Exploratory Simulation for Astrophysics

Rick Walker[a], Peter Kenny[b] and Jingqi Miao[c]

[a]Department of Computing, Canterbury Christ Church University, Canterbury, Kent CT1 1QU, UK;

[b]Computing Laboratory, University of Kent, Canterbury, Kent CT2 7NF, UK;

[c]Centre for Astrophysics and Planetary Science, School of Physical Sciences, University of Kent, Canterbury, Kent CT2 7NR, UK

ABSTRACT

Exploratory simulation involves the combination of computational steering and visualization at interactive speeds. This presents a number of challenges for large scientific data sets, such as those from astrophysics. A computational model is required such that steering the simulation while in progress is both physically valid and scientifically useful. Effective and appropriate visualization and feedback methods are needed to facilitate the discovery process. Smoothed Particle Hydrodynamics (SPH) techniques are of interest in the area of Computational Fluid Dynamics (CFD), notably for the simulation of astrophysical phenomena in areas such as star formation and evolution. This paper discusses the issues involved with creating an exploratory simulation environment for SPH. We introduce the concepts of painting and simulation trails as a novel solution to the competing concerns of interactivity and accuracy, and present a prototype of a system that implements these new ideas. This paper describes work in progress.

Keywords: Computational steering, scientific visualization, modelling, simulation, computational fluid dynamics, smoothed particle hydrodynamics, star formation, star evolution

1. INTRODUCTION

In large-scale physical simulation and modelling, data analysis is usually carried out separately from the simulation. Typically, a simulation is run to completion, or to a set intermediate stage, then the data produced is evaluated. If the results are not sufficiently accurate or some other problem is found, the input parameters of the simulation are adjusted and the simulation restarted from the beginning. This workflow evolved for a number of reasons: chiefly, the need to run the simulation on high-performance supercomputers, while the analysis could be accomplished with lower specification hardware. More recently, the workflow has begun to change. Simulations are increasingly being run on distributed systems,[1,2] or even on desktop computers.[3]

This raises the possibility of closing the simulation/analysis loop through the use of computational steering.[4] In its simplest form, by monitoring the current state of the simulation, decisions concerning appropriate choice of starting parameters can be made when it becomes clear that the simulation is not following the anticipated course. Dependent on the simulation, it may also be physically valid to adjust certain parameters part-way through a run. This leads to a number of productivity benefits - time-saving, what-if analyses, and a clearer picture of cause-effect relationships between the parameters and the data output.

There has also been a growth in the new science of visualization. With the increase in computer power has come an increase in data produced. 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.

Smoothed Particle Hydrodynamics (SPH)[5,6] 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

Further author information: (Send correspondence to Rick Walker)
Rick Walker: E-mail: rick.t.walker@gmail.com, Telephone: +44 (0)1227 827674
Peter Kenny: E-mail: p.g.kenny@kent.ac.uk, Telephone: +44 (0)1227 827236
Jingqi Miao: E-mail: j.miao@kent.ac.uk, Telephone: +44 (0)1227 823834

Visualization and Data Analysis 2007, edited by Robert F. Erbacher, Jonathan C. Roberts, Matti T. Gröhn, Katy Börner
Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 6495, 649509, © 2007 SPIE-IS&T · 0277-786X/07/$18

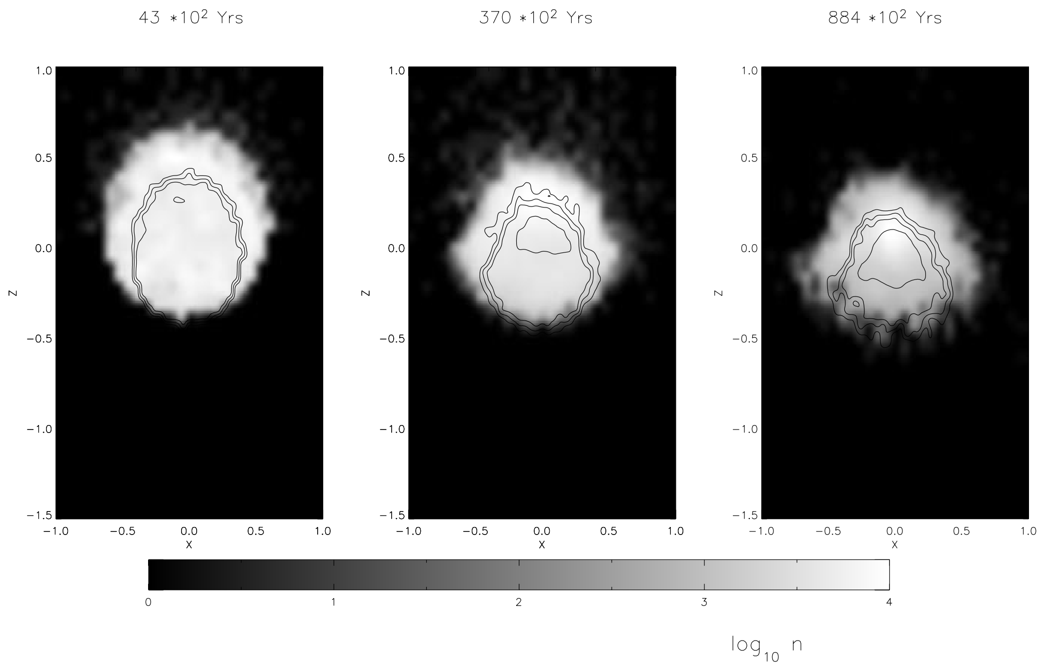

Figure 1. The evolution of the Hydrogen number density n (cm^{-3}) of the simulated molecular cloud from early stages t=4,300 years to t=88,400 years, when a flat core has formed in the front part of the cloud.[11]

move under the influence of hydrodynamic and, frequently, gravitational forces. Each particle of fluid has a fixed mass, and a density value which can change between time steps. Applications for this technique are centred around nonaxisymmetric phenomena in astrophysics - events such as asteroid impacts and collisions,[7] galaxy mergers,[8] supernova explosions[9] and star formation and evolution.[10] The output from SPH simulations consists of a point cloud, with each point having additional characteristics such as chemical composition and temperature.

This paper describes an exploratory simulation environment, under continuing development, for Smoothed Particle Hydrodynamics simulations in the field of astrophysics. Section 2 gives background to SPH and its uses, and looks at background material related to computational steering and visualization. In section 3 we discuss the necessary elements in an exploratory simulation environment for SPH, and introduce the concepts of painting and simulation trails. Section 4 provides brief implementation details.

2. BACKGROUND AND RELATED WORK

This section provides some background to the problem domain, by considering the SPH technique and its associated workflow in astrophysical research, then summarises work in the fields of computational steering and visualization of relevance to the solution of some of the limitations thus identified.

2.1. Smoothed Particle Hydrodynamics and Astrophysics

Grid or mesh-based numerical methods are widely applied to Computational Fluid Dynamics (CFD) problems - typically the Finite Difference Model (FDM) or the Finite Element Model (FEM). The deficits in these techniques are well understood: for FDM, difficulties arise in problems with free surfaces, deformable boundaries or moving interfaces. For FEM, extremely large deformation requires repeated mesh regeneration, which is very expensive computationally. This has led to the development of mesh-free particle based methods.[12]

With Smoothed Particle Hydrodynamics (SPH), the state of the system is represented by a set of particles, which possess individual material properties and move in accordance with the governing Lagrangian conservation equations. A particle represents a discrete quantity of material occupying a finite volume of space but without discrete boundaries. Each particle is influenced by its neighbours. The range of influence (its influence domain, or support domain) is determined by its *smoothing length* (chosen such that it includes a set number of other particles). Allocating a higher density of particles to a region of space produces more accurate results but at the cost of extra computation.

Resolution can be controlled at the start of a simulation through the initial distribution of particles but this distribution will change as the simulation evolves. Consequently, the final distribution may not give sufficient detail for features of interest. Attempts have been made to address this issue. Kitsionas and Whitworth[13] implement a particle-splitting method for astrophysical simulations in 2D, while Lastiwka et al.[14] describe a more general framework for adding and removing particles in other applications of SPH.

The SPH method has undergone continuing development.[15] The modelling of new phenomena or features may require the addition of additional elements to the underlying model (for example, from Nelson et al.[16] to Miao et al.[11] saw the addition of ionising radiation transfer, ionisation heating and recombination cooling and cooling by collisionally excited line radiation).

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. Fig. 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 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 overlooked.

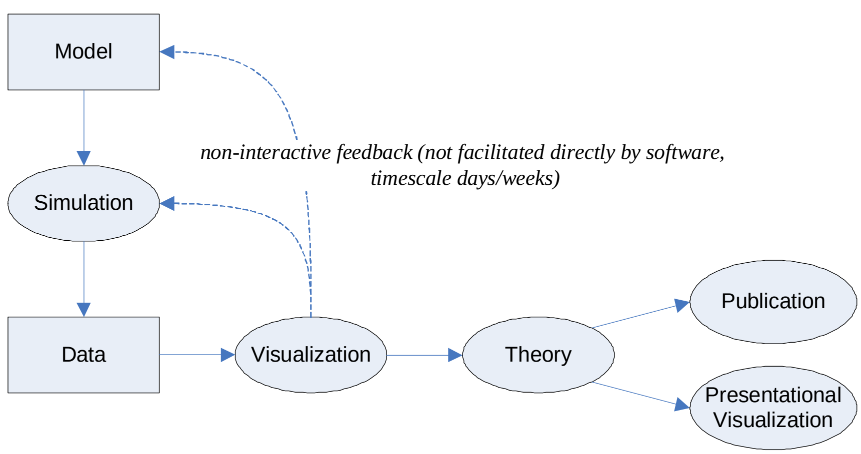

Figure 2. Previous research workflow (typical)

The previous research workflow is shown in Fig. 2. Multiple runs of the simulation are required to establish the input parameters that produce the desired results, and visualization must take place after each run to determine if the resultant data exhibits the correct characteristics. Further visualization may take place at a later stage to produce more presentational views.

2.2. Computational Steering

Computational steering has been an active area of research for well over a decade.[17] There are two main potential uses in exploratory simulation for SPH:[18]

- **Model Exploration:** the goal here is to see how the (often numerous) input parameters influence the simulation behaviour, and to encourage the user to investigate the results of different scenarios, and thus move towards a better understanding of the model. Additionally, changes to the model (for example, the addition of extra physical properties) can be examined.
- **Performance Optimisation:** by directing the effort in the computational process towards those areas of particular interest (and, conversely, reducing the effort in areas of little interest), the efficiency of the simulation can be improved.

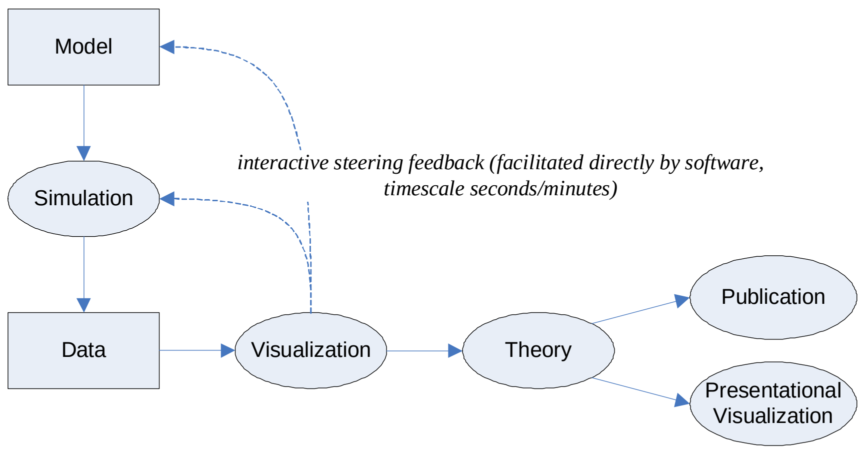

Figure 3. New research workflow with computational steering feedback loop

The workflow for computational steering is shown in Fig. 3. However, supporting computational steering is not often first on the minds of physical scientists when designing simulations. Since it spans the fields of simulation and visualization, it requires extended co-operation between researchers from different fields. To be useful, steering lag[19] (the time between a steering action occurring and the point at which the result of this action is visible) must be minimized, which in turn sets constraints on the computational time available.

2.3. Visualization for Steering

The addition of a visualization of the current state, and allowing the adjustment of parameters via dials and sliders, while an improvement on a batch-mode simulation, falls some way short of fulfilling the potential offered by computational steering.[20] Johnson et al.[21] describe computational steering as integrating the modelling, computational, data analysis, visualization and data input components of a scientific simulation. Chatzinikos and Wright[22] suggest that combining the visualization and data input components together (by taking the input directly from the visualization) can provide an interface more conducive to exploration. Related to this, van Wijk et al.[23] use Parameterized Graphics Objects to allow the design of visual interfaces to parameters.

There remains, too, a gulf between visualizations as produced by visualization researchers[24] and visualization as used by scientists.[11] Meissner et al.[25] discuss this issue in the context of medical data and posit that better integration into the existing workflow would help address this disparity.

3. EXPLORATORY SIMULATION

In this section, we describe the steps that must be taken to support exploratory simulation for astrophysics using the SPH method, aiming to address existing deficiencies in the three areas highlighted in Section 2.

3.1. Visualization

It is clear that driving the simulation through visualization is a desirable end product. So we must consider how best to visualize the output from the simulation at each time step. While the system could provide one, fixed view type for exploration, different features in the data set may be revealed by different views. Further, a new, unfamiliar view on its own is unlikely to prove attractive to scientists already accustomed (and trained to recognise features in) a different view. To bridge this gap, we propose the use of linked, multiform views.[26,27] By providing a familiar view alongside newer, more advanced ones, the scientist can learn to recognise features in other views. It may be that, after a training period, the older types of view can be phased out.

3.2. Interactivity

With a visual interface in place, it falls next to establish meaningful control of the simulation through steering. Chief amongst our concerns is the issue of steering lag. If steering actions have no obvious effect on the simulation in a timely fashion, then it becomes hard to see how a user gains any additional insight into the simulation. The goal, then, is interactivity.

By running the simulation and visualization in separate threads, we attenuate the problem somewhat - time taken for the simulation to produce results for the next time step is not all wasted, since visual exploration of the data can continue - but it is nevertheless important to advance the simulation quickly. With large data sets, this is problematic - in additional to the obvious increase in computation required for a larger number of particles, as the particle density rises, the smoothing length decreases and the time step for integration must decrease to satisfy the Courant-Friedrich-Levy[28] (CFL) stability condition. So new data takes longer to compute, and represents a smaller time difference when more particles are present in the simulation. Yet with only small numbers of particles, the simulation may lack sufficient resolution to resolve features.

A multiresolution approach to the simulation is one way forward. By running two simulation threads, a low resolution "preview" and a higher resolution "update", the simulation can appear to react quickly to user input, then as the user stops and explores the data produced, the views provided are updated as more detailed results become available. This approach is inspired by progressive encoding as used in the JPEG image format.[29]

3.3. Direct Steering

By using an adaptive SPH method, as discussed previously in [subsection above], it also becomes possible to direct computational effort to areas of space. Instead of a standard evolution of particle densities, the user can use their domain-specific experience to decide which areas are of interest, and thus require higher resolution. Conversely, there may be areas where only minimal resolution is required, and particles can be merged. What is required is some way of selecting these areas directly in the visualization. If we limit ourselves to direct visualization (for example, splatting) of the simulation data, then the problem would be one of 3D selection.[30] We propose a different approach.

3.3.1. Painting

While it is possible to allow selection of individual particles for splitting, with large data sets this would quickly become tedious. Further, since the particles in question will change position, this does not guarantee resolution in an area of space for a significant number of time steps. Selection of areas of space is required, and this can be accomplished through a painting metaphor.

With a simple brush tool and palette, the user can mark areas in views with one of three colours: red for areas where greater detail is required, amber for areas of slight interest, and green for areas of little interest. Each colour will denote a different requested particle density in that area, relative to the total number of particles in the simulation. Painting on any view can cause an update in other views - for example, colouring areas of a 2D contour plot would cause the corresponding areas of a 3D point-splatting view to become shaded also. Likewise, using a ray-casting approach to select parts of a 3D isosurface would cause updates to the point splatting view, but not necessarily to contour plots.

This approach offers other benefits. In addition to visualization of the simulation data, we can visualize derived data. One example of this is the Jean's Mass, which is used as the selection criteria for particle splitting in Kitsionas and Whitworth.[13] By presenting a visualization of data such as this alongside the other views, the simulation can offer suggestion maps - areas that it considers may be of special interest - to the painting process.

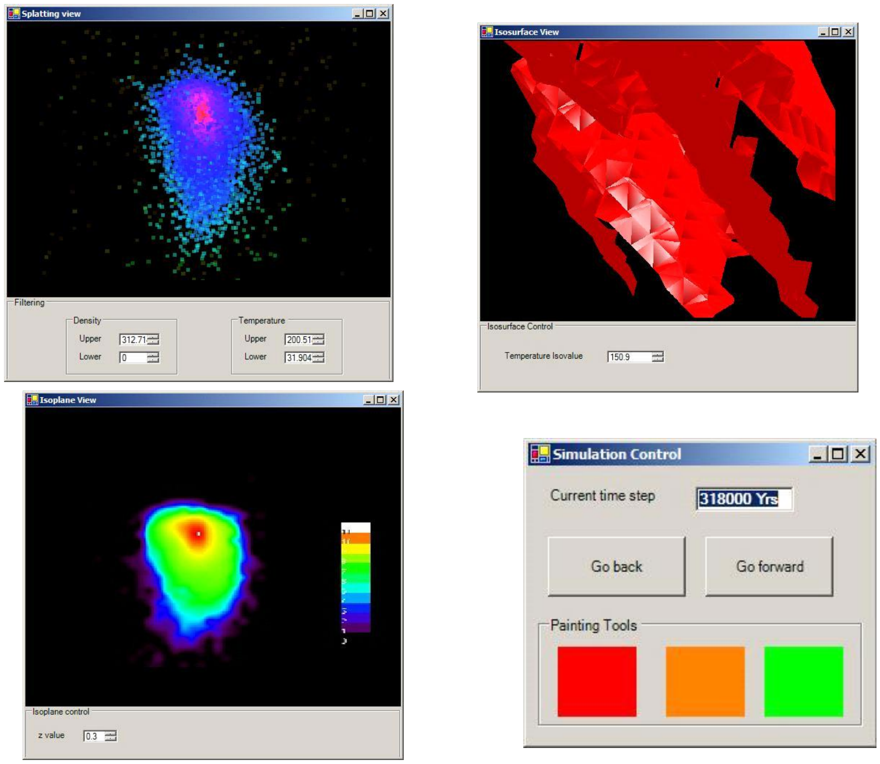

Figure 4. An example screenshot of the application

3.3.2. Simulation Trails

Applying the painting technique over the course of a simulation establishes a simulation trail. This shows which areas were marked for greater detail, and at what point in time. The trails taken on different simulation runs can be compared to discover a more optimal route to feature resolution. More importantly, with the trail already mapped out, the simulation can be repeated with a far greater number of particles in a non-interactive fashion. By providing the user with a way to explore different trails in an interactive fashion, we reduce the time taken to produce the final, high resolution simulation results: instead of ten separate high particle count runs with different parameters, a trail can be mapped once and followed. This represents a huge potential gain in productivity.

4. WORK COMPLETED

The SPH code developed by Nelson et al.[16] was taken as the basis in the development of an exploratory simulation.

4.1. Implementation Language and APIs

The original code was written in FORTRAN77 - still widely used in astrophysical simulations. This version of FORTRAN lacks support for dynamic data structures, multithreading and object-oriented capabilities such as inheritance. In fact, a change to particle numbers in the simulation required recompilation of the program. This made the implementation unsuitable for use with adaptive SPH. The simulation was re-engineered as a C++ application, using OpenGL[31] and the wxWidgets[32] toolkit used for graphical interface components.

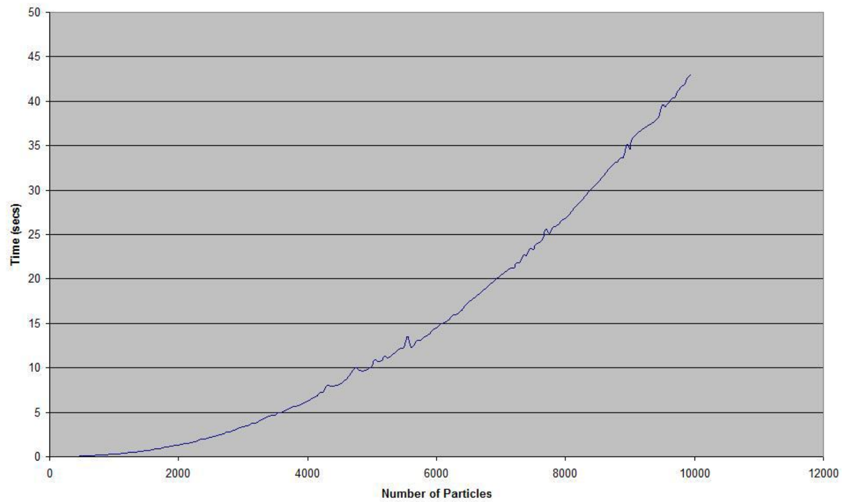

Figure 5. Execution time to reach a particular stage in a simulation vs number of particles - when adjusted for variable time step size, the computational effort required increases roughly quadratically with number of particles. The hardware used was an Intel Pentium 4 running at 3.60GHz.

4.2. Visualization

By combining simulation and visualization, it was possible to use the same data structures in both components - in particular, the octree used in the simulation for nearest neighbour calculations proved a convenient way of storing the particles for visualization. For initial investigations, the views provided were a 3D point-splatting view for overview, an isosurface produced using the Marching Tetrahedra algorithm,[33] and an isoplane. The simulation itself was extended to support a new SPH splitting technique, along the same lines as Kitsionas and Whitworth.[13]

While a choice of three views is offered, the combinations afforded are more numerous. For example, two views could contain different isoplanes, while a third shows both planes in the context of a full 3D view. The same view can be duplicated and rotated, and then linked, to look at the same data from a different angle at the same time. Views can be combined in the same window. A sample of each of the views is shown in Fig. 4.

4.3. Interactivity

A progressive simulation model has been implemented using multithreading. One thread runs a low-resolution simulation (with fewer particles) to provide a rapid preview with interactive steering while another runs a higher-resolution simulation (with more particles) for the final results. Two factors are of key importance in providing an interactive environment - the time taken to advance the preview, and the difference between this and the time to advance the final (high resolution) view to the same point. Since the particle densities differ in these two simulations, the CFL condition requires a smaller time step size for the final view, and thus advancing it to the same time point as the preview will require multiple simulation steps.

The optimum particle numbers in each thread will vary according to the hardware specification - Fig. 5 shows how time required to reach the same point varies as particle count increases.

4.4. Direct Steering

Painting has been implemented for both isoplanes and isosurfaces, with the 3D view updating to show painted areas but not acting as a canvas for painting itself. Other views show painted areas if applicable - for example, painting on an isoplane will not change the view of a different isoplane unless they intersect. Areas painted in the previous time step are shown on the new data initially, providing a basis for further changes.

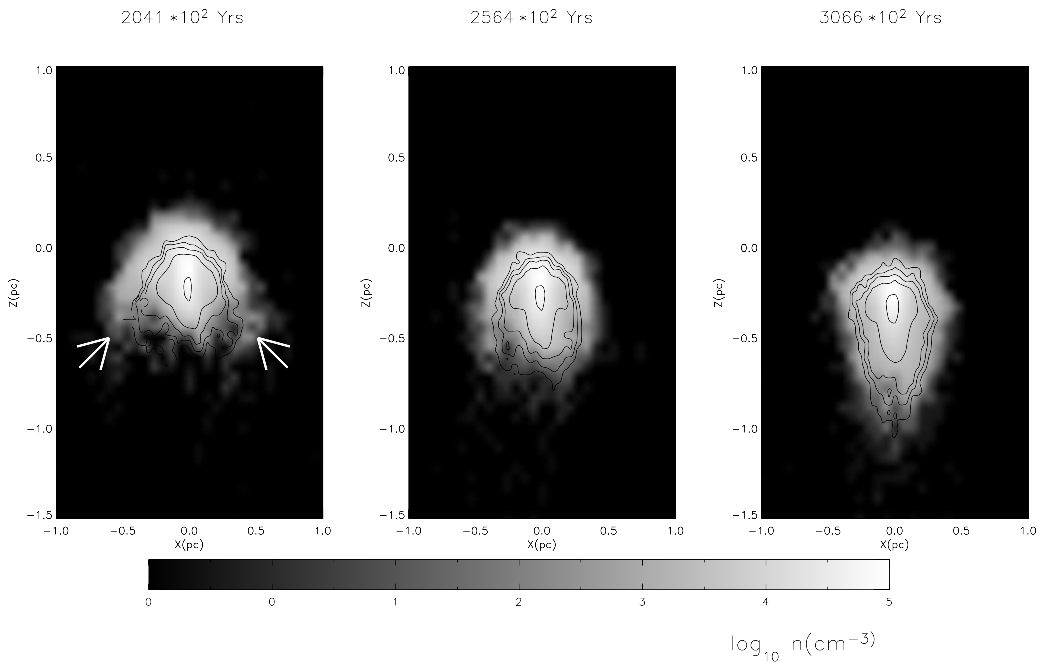

Figure 6. The evolution of the number density n of the simulated molecular cloud over the period of t=204,100 years to t=306,600 years, when the core gradually becomes elongated. The white arrows point to the location of the 'ear-like' structures.[11]

The areas selected in each colour for each time step are stored as well as fed back into the simulation. Trails can be saved at any point in the run, and a new simulation run started with any saved trail. Trail comparison has also been implemented - multiple trails can be overlaid on the same views, to highlight differences in path.

4.5. Example of Application

Fig. 6 shows density contours for advancing time steps in a recent 10,000 particle SPH simulation,[11] of the Eagle Nebula (M16). This is a HII region,[34] composed of clouds of ionized atomic hydrogen. At the point t=204,100, the arrows indicate the formation of ear-like structures in the cloud. However, the contours in this region are unclear and sketchy. This is a direct result of lack of local resolution - the centre regions of the image (the 'nose', discussed along with the ears in Miao et al.[11]) are well resolved because the particle density is much higher.

By running the experiment in our exploratory simulation environment, it should be straightforward to address this issue by painting the ears red to increase the local particle density at an early stage. Since, as can be seen from the later plots in Fig. 6, the ears are a temporary feature, the painting action could be cleared subsequently. This would require substantially less computation than running the entire simulation with a higher density of particles throughout, which represents a clear advantage over the previous approach. However, our code does not yet include the additional physics to model ionising radiation transfer, ionisation heating and recombination cooling and cooling by collisionally excited line radiation, and thus is currently unable to duplicate the results. It is anticipated that the new exploratory simulation environment will be completed in the near future, at which point a real comparative study can be conducted to determine the true effectiveness of the techniques proposed in this paper.

5. CONCLUSIONS AND FUTURE WORK

Computational steering techniques such as painting and simulation trails have the ability to aid research efforts in astrophysics by reducing the time needed to conduct simulations and analyse results. By using adaptive SPH formulations, effort can be directed where it is required. Multiform visualization can help through integration with workflow, by providing an easily extensible test system for new, more advanced, visualizations, and by providing an overview in addition to detail in individual views.

The obvious next steps with this work are to continue development of the simulation to mirror the current state as used in astrophysics research. At this point, the system will become useful both for current experiments, and for model development - future physical model additions can be studied in an interactive fashion, to see if they match observations without a large amount of time-intensive computations.

Defining a trail in the current system can be very time-consuming if the desired volumes do not lie on an isoplane or isosurface. There are a number of 3D selection[35] and sketching[36,37] techniques that could be implemented to streamline this process. One possible technique would be to allow direct selection via points in the 3D view. Selecting two points could then add to the trail the volumes of space in which each point is contained, and also those volumes which contain an interpolated line, of specified thickness, between the two points. Addition of a brushing metaphor to allow users to view point data could aid this process.

For exploration of the data, the current filtering system is functional. However it may be more intuitive to provide a direct, visual method - for example, by selecting two points and using their values for the upper and lower filtering bounds, or by allowing selection on a colour scale for a window without requiring knowledge of the actual values of the variables.

Other possible research directions include harnessing the power of a GPU to increase the speed of the simulation. Li et al.[38] and Scheidegger et al.[39] demonstrate that significant speed-gains that can be achieved by harnessing dedicated vector processors for CFD simulations. Thus far, attention in this area has focussed on grid-based techniques, but there may also be speed gains for meshless approaches.

Beyond that, this study aims to provide an example of the way computational steering through visualization can be used in astrophysics, and with further cooperative research between computer scientists and astrophysicists, it is hoped that its usefulness will increase.

REFERENCES

1. T. Hauser, T. I. Mattox, R. P. LeBeau, H. G. Dietz, and P. G. Huang, "High-cost CFD on a low-cost cluster," in *Supercomputing '00: Proceedings of the 2000 ACM/IEEE conference on Supercomputing (CDROM)*, p. 55, IEEE Computer Society, (Washington, DC, USA), 2000.
2. R. Fatoohi and S. Weeratunga, "Performance evaluation of three distributed computing environments for scientific applications," in *Supercomputing '94: Proceedings of the 1994 ACM/IEEE conference on Supercomputing*, pp. 400–409, ACM Press, (New York, NY, USA), 1994.
3. Comsol, "Femlab." http://www.comsol.com.
4. R. Van Liere, J. D. Mulder, and J. J. van Wijk, "Computational steering," *Future Generation Computer Systems* **12**(5), pp. 441–450, 1997.
5. R. A. Gingold and J. J. Monaghan, "Smoothed particle hydrodynamics - Theory and application to non-spherical stars," *Monthly Notices of the Royal Astronomical Society* **181**, pp. 375–389, nov 1977.
6. J. J. Monaghan, "Smoothed particle hydrodynamics," *Annual Review of Astronomy and Astrophysics* **30**, pp. 543–574, 1992.
7. D. D. Durda, W. F. Bottke, B. L. Enke, W. J. Merline, E. Asphaug, D. C. Richardson, and Z. M. Leinhardt, "The formation of asteroid satellites in large impacts: results from numerical simulations," *Icarus* **170**, pp. 243–257, July 2004.
8. C. Mihos, "Dynamics of Mergers," *Astrophysics and Space Science* **266**, pp. 195–205, 1999.
9. V. Bromm, N. Yoshida, and L. Hernquist, "The First Supernova Explosions in the Universe," *Astrophysical Journal Letters* **596**, pp. L135–L138, Oct. 2003.

10. K. Nagamine, V. Springel, and L. Hernquist, "Star formation rate and metallicity of damped Lyman α absorbers in cosmological smoothed particle hydrodynamics simulations," *Monthly Notices of the Royal Astronomical Society* **348**, pp. 435–450, Feb. 2004.
11. J. Miao, G. J. White, R. Nelson, M. Thompson, and L. Morgan, "Triggered star formation in bright-rimmed clouds: the Eagle nebula revisited," *Monthly Notices of the Royal Astronomical Society* **369**, pp. 143–155, June 2006.
12. G. R. Liu and M. B. Liu, *Smoothed Particle Hydrodynamics a meshfree particle method*, World Scientific Books, 2003.
13. S. Kitsionas and A. P. Whitworth, "Smoothed Particle Hydrodynamics with particle splitting, applied to self-gravitating collapse," *Monthly Notices of the Royal Astronomical Society* **330**, pp. 129–136, Feb. 2002.
14. M. Lastiwka, N. Quinlan, and M. Basa, "Adaptive particle distribution for smoothed particle hydrodynamics," *International Journal for Numerical Methods in Fluids* **47**(10-11), pp. 1403–1409, 2005.
15. J. J. Monaghan, "Smoothed particle hydrodynamics," *Reports of Progress in Physics* **68**, pp. 1703–1759, Aug. 2005.
16. R. P. Nelson and W. D. Langer, "The Dynamics of Low-Mass Molecular Clouds in External Radiation Fields," *Astrophysical Journal* **482**, pp. 796–+, June 1997.
17. R. Marshall, J. Kempf, S. Dyer, and C.-C. Yen, "Visualization methods and simulation steering for a 3D turbulence model of Lake Erie," in *SI3D '90: Proceedings of the 1990 symposium on Interactive 3D graphics*, pp. 89–97, ACM Press, (New York, NY, USA), 1990.
18. J. D. Mulder, J. J. van Wijk, and R. van Liere, "A survey of computational steering environments," *Future Gener. Comput. Syst.* **15**(1), pp. 119–129, 1999.
19. E. Kraemer, D. Hart, and G.-C. Roman, "Balancing Consistency and Lag in Transaction-Based Computational Steering," in *HICSS '98: Proceedings of the Thirty-First Annual Hawaii International Conference on System Sciences-Volume 7*, p. 137, IEEE Computer Society, (Washington, DC, USA), 1998.
20. H. Wright, "Putting Visualization First in Computational Steering," in *Proceedings of UK e-Science All Hands Meeting*, S. Cox, ed., pp. 326 – 331, AHM2004, EPSRC, 2004.
21. C. Johnson, S. G. Parker, C. Hansen, G. L. Kindlmann, and Y. Livnat, "Interactive Simulation and Visualization," *Computer* **32**(12), pp. 59–65, 1999.
22. F. Chatzinikos and H. Wright, "Computational Steering by Direct Image Manipulation," in *VMV '01: Proceedings of the Vision Modeling and Visualization Conference 2001*, pp. 455–462, Aka GmbH, 2001.
23. J. J. van Wijk, R. van Liere, and J. D. Mulder, "Bringing Computational Steering to the User," in *Dagstuhl '97, Scientific Visualization*, pp. 304–313, IEEE Computer Society, (Washington, DC, USA), 1999.
24. M. Hopf, M. Luttenberger, and T. Ertl, "Hierarchical Splatting of Scattered 4D Data," *Computer Graphics and Applications* **24**(4), pp. 64–72, 2004. submitted version.
25. M. Meissner, B. Lorensen, K. Zuiderveld, V. Simha, and R. Wegenkittl, "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*, pp. 575–578, IEEE Computer Society, 2002.
26. J. C. Roberts, "Exploratory Visualization with Multiple Linked Views," in *Exploring Geovisualization*, A. MacEachren, M.-J. Kraak, and J. Dykes, eds., Amsterdam: Elseviers, December 2004.
27. R. Walker, P. Kenny, and J. Miao, "Visualization of Smoothed Particle Hydrodynamics for Astrophysics," in *Theory and Practice of Computer Graphics 2005*, L. Lever and M. McDerby, eds., pp. 133–138, Eurographics Association, (University of Kent, UK), June 2005. (Electronic version http://diglib.eg.org).
28. R. Courant, K. O. Friedrichs, and H. Lewy, "On the partial difference equations of mathematical physics,," *IBM Journal of Research and Development* **11**, pp. 215–234, 1967.
29. G. K. Wallace, "The JPEG still picture compression standard," *Commun. ACM* **34**(4), pp. 30–44, 1991.
30. B. D. Conner, S. S. Snibbe, K. P. Herndon, D. C. Robbins, R. C. Zeleznik, and A. van Dam, "Three-dimensional widgets," in *SI3D '92: Proceedings of the 1992 symposium on Interactive 3D graphics*, pp. 183–188, ACM Press, (New York, NY, USA), 1992.
31. OGL, "OpenGL." OpenGL Architecture Review Board, http://www.opengl.org.
32. wxWidgets, "wxWidgets Cross-Platform GUI Library." http://www.wxwidgets.org/.

33. A. Doi and A. Koide, "An Efficient Method of Triangulating Equivalued Surfaces by using Tetrahedral Cells," *IEICE Transactions Commun* **E74(1)**, pp. 214–224, January 1991.
34. J. Franco, G. Tenorio-Tagle, and P. Bodenheimer, "On the formation and expansion of H II regions," *Astrophysical Journal* **349**, pp. 126–140, Jan. 1990.
35. C. Hand, "A survey of 3d interaction techniques," *Computer Graphics Forum* **16**(5), pp. 269–281, 1997.
36. T. Igarashi, S. Matsuoka, and H. Tanaka, "Teddy: A sketching interface for 3d freeform design.," in *SIGGRAPH*, pp. 409–416, 1999.
37. E. Sachs, A. Roberts, and D. Stoops, "3-draw: A tool for designing 3d shapes," *IEEE Comput. Graph. Appl.* **11**(6), pp. 18–26, 1991.
38. W. Li, X. Wei, and A. Kaufman, "Implementing lattice Boltzmann computation on graphics hardware," *The Visual Computer* **19**, pp. 444–456, Dec. 2003.
39. C. E. Scheidegger, J. L. D. Comba, and R. D. da Cunha, "Practical CFD Simulations on Programmable Graphics Hardware using SMAC," *Computer Graphics Forum* **24**(4), pp. 715–728, 2005.