EG UK Theory and Practice of Computer Graphics (2005)
L. Lever, M. McDerby (Editors)

Visualization of Smoothed Particle Hydrodynamics for Astrophysics

Rick Walker[1], Peter Kenny[2] and Jingqi Miao[2]

[1] Canterbury Christ Church University College [2] University of Kent

Abstract

Scientific visualization still presents a number of challenges. Effective visualization straddles several problem domains - the data structures needed to support visualization of large data sets, rendering techniques for fast and interactive display of this data, and enough understanding of the data involved to construct visualizations that provide real insight into the problem. Data from Smoothed Particle Hydrodynamics simulations is of particular interest, due to its time-dependent, point-based nature and its prevalence in simulation in astrophysics in areas such as star formation and evolution. This paper looks at some of the issues associated with building a useful, usable visualization tool for SPH data from astrophysics, and describes a prototype of such a system. This paper describes work in progress.

1. Introduction

Smoothed Particle Hydrodynamics (SPH) [GM77, Mon92] is a technique for Computational Fluid Dynamics (CFD) simulations which uses the Lagrangian description of fluid flow on a set of simulated particles. These fluid elements move under the influence of hydrodynamic and, frequently, gravitational forces. Each particle of fluid has equal mass, and a density value that may differ. Applications for this technique are centred around nonaxisymmetric phenomena in astrophysics - events such as asteroid impacts and collisions [DBE*04], galaxy mergers [Mih99], supernova explosions [BYH03] and star formation and evolution [NSH04]. The output from SPH simulations consists of a point cloud, with each point having additional characteristics such as density and temperature.

As with other forms of scientific data, the amount of information produced by such a simulation can be huge. While the original paper [GM77] used a set of 80 particles, modern supercomputers are capable of running the same technique with tens of millions of particles. Since data is produced for each time step, the quantities of data involved are immense. Scientific visualization aims to tackle the problem of interpretation of this data, by providing visualizations that help provide insight into the science of the system. This is especially true in astrophysics, where hypothesis formation is aided hugely by even simple visualizations.

Alongside scientific visualization has grown the new science of information visualization [Spe01], where the aim is to provide insight into more abstract data sets through visualization, insight that might lead to improved understanding or new ideas. The overlap between the two fields tends to be minimal. Many scientific applications would not necessarily benefit from more abstract visualizations - medical applications are a case in point.

However, data from SPH simulations is interesting for a number of reasons. Since it has no inherent structure (unlike grid-based CFD methods), there are many possibilities for visualization. Its point-based nature means that many recent techniques for point-clouds can be applied. Its time-based nature suggests that animation would be a route worth pursuing. In addition, since existing visualizations are essentially 2D plots of the data [BYH03, DBE*04], there may be a whole range of techniques from information visualization that can be usefully applied, by working together with astrophysicists involved in SPH simulation.

This paper describes work in progress to produce a visualization system for Smoothed Particle Hydrodynamics datasets from the field of astrophysics. Section 2 gives background to the data set and the disparate fields brought together by this research. In section 3 we present some ideas to be applied to the visualization of this data, in the form of desirable features for our system. Section 4 provides brief implementation details and some examples of visualization of an SPH dataset.

2. Background and Related Work

This section looks at existing practice in SPH visualization, and provides some background on the three main fields of interest to the present research: data structures for large data set visualization, rendering techniques for point-cloud data, and information visualization.

2.1. SPH Visualization

The point cloud output from an SPH simulation is unstructured. Data points may be tightly clustered or sparsely spread, dependent on the results of the simulation. Furthermore, distribution may change significantly between time steps. The range of values for point attributes such as temperature and density can be very large.

Currently, data visualization often takes the form of contour plots for density over time steps. These are produced by taking slices through the data and performing resampling. The resultant regular grid is then used to produce the plots. By examining plots for a number of time steps, the formation of features can be tracked and described. Figure 1 shows the gradual elongation of the core in an SPH simulation of the Eagle Nebula. Animations of plots make the formation of features easier still to track.

Without foreknowledge of the existence of features, these visualization techniques can be challenging to apply in a timely fashion. Since so much depends on the choice of slice (its angle and location), often several attempts are needed to reveal features of interest in the data set. This also presents the risk that potentially interesting features may be missed or overlooked.

2.2. Data Structures for Large Data Set Visualization

The nature of SPH data is such that it lends itself to storage in a wide variety of structures. BSP trees, k-d trees and octrees ([FKN80], [GG98]) are all spatial decomposition structures that can be used to store 3D data of this type. This spatial decomposition makes it hard to predict a priori the storage requirements of such structures, since the point-cloud data is unlikely to follow a regular pattern. Distance fields [FPRJ00] can be used to store efficiently surface information for volumetric data sets, but this would limit the rendering to one or more isosurfaces.

While the use of point-clouds to approximate surfaces is an ongoing research topic, structures used in this research are typically designed to store point-representations of surfaces. Thus structures such as LDC trees [PZvBG00] and the Bounding Sphere [RL00] hierarchy are not well suited to unstructured clouds. Elsewhere, [HE03] generates an advanced hierarchical data structure using principal component analysis for clustering which generalises well to multiple time steps [HLE04].

2.3. Rendering Point Cloud Data

The rendering of point cloud data can be accomplished through a range of volumetric rendering techniques. One such technique is to extract isosurfaces from the data (through algorithms such as marching cubes [LC87]), and render it as a polygonal surface. This approach works well for data sets with clear boundaries and structure (for example, in medical data) but for astrophysics simulation data the distinctions are likely to be less clear. To visualize the point cloud usefully using a surface representation, it would be necessary to depict multiple surfaces (at different isovalues), but it is not obvious which isovalues should be used.

The alternative to surface rendering is to project the whole data set to the screen at once. Technological and algorithmic advances have made this direct volume rendering much more achievable. 3D texture mapping [CCF94] uses hardware on modern graphics cards to project whole slices of data to the screen at a time, with the final image composed of a number of these slices layered. This approach assumes the points to lie on a regular grid, hence the data would likely need to be resampled to such a structure, and slices extracted for each different viewing angle. Ray-casting [Lev90] traces a ray from each pixel in the image through the object domain and calculates the light and shade of that pixel as a function of the voxels along the ray's route. Interactive ray tracing can, however, be prohibitively expensive in computational terms.

Splatting works by projecting each voxel cell to the image, with footprint determined by properties of the data. Real-time rendering using splatting has been possible by using Gouraud shaded polygons [Wes90] to approximate splats and encoding the data as an octree. More recently, work [RPZ02] has aimed at the use of points instead of triangles as the primitive of choice for images, and again this has been aided by recent advances in hardware. For example, for simple cases, rendering a volumetric dataset is possible at interactive frame-rates by simply loading the dataset into memory on the graphics card then drawing each point separately to the screen. Rendering quality can be enhanced through the use of vertex shaders to specify additional parameters for the point data [HE03].

2.4. Information Visualization

The field of information visualization can inform this research on two fronts: firstly, by the variety of generic techniques in this area that might be applicable to scientific as well as more abstract data sets, for example the focus-and-context model [CB04], and secondly by providing a framework for the construction and evaluation of new visualiza-

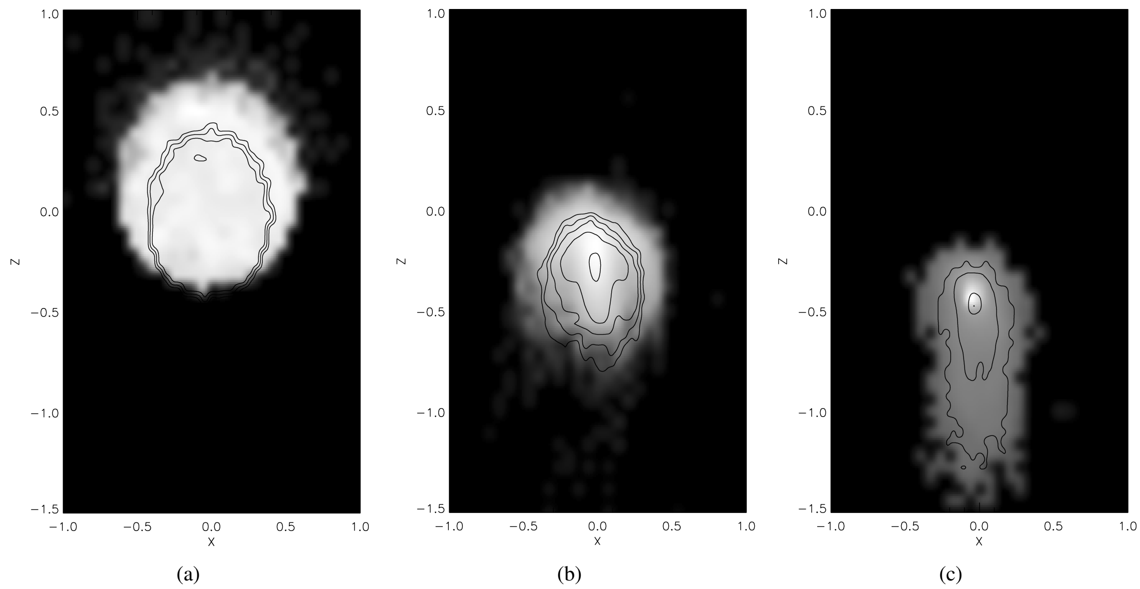


Figure 1: *Evolution of Bok globules at t= (a) 4.3×10^3 yrs (b) 2.29×10^5 yrs (c) 3.75 ×10^5 yrs, grayscale value proportional to* $\log(density)$*, density scale varies between images*

tions [CM97, Shn96]. Questions such as whether a particular visualization is appropriate to the task for which it is used, or whether researchers are working closely enough with the end-users of such systems [MLZ*02] can be addressed in this context. Scientific visualization tools typically focus on plotting the data accurately in context [Com]. This is not necessarily the best way of providing insight into and understanding of the data set.

3. Interactive SPH Visualization

In designing a visualization system for SPH data, it is necessary to consider first the goals of such a system. The system can work with just the data produced from the simulation. There are some interesting possibilities for computational steering, and hence to exploratory simulation, with the goal of providing insight not merely into the data, but into the system as a whole. However, to add such possibilities to a system requires an in-depth knowledge of the simulation software and method.

Given that we can work with the data directly, the problem is reduced to visualization of six-dimensional data - three coordinates, temperature, density and time. Typically, existing visualizations [BYH03, BCL99] used in the field would be of the form of a two dimensional contour image formed from a slice through the data set. This involves a resampling process, where currently the nearest-neighbour algorithm is used, and averages taken if multiple points map to the same grid square.

[Eic97] suggests a set of eight design guidelines for effective visualization - task-specific, reduced representation, data encoding, filtering, "drill down", multiple linked views and user interface. Of these, task-specific is taken for granted. Reduced representation is not desirable. Data encoding is a simple transfer function that maps temperature to a colour scale and density to the opacity of each splat. "Drilling down" through the data is less important than in other fields, since the precise values of a single point are of less importance. The remaining guidelines map to the desired feature set below.

- **Interactivity:** it should be possible to rotate and zoom on the data set, as displayed, in real-time. This encourages data exploration.
- **Selection:** while it should be possible to view the complete data set (or at least, one complete time step), it should also be possible to restrict the visualization to specified ranges of temperature or density. This allows focus on interesting or informative patterns in the data without distraction.
- **Animation:** given a time-based data set, it would be advantageous to see how the data changes over time visually. This can be represented by difference images or, more simply, through animation. It should be possible to animate smoothly between time-steps in a controllable fashion.
- **Multiform views:** as discussed in section 2.3, it is possible to represent this data in several ways, by, for example rendering an isosurface for a specified value, or by splatting the entire set. Viewing the same data in two or more different ways at the same time may lead to understand-

ing that a single view would not provide [Rob00]. It also allows the system to work at least as well as the previous one. If one of the views offered is a 2D slice through the 3D view, then it becomes easier to relate the new image to existing images and states. The addition of features such as brushing could aid coordination between views.

4. Work Completed

It seemed natural to begin the research by implementing these features in order. So this section discusses implementation of interactivity, selection, animation and multiple (multiform) views in a prototype system.

4.1. Implementation Techniques

The system was implemented in an extensible, object oriented fashion using C++ and OpenGL [OGL] with the GLUI [Rad] toolkit for additional interface components. This has obvious advantages:

- OpenGL and C++ can interact in a platform-independent fashion, so the system is portable between Windows and Unix systems.
- Compiled C++ code is fast, and the use of OpenGL ensures that hardware acceleration is handled at driver level.
- Since each view has a class of its own, overlaying multiple views in the same window is easily accomplished.
- The structure is easily extensible through inheritance for experimentation.

For the purposes of experimentation, a simple flat memory structure was used for the data. This limited the amount of processing necessary before visualization through splatting. To extract isosurfaces, the data was first resampled into a regular grid using the same nearest-neighbour algorithm applied to produce the 2D plots mentioned in section 3 (for consistency), then the marching tetrahedra algorithm [DK91] applied and the resultant triangle set stored.

4.2. Feature Implementation

Interactivity: a simple mouse-based interface allows rotation and zooming of the data set in real time. This proved possible at around 35 frames per second even on relatively modest systems through OpenGL. This corresponds to a system reaction time of 28ms, and is well within the parameters defined in [Eic97] for interactivity.

Selection: the interface, as shown in Figure 2, allows for selection of a range of values to display for both temperature and density, through means of interface components. However, selection is only possible of a single range. For example, temperature ranges of 0-60 and 100-120 cannot both be selected at the same time. This could be useful in cases where only set ranges contain "useful" information to the end user.

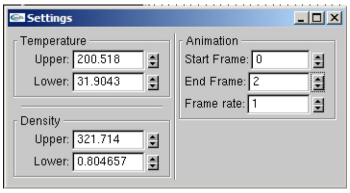


Figure 2: *Selection interface*

Animation: the interface allows selection of time-step, animation between two selected time-steps, and selection of speed of animation in frames-per-second. However, the nature of the simulation can create problems for animation: the default behaviour is to output a time-step data set when the data has changed significantly since the last step. While this is sensible for static images, and reduces storage requirements for large simulations, it means that any two time-steps may be essentially disparate, and hence the animation could be a slide show rather than a smooth flow between images.

Multiform views: currently, while two different views are available, the user is presented with only one window in which to view them. The overlay of a polygonized isosurface onto the splatted image could have some useful applications (it might, for example, make it easier to see whether points lie inside or outside of a zone), but multiple windows with the option to overlay would be preferable.

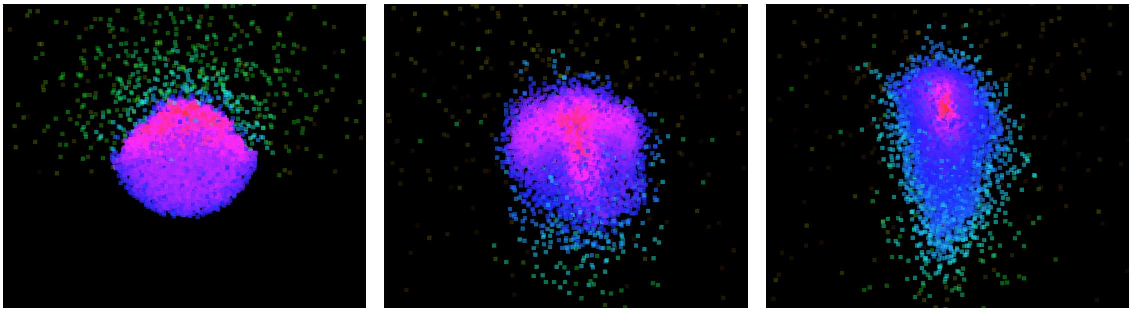
4.3. Example of application

The system was tested with a SPH data set of 10,000 particles, showing the evolution of Bok globules [BR47] in the Eagle Nebula (M16). Each particle has a position in 3D space, as well as temperature and density values. The set contained 52 separate time steps (so, 52 sets of 10,000 points), with a total input file size of c. 45MB. Results corresponding to the contour plots in figure 1 are presented in figure 3 (a)-(c).

There are several advantages to this visualization. The symmetry of the object can be observed as it evolves through rotation of the model while animation continues. This can be difficult to track with two-dimensional data plots. The selection controls allow more focused observation of particular sets of particles. Finally, the application offers a more integrated approach to visualization than techniques currently in use.

5. Conclusions and Future Work

Improved visualizations of SPH data have the ability to aid research efforts in astrophysics on two fronts. Through integration with the scientist's workflow pipeline, they can aid the speed of analysis of data produced. Through interactivity, they can help increase robustness by allowing exploration of the data set.



(a) (b) (c)

Figure 3: *Visualization at timesteps corresponding to figure 1, density mapped to hue and transparency*

From a technical perspective, there are many obvious improvements that could be made to the system, in the areas of data storage and rendering, as well as interface and efficiency considerations. Work completed thus far does however point to some interesting routes to pursue, and the risk of alienating physical scientists with visualizations that are actually less useful than the current status quo can be eliminated through the use of multiform views.

Some other possible research directions concern the issue of resampling [BM97]. This is relevant on two fronts - nearest-neighbour resampling produces a sharp difference in image quality between simulations with higher numbers of particles and those that use fewer. Different resampling schemes could conceivably reduce this gap in image quality. Aside from quality, the issue of how to resample these points realistically remains an open question: although they are modelled as particles, each particle has a mass and a density, which thus projects to a size. If a particle spans multiple cells in the resampled set, nearest-neighbour will not provide an accurate depiction of the data set. It will likely be important to use the same resampling scheme in multiple visualizations to maintain consistency, but this scheme might differ from the current system.

Beyond that, this study aims to provide a better understanding of the way scientific visualization is used in astrophysics, and how that process can be improved. This will require further cooperative research with astrophysicists to provide feedback on the system and its usefulness.

References

[BCL99] BROMM V., COPPI P. S., LARSON R. B.: Forming the First Stars in the Universe: The Fragmentation of Primordial Gas. *Astrophysical Journal Letters 527* (Dec. 1999), L5–L8.

[BM97] BRODIE K., MASHWAMA P.: Controlled interpolation for scientific visualization. In *Scientific Visualization, Overviews, Methodologies, and Techniques* (1997), IEEE Computer Society, pp. 253–276.

[BR47] BOK B. J., REILLY E. F.: Small Dark Nebulae. *Astrophysical Journal 105* (Mar. 1947), 255–+.

[BYH03] BROMM V., YOSHIDA N., HERNQUIST L.: The First Supernova Explosions in the Universe. *Astrophysical Journal Letters 596* (Oct. 2003), L135–L138.

[CB04] COHEN M., BRODLIE K.: Focus and context for volume visualization. In *Proceedings of Theory and Practice of Computer Graphics 2004, Eurographics UK* (2004), IEEE Computer Society, pp. 32–39.

[CCF94] CABRAL B., CAM N., FORAN J.: Accelerated volume rendering and tomographic reconstruction using texture mapping hardware. In *VVS '94: Proceedings of the 1994 symposium on Volume visualization* (1994), ACM Press, pp. 91–98.

[CM97] CARD S. K., MACKINLAY J.: The structure of the information visualization design space. In *IEEE Symposium on Information Visualization* (1997), pp. 92–99.

[Com] COMSOL: Femlab. http://www.comsol.com.

[DBE*04] DURDA D. D., BOTTKE W. F., ENKE B. L., MERLINE W. J., ASPHAUG E., RICHARDSON D. C., LEINHARDT Z. M.: The formation of asteroid satellites in large impacts: results from numerical simulations. *Icarus 170* (July 2004), 243–257.

[DK91] DOI A., KOIDE A.: An efficient method of triangulating equivalued surfaces by using tetrahedral cells. *IEICE Transactions Commun E74(1)* (January 1991), 214–224.

[Eic97] EICK S. G.: Engineering perceptually effective visualizations for abstract data. In *Scientific Visualization, Overviews, Methodologies, and Techniques* (1997), IEEE Computer Society, pp. 191–210.

[FKN80] FUCHS H., KEDEM Z. M., NAYLOR B. F.: On visible surface generation by a priori tree structures. In *SIGGRAPH '80: Proceedings of the 7th annual conference on Computer graphics and interactive techniques* (1980), ACM Press, pp. 124–133.

[FPRJ00] FRISKEN S. F., PERRY R. N., ROCKWOOD

A. P., JONES T. R.: Adaptively sampled distance fields: a general representation of shape for computer graphics. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques* (2000), ACM Press/Addison-Wesley Publishing Co., pp. 249–254.

[GG98] GAEDE V., GÜNTHER O.: Multidimensional access methods. *ACM Computing Surveys 30*, 2 (1998), 170–231.

[GM77] GINGOLD R. A., MONAGHAN J. J.: Smoothed particle hydrodynamics - Theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society 181* (nov 1977), 375–389.

[HE03] HOPF M., ERTL T.: Hierarchical Splatting of Scattered Data. In *Proceedings of IEEE Visualization '03* (2003), IEEE, pp. 433–440.

[HLE04] HOPF M., LUTTENBERGER M., ERTL T.: Hierarchical Splatting of Scattered 4D Data. *Computer Graphics and Applications 24*, 4 (2004), 64–72. submitted version.

[LC87] LORENSEN W. E., CLINE H. E.: Marching cubes: A high resolution 3d surface construction algorithm. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques* (1987), ACM Press, pp. 163–169.

[Lev90] LEVOY M.: Efficient ray tracing of volume data. *ACM Trans. Graph. 9*, 3 (1990), 245–261.

[Mih99] MIHOS C.: Dynamics of Mergers. *Astrophysics and Space Science 266* (1999), 195–205.

[MLZ*02] MEISSNER M., LORENSEN B., ZUIDERVELD K., SIMHA V., WEGENKITTL R.: Volume rendering in medical applications: we've got pretty images, what's left to do. In *VIS '02: Proceedings of the conference on Visualization '02* (2002), IEEE Computer Society, pp. 575–578.

[Mon92] MONAGHAN J. J.: Smoothed particle hydrodynamics. *Annual Review of Astronomy and Astrophysics 30* (1992), 543–574.

[NSH04] NAGAMINE K., SPRINGEL V., HERNQUIST L.: Star formation rate and metallicity of damped Lyman α absorbers in cosmological smoothed particle hydrodynamics simulations. *Monthly Notices of the Royal Astronomical Society 348* (Feb. 2004), 435–450.

[OGL] OGL: Opengl. OpenGL Architecture Review Board, http://www.opengl.org.

[PZvBG00] PFISTER H., ZWICKER M., VAN BAAR J., GROSS M.: Surfels: Surface elements as rendering primitives. In *Siggraph 2000, Computer Graphics Proceedings* (2000), Akeley K., (Ed.), ACM Press / ACM SIGGRAPH / Addison Wesley Longman, pp. 335–342.

[Rad] RADEMACHER P.: Glui user interface library. http://www.cs.unc.edu/~rademach/glui/.

[RL00] RUSINKIEWICZ S., LEVOY M.: QSplat: A multiresolution point rendering system for large meshes. In *Siggraph 2000, Computer Graphics Proceedings* (2000), Akeley K., (Ed.), ACM Press / ACM SIGGRAPH / Addison Wesley Longman, pp. 343–352.

[Rob00] ROBERTS J. C.: Multiple-View and Multiform Visualization. In *Visual Data Exploration and Analysis VII, Proceedings of SPIE* (January 2000), Erbacher R., Pang A., Wittenbrink C.,, Roberts J., (Eds.), vol. 3960, IST and SPIE, pp. 176–185.

[RPZ02] REN L., PFISTER H., ZWICKER M.: Object space ewa surface splatting: A hardware accelerated approach to high quality point rendering, 2002.

[Shn96] SHNEIDERMAN B.: *The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations*. Tech. Rep. UMCP-CSD CS-TR-3665, College Park, Maryland 20742, U.S.A., 1996.

[Spe01] SPENCE R.: *Information Visualization*. Addison-Wesley, 2001.

[Wes90] WESTOVER L.: Footprint evaluation for volume rendering. In *SIGGRAPH '90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques* (1990), ACM Press, pp. 367–376.