Interactive 3-D visualization of particle systems with Partiview

Stuart Levy

National Center for Supercomputing Applications, University of Illinois Urbana-Champaign, 605 E. Springfield, Champaign, IL 61820

Abstract. This paper describes partiview, a free software tool for interactive graphical display of collections of particles in 3-space, and its application in studying the results of N-body collisional stellar dynamics calculations from Starlab.

1. Partiview

Partiview is an interactive graphical software tool, focused on flexible display of particles in 3-space.

Input data to partiview is provided as a (possibly time-varying) collection of particles, each with a 3-D position and an arbitrary number of other floating-point attributes (*fields*), and a configuration script specifying which fields to map into visible properties, including color and luminosity. Text-based commands can change these selections interactively, switching between coloring stars by e.g. temperature, mass, or space velocity on the fly. For example, if a field were named Tlog, `color Tlog 3.2 4.5` assigns colors by using the Tlog field as an index into a user-supplied color table via a linear mapping that associates 3.2 and 4.5 with the colormap's endpoints.

From each particle's luminosity and distance from the current viewpoint, partiview draws a dot whose screen brightness and size suggest its computed apparent brightness. With dots up to a few pixels across, apparent brightness may usefully range by several hundredfold, and larger ranges can be suggested by adding textured polygons – "haloes" – whose size varies similarly. The result is good enough to yield plausible naked-eye starfields given a table of stellar luminosities, colors and 3-D positions as in Figure 1, drawn using Hipparcos data.

This kind of viewpoint-dependent apparent brightness is a feature that few other scientific visualization packages seem to offer, even though it's inexpensive to compute and can be useful. (Where not useful, as when making orthographic plots of 3-D scenes, it can be switched off in partiview.)

Some database-like operations are provided. For example, one can display only the subset of particles where some (single) field has values in a given range or set, or look only at particles lying within a given rectangular subvolume, Also one can print a histogram of values of a field, over all particles or the selected subset.

Figure 1.   Desktop version of partiview showing star field from Hipparcos data with Sun at upper left, marked by 0.1 pc crosshair. Normally displayed in light colors on a dark background, this and all images in this article are rendered as black-on-white for publication.

2. Scripting

Partiview can be controlled by external scripts in a limited way, by invoking a subprocess which emits a series of partiview commands. This allows writing, for example, animation-recording programs which drive partiview to display some computed sequence of views and record each image to a file. It may be adequate for creating simple graphical controls for features of partiview that aren't provided on the existing graphical panel. The major limitation is the lack of feedback: there's no way for external driver programs to read out the viewer's state.

A future version of the software may have an embedded scripting language, perhaps Python or Ruby. This should make it easier to offer a variety of graphical interfaces (or to change the graphical interface dynamically), to couple the viewer more easily with external data-mining tools, and so on.

3. Desktop and dome

The same graphical and data-handling code is embedded in multiple guises for different computing environments. Both accept the same data and configuration files, and most of the same text-based interactive commands. Figure 1 illustrates the desk- (or lap-)top version, mouse and keyboard driven with conventional buttons and sliders for common controls, available for Unix-like systems and for Windows. Figure 2 shows the virtual-reality version, built using the Virtual
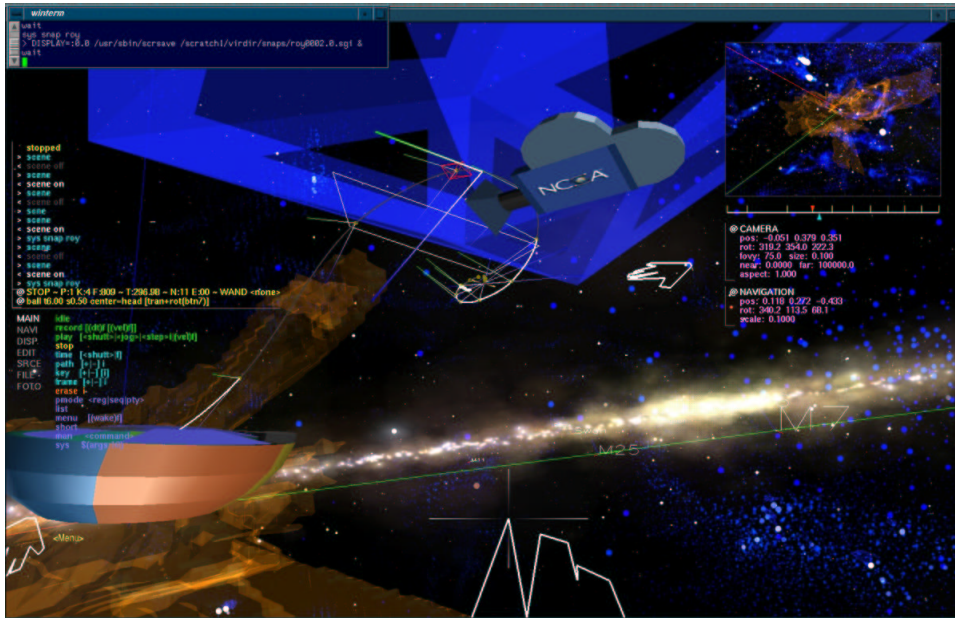
Figure 2.   Virtual Director version: collaborating between Hayden Planetarium (dome avatar at lower left) and NCSA while designing an animation path in a simplified Milky Way model.

Director virtual-choreography framework <http://virdir.ncsa.uiuc.edu/virdir/> and the CAVE library <http://www.evl.uic.edu/pape/CAVE/>; it is currently restricted to Silicon Graphics computers but, unlike the desktop implementation of partiview, it can run on systems with multiple graphics pipes.

Though the latter was originally written for the CAVE virtual reality room at NCSA, it is used elsewhere as well. The Hayden Planetarium at the American Museum of Natural History in New York built a Silicon Graphics-driven display for their planetarium dome; this turned out sufficiently CAVE-like that the same software runs in the Hayden dome and is regularly used there.

The Virtual Director framework supports networked collaboration too, allowing widely-separated participants to exchange viewpoints (represented by "avatars"), animation paths, display settings and so on. This has proven helpful for bringing together distributed expertise, and we've used it between Illinois, Hawaii, New York and elsewhere on several occasions, most recently when designing some animations for Hayden's 2002 space show, as seen in Figure 2.

4.   N-body dynamics: examining Starlab traces

Stellar dynamics simulations done in Starlab <http://www.manybody.org/> produce *traces* as output. Trace files record various information about each star as a function of time: physical properties such as mass, luminosity and temperature; position and three time derivatives; and hierarchical descriptions (binary trees) of interacting groups.

Figure 3.   Star cluster dispersing in a tidal field, with recent motion
shown by trails. Small circles denote binary or multiple systems.

Partiview, coupled with the Starlab libraries to read and interpolate traces,
is adapted to display these properties as the cluster evolves.

Figure 3 shows a cluster evolving in a tidal field. "Trails" show the recent
motion history of each star, so long trails show high speed, and curved ones high
acceleration. Note the stars escaping through the tidal tails at left and right.

5.   Microscopy

A sort of interaction microscope is shown in figure 4 on a tiny test case. Bound
or strongly-interacting systems are circled, and colored according to the number
of members of the group. Circle sizes are related to separation or semimajor axis,
but can be constrained to a range of screen sizes to ensure that even tightly-
bound systems are always visibly marked. The binary trees of interacting groups,
with stars as leaves and center-of-mass points as nonleaf nodes, are shown as
stick figures. Tick marks perpendicular to the tree branches, drawn in the screen
plane, serve a dual purpose: their positions show the center-of-mass location,
and their lengths are proportional to the instantaneous true separation of the
nodes on that branch.

Note the triple system in figure 4. Although in this view its lower-right
pair of stars seem very close together, this must be a projection effect; as shown
by the tick marks, their true separation is nearly as large as that from their
center-of-mass to the upper-left star.

Stellar cluster simulations, with time scales spanning many orders of magni-
tude, need some sort of temporal microscope too. Partiview has limited facilities
for this, with a logarithmic speed scale (simulation time per unit displayed time),

Figure 4.   Interacting groups of stars from Starlab.

keystroke commands to home in on interesting events, and a trip meter to mark a (single) time reference point.

6.   Sifting needles from a haystack

In all but the smallest systems, direct visual inspection quickly becomes an impractical way to study interaction patterns – too many stars clutter the view. But visualization might still be helpful, if there is a way to focus on a subset of the system that sheds light on the question at hand.

For example, to study some sequence of events – the ejection of a star from the cluster, or the formation of a hard binary – one might imagine focusing by looking at some limited neighborhood, e.g. all stars passing through some volume during some time interval. But, since starlab simulation traces explicitly mark closely-interacting stars, we can do better than that; we can use *interaction tracking* to pick out just the set of stars that pass close enough to be dynamically important.

To do this, the user constructs an initial small *selection-set* S of interesting stars at a given time – perhaps by a threshold of some field ("all stars with mass exceeding 30 M_{sun}", or "all triple systems", or "star number 1792"), or by clicking on some star, e.g. an escapee. This set is given to a `kira intsel` $T = S$ command, and the user plays the trace forward or backward in time for a while. When `intsel` is in effect, whenever any member of S interacts with another star, the software adds the latter star to the set T. If S and T refer to the same set, then that set accumulates the transitive closure of interactions with S.

Other commands can then use membership in T to affect the display: to show only stars which have or haven't interacted during the given time span; to show all stars, but brightening those in T; to draw trails behind those selected; etc.
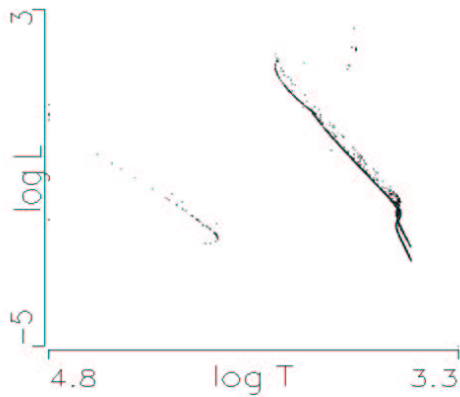
Figure 5.   H-R diagram

7.   Other views

Of course, plots of 3-D positions aren't the only useful way to display the state of a cluster. For Starlab output traces which record stellar temperature and luminosity, a Hertzprung-Russell diagram is also available, as shown in Figure 5. One can click on a star in either view and locate it in the other.

Other types of views are contemplated for future work; for example, world-line plots of two spatial dimensions plus time, or a velocity-space plot.

8.   Offerings

The desktop version of partiview is available as free software from the Starlab CVS archive; see Peter Teuben's web pages at:
<http://www.astro.umd.edu/nemo/amnh/>.

As described in Chandler (2002), Brian Abbott and Carter Emmart of the American Museum of Natural History have assembled a 3-D Milky Way model (nearby stars, open and globular clusters, H II regions, pulsars, etc.) from various sources and are offering it in partiview form, along with partiview itself for Linux and Windows, on the Hayden Planetarium web site:
<http://www.haydenplanetarium.org/hp/vo/partiview/index.html>.

9.   Acknowledgements

Thanks are due in many directions: to my colleagues Donna Cox and Robert Patterson at NCSA who've encouraged me to develop this; to them and other patient users, Emmart and Abbott of Hayden/AMNH; Brent Tully of U. of Hawaii; Piet Hut, Peter Teuben, Steve McMillan, Simon Portegies-Zwart, and Jun Makino of the Starlab group. All have offered data to visualize and plenty of good ideas, a few of which are implemented here. Special thanks are due to Teuben, who wrote much of the extant documentation, and to McMillan for the Starlab interface libraries. Thanks finally to the NCSA, which is supported in turn by the National Science Foundation and by the state of Illinois.

References

Hut, P. 2002, these proceedings.

Chandler, David L. 2002, S&TAugust 2002, 61