electromagnetic field lines data possible. The typical saving is about a factor of 25, which would allow many time steps of electromagnetic field lines to reside in memory for interactive viewing. We are presently parallelizing the field line calculations on PC clusters to speed up this preprocessing task.

Figure 9 shows the inside of a 12-cell linear accelerator structure. The front half of the mesh has been removed to permit viewing inside. Electric field lines, shown in blue, originate and terminate at the surface of the mesh. Power flows in from the top and bottom through input ports, and then flows to the right. Charged particles, under the influence of the propagating field, would be accelerated from left to right. Ideally, the electric field should be perfectly radially symmetrical. However, the radial asymmetry in the geometry of the ports causes asymmetry in the electric field.

Note that simulation of this 12-cell structure reaches steady state at about 40 nanoseconds, which corresponds to 326,700 time steps. Since it would take about 80 megabytes of storage space to save one time step of the electric and magnetic fields together, over 26 terabytes of storage space would be needed for the overall data set. Storing the pre-integrated field lines instead and using the seeding strategy described make it possible for us to visualize the data.

The sequence of images in Figure 10 shows incremental loading of field lines as in Figure 7 but with line transparency and color assigned according to the field strength. The key is that the scientist is allowed to interactively change these visualization and viewing parameters, and then see the resulting visualization immediately.

4 Conclusions

The overall objective of this national research and development effort is to establish a comprehensive terascale simulation environment for use by the U.S. particle accelerator community. The effective visualization methods presented in this paper allow scientists to gain better insights into the simulation results and communicate with others their findings in a more intuitive way.

This paper discusses the visualization challenges of the terascale applications, introduces two novel visualization

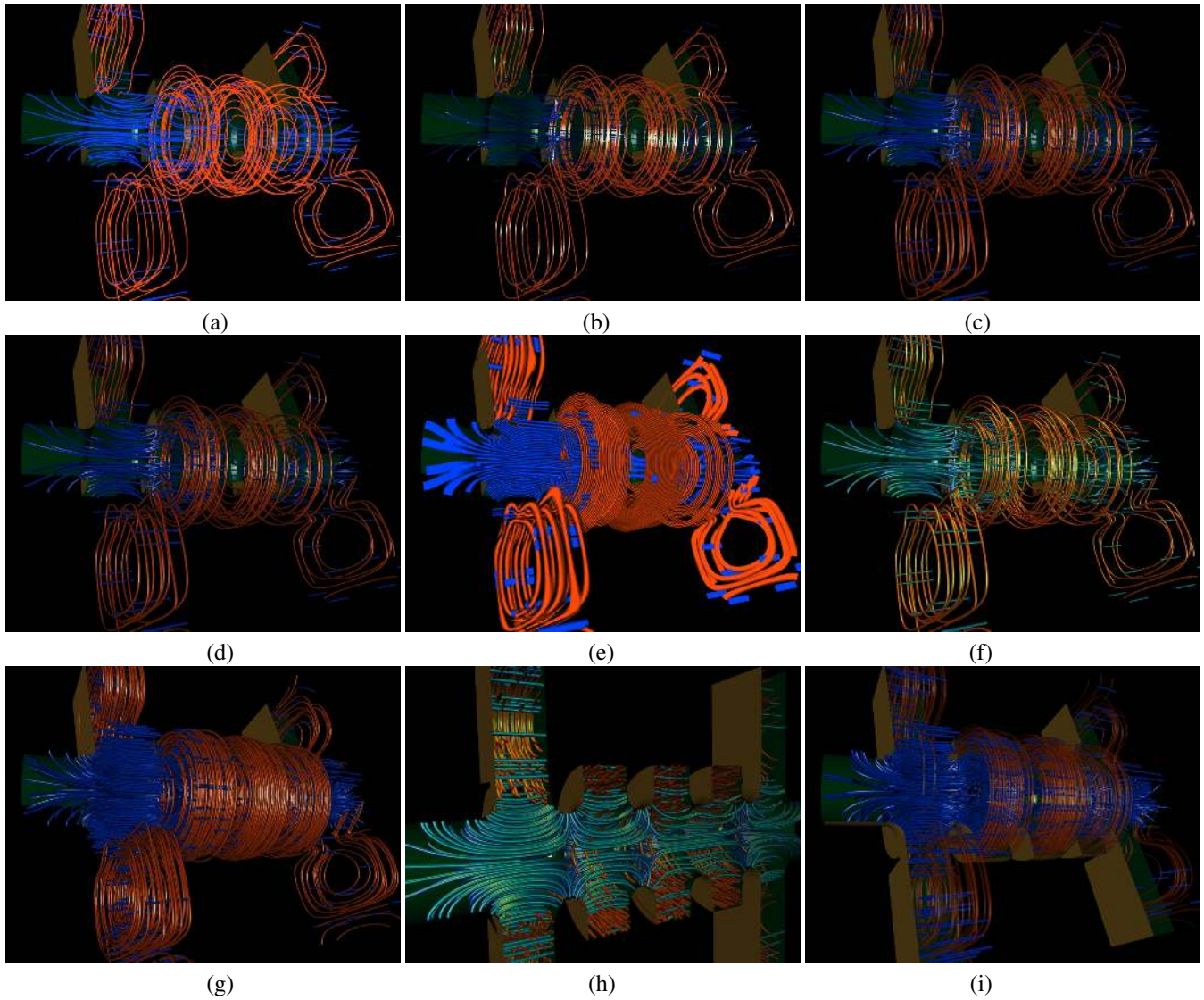


Figure 6: Visualization of electromagnetic field corresponding to a section of an accelerator structure. (a) conventional line drawing. (b) illuminated streamline technique. (c) conventional streamtube technique. (d) self-orienting surface technique. (e) compact textured ribbon representation. (f) with enhanced lighting. (g) dense lines. (h) cutting away. (i) use of transparency.

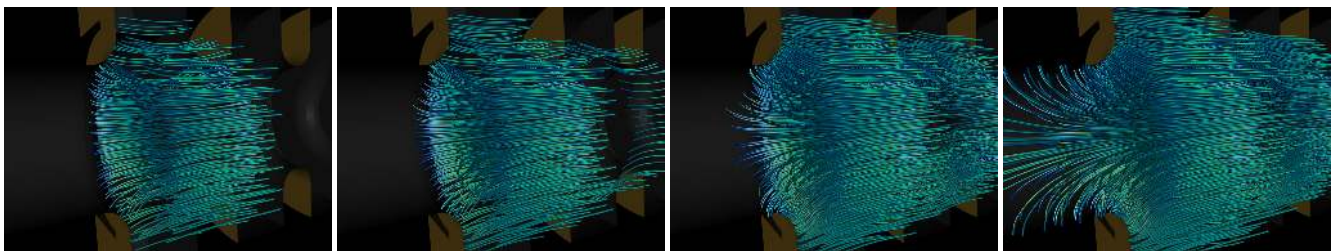


Figure 7: From left to right, incremental loading and rendering of electric field lines in a port and two cells of an accelerator structure.

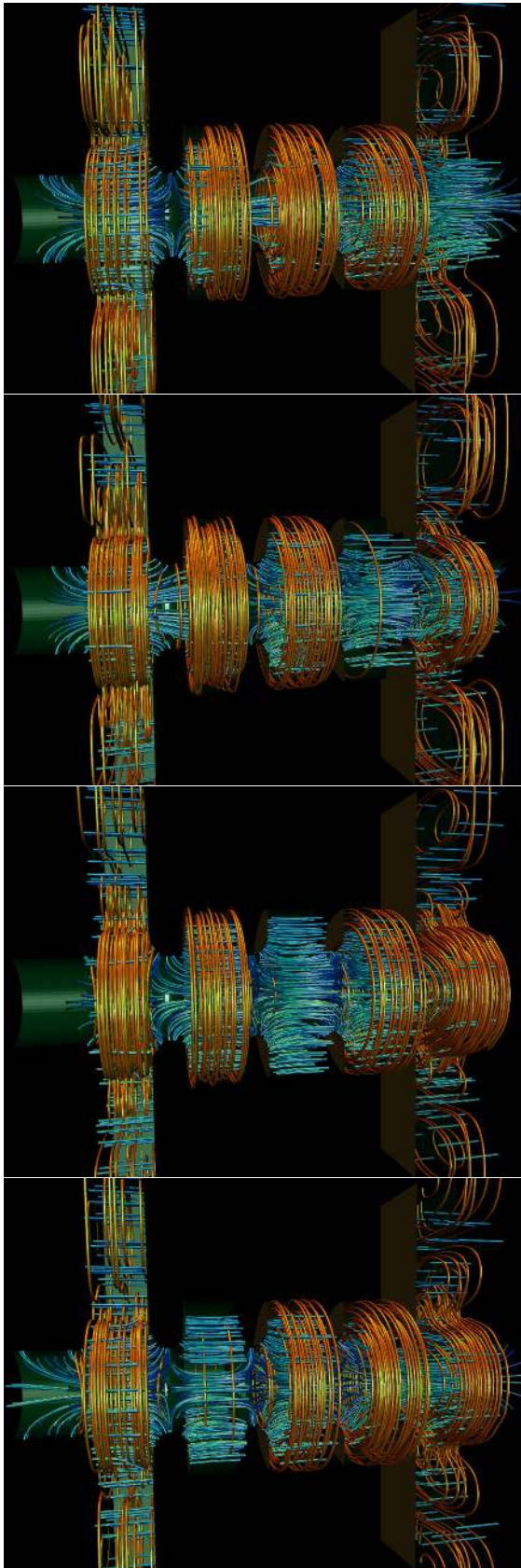


Figure 8: Selected time steps which show RF waves propagate in through the input ports (left) and out through the output ports (right).

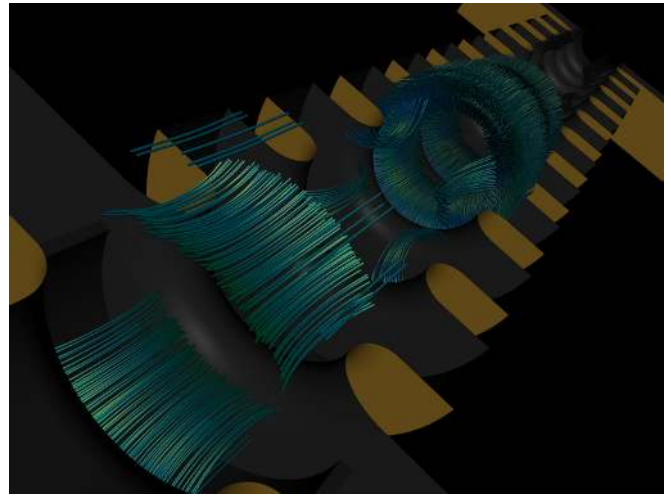


Figure 9: Time step 123,345 in the simulation of a 12-cell linear accelerator structure containing a total of 1.6 million mesh elements.

techniques addressing the challenges, and demonstrates our test results using large data sets obtained from time-varying simulations of accelerator physics. In particular, we describe how high-performance computing, commodity graphics hardware, and a hierarchical organization of visualization primitives can assist the interactive visualization of large time-varying data.

We believe a viable solution of the large data visualization problem is built upon a fusing of *supercomputing* and *commodity computing* as well as on more efficient resource allocation, investing computation on parts of the images that are the most perceptually relevant. In this way, interactive visualization can be made possible to a wide range of users from people who conduct the terascale simulations to people who use the simulation results for design.

This paper contributes to the supercomputing community in the following two ways. First, we show that for data exploration our hybrid rendering approach supporting budget-sensitive progressive visualization is more cost-effective than brute-force visualization supercomputing. It is clear that other demanding visualization applications can benefit from a similar hybrid approach. In fact, a hybrid rendering method has been developed for hardware-accelerated rendering of regular-gridded voxel data [15]. Second, we demonstrate how a compact graphics representation like our self-orienting surfaces can effectively cut down both storage and computational requirements without degrading image quality to enable interactive visualization on a commodity PC. Terascale simulation coupled with interactive visualization, used as a tool of discovery, will allow researchers to advance the frontiers of accelerator science and technology and lead to new discoveries in beam-driven science.

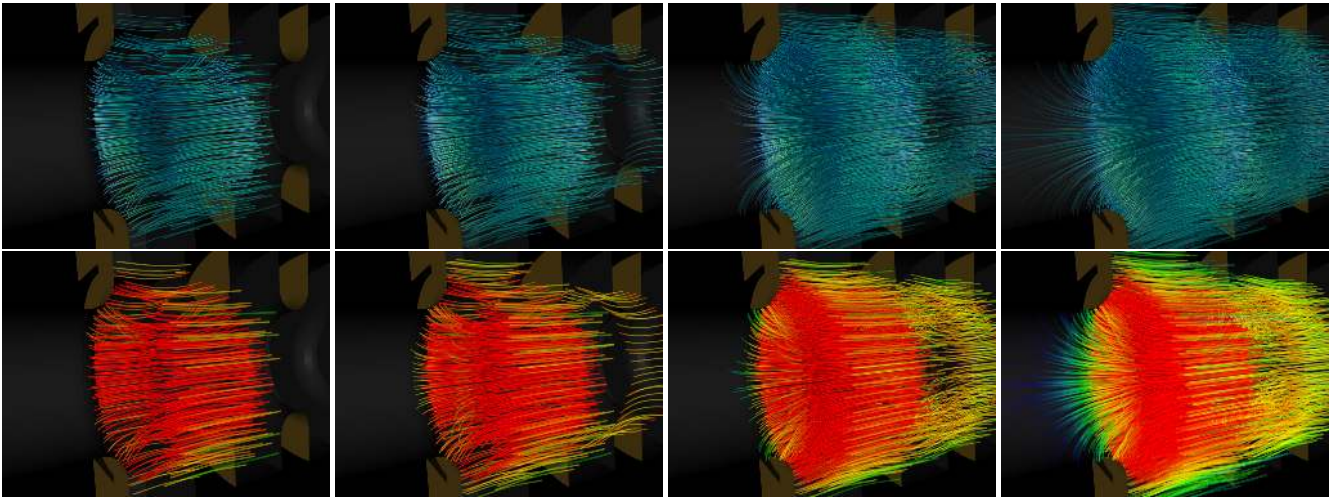


Figure 10: From left to right, incremental loading and rendering of electric field lines. Top: with the line opacity proportional to local field strength. Bottom: with enhancement using both color and transparency.

Acknowledgments

This research was sponsored by NSF PECASE, NSF LSS-DSV, and DOE SciDAC, and used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098. The authors would especially like to thank Pat McCormick at Los Alamos National Laboratory, other members of our SciDAC project team, and the Visualization Group at Lawrence Berkeley National Laboratory for the valuable discussions and assistance they have provided to us.

References

- [1] V. Interrante and C. Grosch. Strategies for effectively visualizing 3D flow with volume LIC. In *Proceedings of Visualization '97*, pages 421–424, 1997.
- [2] B. Jobard and W. Lefer. Creating evenly-spaced streamlines of arbitrary density. In W. Lefer and M. Grave, editors, *Visualization in Scientific Computing '97. Proceedings of the Eurographics Workshop in Boulogne-sur-Mer, France*, pages 43–56, Wien, New York, 1997. Springer Verlag.
- [3] K. Ko. High performance computing in accelerator physics, March 2002. submitted to ACES (Applied Computational Electromagnetic Society) Conference, http://scidac.nersc.gov/accelerator/pdf/ko_aces.pdf.
- [4] E. Lum, K.-L. Ma, and J. Clyne. Texture hardware assisted rendering of time-varying volume data. In *Proceedings of Visualization 2001 Conference*, October 2001.
- [5] K.-L. Ma and D. Camp. High performance visualization of time-varying volume data over a wide-area network. In *in Proceedings of Supercomputing 2000 Conference*, November 2000.
- [6] K.-L. Ma and S. Parker. Massively parallel software rendering for visualizing large-scale data sets. *IEEE Computer Graphics and Applications*, pages 72–83, July/August 2001.
- [7] X. Mao, Y. Hatanaka, H. H., and A. Imamiya. Image-guided streamline placement on curvilinear grid surfaces. In *Proceedings of Visualization'98 Conference*, pages 135–142, 1998.
- [8] P. McCormick, J. Qiang, and R. Ryne. Visualizing high-resolution accelerator physics. *IEEE Computer Graphics and Applications*, 19(5):11–13, 1999.
- [9] J. Meredith and K.-L. Ma. Multiresolution view-dependent splat based volume rendering of large irregular data. In *Proceedings of 1999 Parallel and Large Data Visualization Symposium*, October 2001.
- [10] J. Qiang and R. Ryne. Beam halo studies using a 3-dimensional particle-core model. *Physical Review Special Topics - Accelerators and Beams*, 3(064201), 2000.
- [11] J. Qiang, R. Ryne, S. Habib, and V. Decyk. An object-oriented parallel particle-in-cell code for beam dynamics simulation in linear accelerators. *Journal of Computational Physics*, 163(434), 2000.
- [12] G. Schussman and K.-L. Ma. Scalable self-orienting surfaces: A compact, texture-enhanced representation for interactive visualization of 3d vector fields. In *Pacific Graphics 2002 (to appear)*, October 2002.

- [13] D. Stalling, M. Zockler, and H.-C. Hege. Fast display of illuminated field lines. *IEEE Transactions on Visualization and Computer Graphics*, 3(2):118–128, April 1997.
- [14] G. Turk and D. Banks. Image-guided streamline placement. *Computer Graphics*, 30(Annual Conference Series):453–460, 1996.
- [15] B. Wilson, K.-L. Ma, and P. McCormick. A hardware-assisted hybrid rendering technique for interactive volume visualization. In *2002 Volume Visualization and Graphics Symposium (to appear)*, October 2002.
- [16] M. Wolf, A. Guetz, and C.-K. Ng. Modeling large accelerator structures with the parallel field solver tau3p, March 2002. submitted to ACES (Applied Computational Electromagnetic Society) Conference, http://scidac.nersc.gov/accelerator/pdf/wolf_aces.pdf.