Quantitative and comparative visualization applied to cosmological simulations

Institute of Physics Publishing Journal of Physics: Conference Series **46** (2006) 526–534
doi:10.1088/1742-6596/46/1/073 SciDAC 2006

Quantitative and comparative visualization applied to cosmological simulations

James Ahrens, Katrin Heitmann, Salman Habib, Lee Ankeny, Patrick McCormick, Jeff Inman
Los Alamos National Laboratory

Ryan Armstrong, Kwan-Liu Ma
University of California at Davis

E-mail: `ahrens@lanl.gov`

Abstract. Cosmological simulations follow the formation of nonlinear structure in dark and luminous matter. The associated simulation volumes and dynamic range are very large, making visualization both a necessary and challenging aspect of the analysis of these datasets. Our goal is to understand sources of inconsistency between different simulation codes that are started from the same initial conditions. Quantitative visualization supports the definition and reasoning about analytically defined features of interest. Comparative visualization supports the ability to visually study, side by side, multiple related visualizations of these simulations. For instance, a scientist can visually distinguish that there are fewer halos (localized lumps of tracer particles) in low-density regions for one simulation code out of a collection. This qualitative result will enable the scientist to develop a hypothesis, such as loss of halos in low-density regions due to limited resolution, to explain the inconsistency between the different simulations. Quantitative support then allows one to confirm or reject the hypothesis. If the hypothesis is rejected, this step may lead to new insights and a new hypothesis, not available from the purely qualitative analysis. We will present methods to significantly improve the scientific analysis process by incorporating quantitative analysis as the driver for visualization. Aspects of this work are included as part of two visualization tools, ParaView, an open-source large data visualization tool, and Scout, an analysis-language based, hardware-accelerated visualization tool.

1. Introduction

As is widely recognized, the immense growth in size and complexity of the available datastream from observations, experiments, and simulations makes information extraction from the data a remarkably daunting task. Additionally, developing intuition about the underlying physical processes becomes ever more difficult. Even “one-bit” questions (e.g., is this object a rock?) are not easy to answer if the data is presented in an unfamiliar basis. The situation is far worse when it comes to truly complex systems where it is not only the richness of the data – but also the task of discovering and expressing the appropriate question – that can defeat attempts at qualitative and quantitative analysis. Large datasets describing simulations or observations of complex systems present diverse opportunities to investigate them, e.g., multilevel changes with time, characteristic/anomalous features, nonlinear component interactions, and parametric variations.

It is our contention that a "top-down" approach to dealing with this situation – one that does not take into account specific details and characteristics of the dataset being investigated – is doomed to failure. In this paper we advocate instead a "bottom-up" path by bringing together computer and application scientists in an approach that aims to organically combine qualitative and quantitative methods. The complex system we wish to investigate is nothing less than the Universe itself.

Upcoming cosmological observations that survey large patches of the sky such as the Large Synoptic Survey Telescope project will produce 30 TB per night of data. Simulations carried out to interpret observations of this type already produce TBs of data per simulation and this will rise to PBs within a decade. To have any hope of realizing, encapsulating, and interpreting the enormous wealth of information contained in such datasets, we have to find very efficient ways to explore and analyze them, with the goal of eventually automating many such tasks.

The first step in the analysis process is to actually "look" at the data via a visualization system. Our eye/brain complex still remains one of the most sensitive analysis tools, albeit (apparently) optimized for certain types of qualitative analysis. Due to the size and complexity of the datasets this first step is already highly nontrivial. For our application, the visualization desiderata consist of (i) a scalable/interactive system, that (ii) allows hierarchical (coarse/fine) views of the data, including projections, and (iii) is steerable in the high-dimensional space of the dataset. The linchpin of our methodology is the goal of seamlessly integrating the above with quantitative analysis driven by the visualization process, including the possibility of on-the-fly definition, implementation, and trials of new analysis measures. Such a tool, which powerfully melds the strengths of *qualitative and quantitative analysis capabilities*, would be invaluable for the study of very large datasets.

We will describe a broad outline and initial work on a general framework for doing visualization-directed analysis (called VDA) of complex datasets. As part of our framework we will incorporate a component architecture for tightly coupling data analysis with visualization. Interfaces will allow users to easily extend the framework by providing custom analysis and visualization components. While the framework will be applicable to a wide range of scientific domains, specific components will be created for analyzing and visualizing data from both observations and simulations in the domain of cosmology. Initial work in this direction has been very encouraging and has already generated new scientific results [1]. The development will be kept general enough that the framework can be modified for use in other areas of science where (broadly similar) complex systems are under investigation. The very close collaboration of the computer scientists with the physicists will ensure that the framework addresses immediate and future needs of the physics community – the more widely it is adapted, the faster will it progress. We reiterate that the integration of visualization and quantitative analysis envisaged here is directly targeted at the generation of *new* science, not at making pretty pictures.

In 1991, Peskin et al. [2] suggested the methodology of "Interactive Quantitative Visualization" (IQV), which allows the user to extract *quantitative* information directly from the visual presentation. They state that, "At present, IQV and some of the more complex visualization techniques should be viewed as separate but complementary methodologies; as visualization techniques advance, we expect that these methodologies will merge." Fifteen years later, the time is ripe for this merging process and this is precisely what the VDA framework seeks to achieve. Most current applications focus on query based searches in a dataset, e.g. show data above a certain temperature threshold [3], or feature extraction and tracking [4]. Our new framework will incorporate these methodologies, but it will go a very significant step further as described below.

As already mentioned, the main data sources for this project will be cosmological observations and simulations. We will specifically target data from the largest-ever astronomical survey, the Sloan Digital Sky Survey (SDSS) [6], of which Los Alamos is a full member. The institutional membership grants our team access to the entire SDSS database. To date, the database contains roughly 30 TB of images and spectra. These data support studies ranging from asteroids and nearby stars to the large scale structure of the Universe. In order to interpret cosmological observations and advance our understanding of the evolution and dynamics of the Universe we have to carry out sophisticated, large-scale simulations encompassing different, complex physical processes. Los Alamos researchers are at the forefront of this endeavor and we have carried out some of the largest suites of cosmological simulations ever performed [7].

2. General Approach

The broad overarching goal of this project is the creation of an integrated visualization-directed analysis (VDA) framework aimed at generating new scientific insights and results. This goal however needs to be more precisely defined in order for it to be effectively targeted. For the work outlined here, the scientific context lies in the analysis of observational and simulation datasets. The problems to be addressed by applying VDA are diverse and range from searching for specific structures, conducting morphological analysis, uncovering and understanding subtle differences between results from different codes, and between simulations and observations, testing adaptive mesh refinement (AMR) criteria, studying the environmental influence on structure formation, etc. One important goal is to bring verification and validation of complex codes to the next level by an integrated analysis of observations and simulations using the new VDA framework. The scientific goals in turn set priorities for the computer science tasks such as modular methods to express the definition of relevant structures followed by integration of the new object as a "first-class" citizen in the same sense as the original structured/unstructured computational grid. Investigations of the newly defined structures then inherit all three of the visualization desiderata set down earlier. The computational starting point for this project are two visualization tools, ParaView, an open-source large data visualization tool, and Scout, an analysis-language based, hardware-accelerated visualization tool. In the next two subsections, we will describe properities of these tools, ParaView and Scout, and their application to quantitatively comparing visual and analytic results of cosmological simulations.

2.1. Using ParaView

ParaView is a scalable visualization tool which is designed as a layered architecture [5]. The foundation and first layer of ParaView is the visualization toolkit (VTK). VTK provides data representations, algorithms, and a mechanism to interconnect these to form a working program. The second layer is a parallel extension to the visualization toolkit which supports streaming of all data types and parallel execution on shared and distributed memory machines. The third layer is ParaView itself. ParaView provides a graphical user interface and transparently supports the visualization and rendering of large datasets via hardware acceleration, parallelism, and level-of-detail techniques. Figure 1 shows an example of ParaView running on a tiled display wall.

An example taken from our preliminary work provides a good illustration of the steps involved – the scientific workflow – in an integrated VDA framework. The example is a code comparison test where two codes are run with exactly the same cosmological initial condition, but with very different algorithms to solve the N-body problem. The first code, FLASH [8], is a grid-based AMR code, the force resolution being increased (the grid spacing refined) when the density crosses a certain threshold. The second code is the Hashed-Oct Tree code, HOT, developed at LANL by Michael Warren [9]. Using comparative visualization functionality we added to ParaView, we create Figure 2. This figure shows the visualization of a specific zoomed-in region in both simulations, with substantial substructure. The comparison immediately shows that the FLASH results (first panel) have apparently much less substructure than found by HOT (second

Figure 1. ParaView displaying a visualization of FLASH simulation data using vector arrow glyphs sized and colored by velocity magnitude on an 18 panel (3 by 6) tiled display wall.

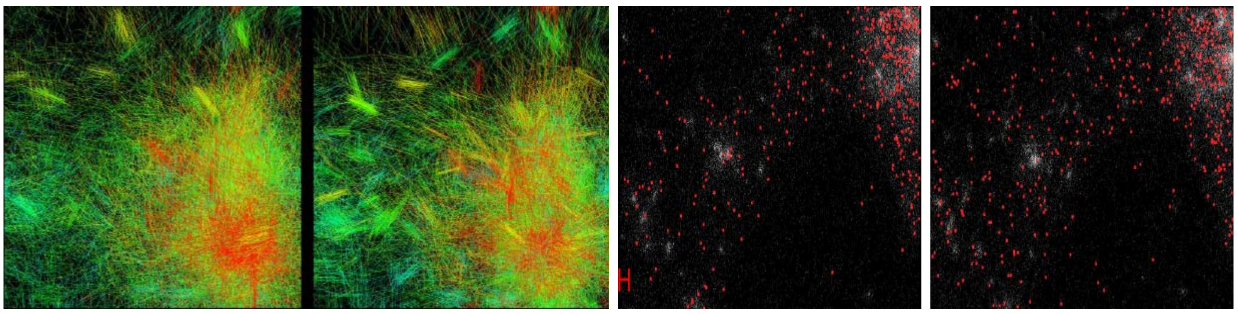

Figure 2. Results from ParaView. First panel: zoom into a high density region from the FLASH simulation, second panel: zoom into the same region from the HOT simulation. Note the deficiency of substructure in the FLASH output. Third panel: halos from FLASH, fourth panel: halos from HOT in the same region.

panel). This qualitative statement must now be quantified in a way that can directly help to improve the FLASH algorithm so as to resolve the substructure.

The first task is to define a suitable structure – this is the "halo." We measure the distance between a pair of particles and if this distance is below some fixed value, e.g. a fifth of the mean interparticle separation, they are said to belong to the same halo. The query process is extended to their immediate neighbors to check whether they are also halo members; this group-finding algorithm is the friends-of-friends finder [10]. We can now count the number of halos in selected regions (or the whole simulation), leading to a *quantitative* measure of the amount of structure in the codes. In fact, we could have chosen other ways to define these structures: (i) by identifying regions of very high overdensity, or (ii) by measuring the infall velocity of particles in a specific region. The *definition* and *description* of such structures in a way that can be simply quantified is a key point. We first visually inspect the distribution of the halos in FLASH and HOT as shown in the third and fourth panels of Figure 2. FLASH has clearly many fewer halos than HOT. Since the refinement criterion for the FLASH grid is based on density thresholds, the next logical question is, how does the number of halos depend on the density? To answer this question, we added a module to ParaView to set different density thresholds and count halos within regions specified by different density cuts (Figure 3). The histograms display the number

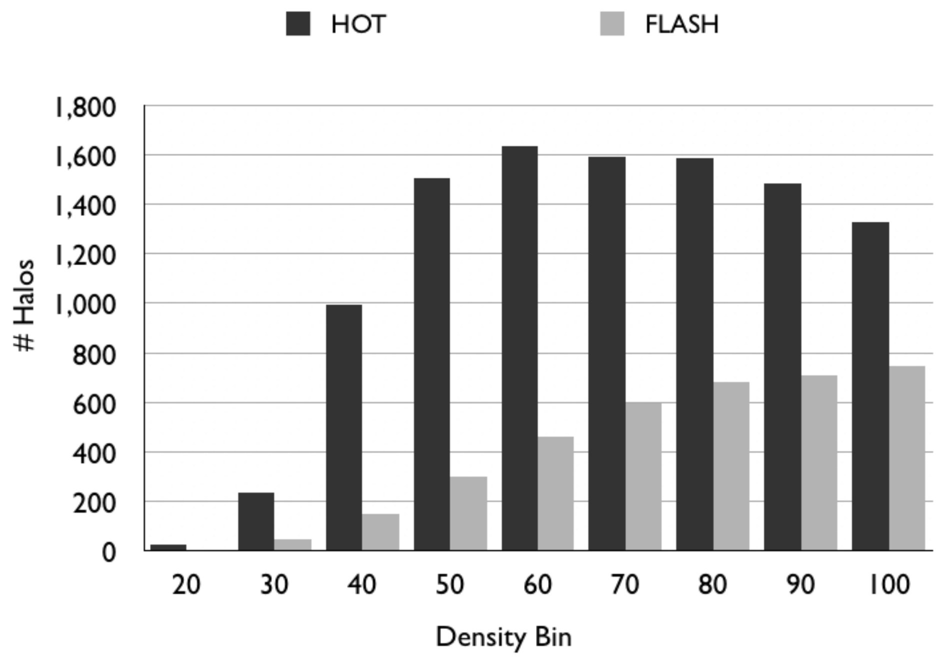

Figure 3. Results from ParaView. Number of halos in a given density bin for HOT (dark gray) and FLASH (light gray).

of halos in a given density bin. The comparison shows that FLASH does not capture the correct number of halos in any density region, even at higher densities, where the refinement should have been sufficient. This tells us that the refinement must begin at an earlier epoch, and that early-time evolution plays a fundamental role in the formation of structures at late times. This example shows that a visualization tool with flexible and powerful quantitative capabilities can play an important role in advancing our understanding of the simulation and is a crucial element in the pathway to solve the identified problem.

The general workflow in such a case is the following: (i) Inspect the dataset and identify problems or features and structures of interest (missing substructure in the previous example). (ii) Find a way to define the structures of interest (find halos with a friends-of-friends algorithm). (iii) Quantify the structures of interest (count the halos), perhaps specify more queries for this quantification (count halos with respect to density thresholds). After this last step, the process might have led to an answer or to a new hypothesis (new criterion for mesh-refinement is needed) which can then be tested in turn. It is very important to note that a VDA framework *immensely* shortens the time and effort for the scientist to analyze the problem and find a solution. While the example described above would take several days to be carried out in the standard way – the scientist writes or plugs in a parallel halo finder, an interpolater between particles and density, a routine to define density thresholds, and then after each step goes back to the analysis tool, reads in the data and draws new conclusions – our integrated VDA framework will result in a quantitative analysis of the data in a very short amount of time by providing default analysis components for common tasks, while additionally allowing for custom components as plugins.

In fact, if one follows the conventional path (and we speak here from personal experience!), some parts of the analysis will just never get done because they take too long and are considered simply too difficult to do. Important opportunities to explore the data and generate new science can be missed, and are undoubtedly being missed.

2.2. Using Scout

The visualization and analysis process involves the investigation of relationships between the numerical and spatial properties of one or more data sets. Several different techniques use indirect mappings, such as transfer functions, to assign optical properties like color and transparency to data values. While these techniques are powerful, they have had limited acceptance in the scientific community because they require scientists to work in a secondary data space (e.g. the transfer function domain). This can make it difficult to efficiently express queries and mathematical operations that would be natural in the in the original data space. As the success of many different scripting languages have shown, it is beneficial to have the ability to directly perform mathematical operations when exploring and analyzing data. The capabilities of today's commodity graphics hardware provides a unique opportunity to help reduce the impact on performance of these calculations.

Scout is a software system that supports a data parallel programming language that supports expression based queries that are evaluated in the data space. In this case, we consider a query to be a set of relational and conditional expressions based numerical values. The system reduces the computational bottleneck by utilizing the graphics processor (GPU) not only for rendering, but also as a computational co-processor responsible for offloading portions of the work that would traditionally be executed on the CPU. The language-based interface allows scientists to process multivariate data, express derived data, and define the associated mappings to the final image. Supported visualization and rendering techniques include two dimensional texture mapping, volume rendering, ray casting, point rendering and isosurfaces.

Current graphics processors must be programmed directly using a graphics-specific API, such as OpenGL or DirectX. The clear disadvantage of this approach is that all operations must be expressed in terms of graphics primitives. In designing the Scout programming language we had several goals in mind:

(i) It should be simple and concise. The primary motivation for this was to make the language easy to learn and use. In addition, we wanted to assure that compilation times would be fast enough to achieve interactive performance for the user.

(ii) It should reveal the parallel nature of the underlying hardware without complicating the language.

(iii) The language should hide any nuances introduced by the graphics API and the graphics hardware.

(iv) It should provide the user with flexible methods for producing both general purpose computations and visualization results.

The Scout language is loosely based on a combination of the *C** and CM FORTRAN programming languages. This results in a language structure that can be applied to many areas within the scientific and visualization communities. In Figure 4 we show a portion of a Scout program to create a visualization application that displays dark matter particles and halos selected by density region and halo mass. Figure 5 shows the results of running the Scout program. The resulting program provides an interactive interface for selecting and displaying specific density regions and halo mass values. Using the GPU, the program can query and display results, extremely quickly, at 24 frames a second.

```
// Create a viewport that is 1/2 of the window...
viewport "MC2" (0.0, 0.0, 0.5, 1.0) {

  // Compute the magnitude of the velocities...
  float mag(shapeof(mc_velocity));
  compute with shapeof(mag)
    mag = magnitude(mc_velocity);

  // Render points...
  render points with shapeof(mc_points) {
    where(density >= density_lo && density <= density_hi)
      image = hsva(240.0 * (max(mag) - mag) / (max(mag) - min(mag)),
                   1.0, 1.0, 1.0);
    else
      image = null;
  }

  // Render halos...
  render points with shapeof(mc_halos)  {
    where(mass >= mass_lo && mass <= mass_hi &&
          density >= density_lo && density <= density_hi)
      image = rgba(1.0, 0.0, 0.0, 1.0); // color halos red...
    else
      image = null;
  }
}
```

Figure 4. A portion of a Scout program that computes velocity magnitude and selects and renders dark matter particles and halos.

3. Future Work and Conclusions

The successful completion of this project will result in a powerful, high-performance VDA framework that allows very efficient qualitative and quantitative analysis of complex data. Cosmology, our targeted science field, has over the last decade entered the era of "precision cosmology". Predictions from theory and simulations are being confronted by high-accuracy observations. To extract information from the combination of observations and simulations requires a sophisticated framework which provides high-precision analysis tools and which supports the workflow of the scientific data analysis at every step. A unique aspect of our proposed VDA framework is the feedback loop between visualization, analysis, and scientific discovery: due to the integration of new analysis features the VDA framework becomes more flexible, in turn allowing the scientist to view his/her data in a new way leading to new scientific insights. These scientific insights can then result in entirely new data analysis strategies.

The creation of the VDA framework has to proceed in well-defined stages. To compare simulations with observations, the simulation datasets have to be viewed in exactly the same way as the observations, i.e., projections on the celestial sphere with appropriate redshift information extracted from the line-of-sight velocity of the simulated object (star/galaxy/cluster). In addition, to compare simulations and obversations we also define classes of useful primitives, e.g., halos, densities, morphological measures such as Minkowski functionals, and the associated operators that can act on each class, e.g., filters, scaling functions, etc. The framework will have two important features, viz., these primitives are composable (e.g., show halos only where the underlying particle density field satisfies a threshold condition) and that operators automatically

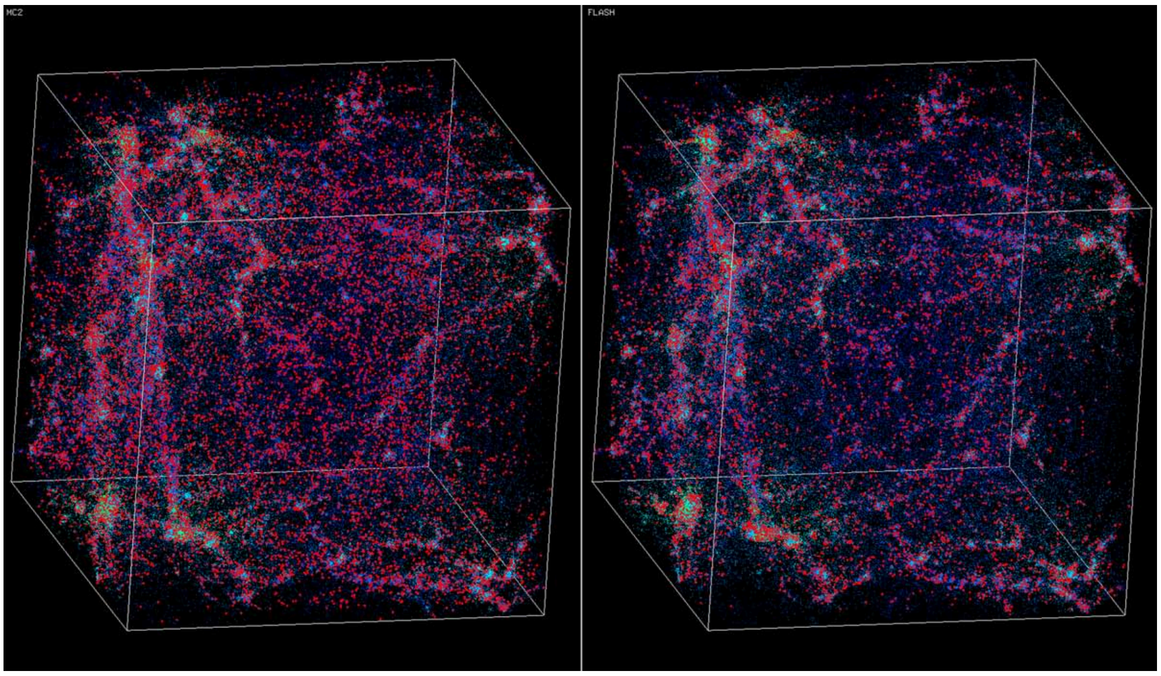

Figure 5. A qualitative comparison of the results produced by MC^2 (left) and FLASH (right) using Scout. The results show halos with low mass in low density regions.

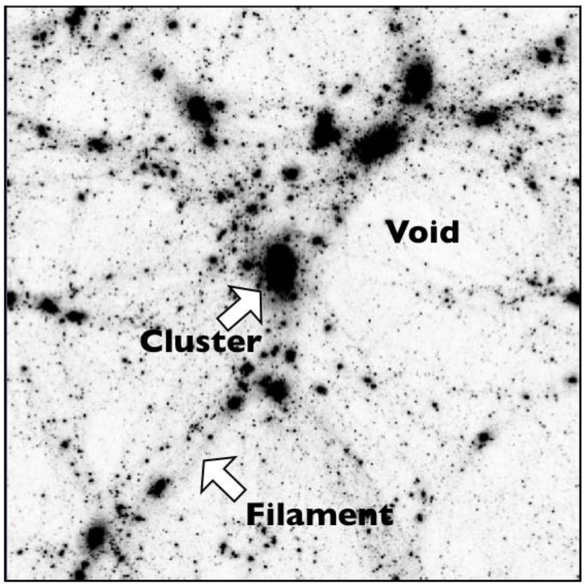

Figure 6. Structures in a cosmological simulation.

act on any new primitives created in the derived classes.

As members of a derived class, halos inherit attributes of the base "particle" class, thus halo density can be defined in the same way as particle density using a cloud-in-cell density algorithm applied to the halos. The primitives live at the bottom level of a hierarchy of "structures" which are built up from combining primitives with different levels of complexity. Extending the halo primitive, the next step could be a "filament" or a "cluster" or a "void" which are collections of halos with very different geometries and halo populations. An example of these objects is shown in Figure 6.

In conclusion, we have described an integrated visualization-directed analysis (VDA) framework aimed at generating new scientific insights and results. We have presented initial results implementing this framework in ParaView and Scout.

4. References

[1] J. Ahrens et al. "Quantitative and Comparative Visualization Applied to Cosmological Simulations", LANL technical report.

[2] R.L. Peskin, S.S. Walther, A.M. Froncioni, and T.I. Boubex, "Interactive Quantitative Visualization", IBM J. Res. Develop., 35, 205 (1991).

[3] K. Stockinger, J. Shalf, K. Wu and E.W. Bethel, "Query-Driven Visualization of Large Data Sets", IEEE Visualization 2005, Minneapolis, MN, pp.167-174.

[4] J. Chen, D. Silver and M. Parashar, "Real Time Feature Extraction and Tracking in a Computational Steering Environment" Proceedings of HPC2003, San Diego, pp 155-160, March 2003.

[5] J. Ahrens, B. Geveci, and C. Law, "ParaView: An End-User Tool for Large Data Visualization". In C. Hansen and C. Johnson, editors, *The Visualization Handbook*, pages 717-731, Academic Press, 2005.

[6] For more information on the Sloan Digital Sky Survey see: *http://www.sdss.org*

[7] K. Heitmann, Z. Lukic, S. Habib, and P.M. Ricker, "Capturing Halos at High Redshifts", astro-ph/0601233, ApJL submitted.

[8] B. Fryxell et al., ApJS **131**, 273 (2000).

[9] M.S. Warren and J.K. Salmon, in *Supercomputing '93*, 12, Los Alamitos IEEE Comp. Soc. (1993).

[10] J. Einasto, A.A. Klypin, E. Saar, S.F. Shandarin, MNRAS, **206**, 529 (1984).