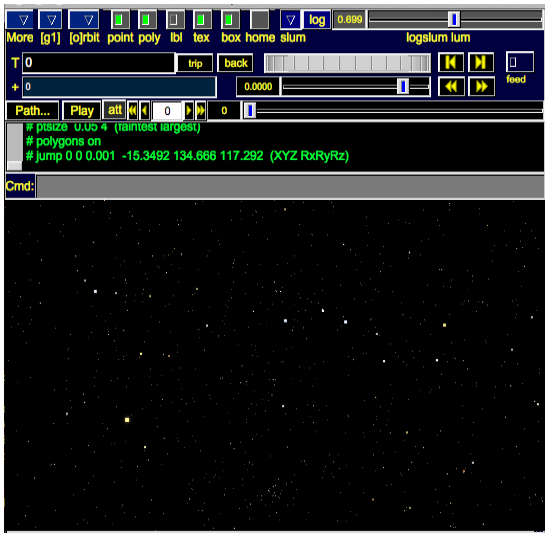
Partiview Examples

in the `partiview/examples` folder

hipmotion

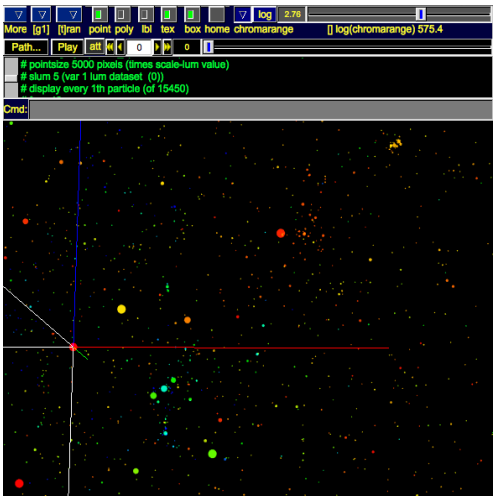


Nearby stars (from HIPPARCOS catalog) with space motions. Press the >> button to see the stars move at 10,000 years per second (or adjust the speed slider). Scripts and data are in the `data` folder – see "hipmotion", which reads "hipbrightv.speck".

In the **hipmotion** script, note the "lum", "color", "cmap" commands etc. to determine brightness/color; "polylumvar" and "polysize" to associate a screen-facing polygon with each star, proportional to its brightness; and "texturevar" and "texture" to apply a glow to each star. (Adjust brightness ("slum") slider, or click "tx" texture button off.)

Also note the "warp" command. "warp -extrap ..." changes the position of each star as a (linear, in this case) function of time, based on its initial position and a group of three data variables giving velocity.

hipchroma



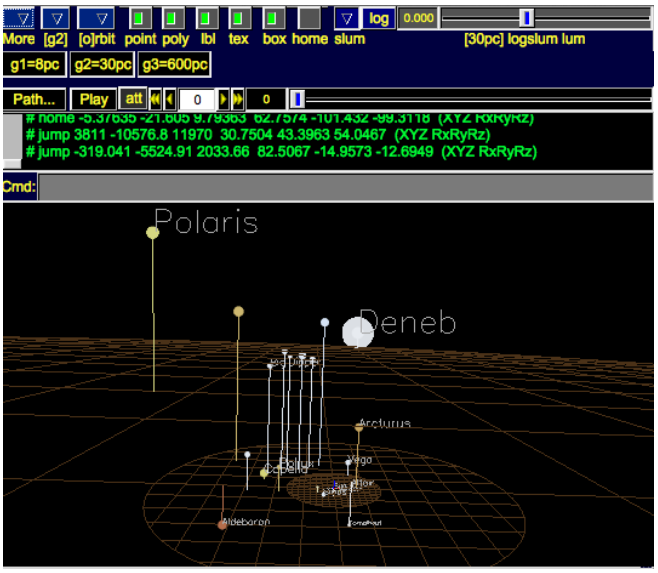
Nearby stars, color-coded by distance from the viewpoint. (Stars in a cluster look the same color!) Scripts/data are in the `data` folder – see "hipchroma". Thanks to Prof. Anthony Fairall for the idea.

Designed for Chromatek stereo glasses, which make red things look nearby, blue ones distant. But it's helpful even without those glasses.

Try adjusting distance-to-color mapping: click the "slum" pulldown-menu at left of the brightness slider to make slider control other things. Choose "chromarange", then drag the slider.

Try "chromadepth off" and "chromadepth on".

starwalk



If you could walk to the nearest star... Scripts/data in `starwalk` folder, starting with `starwalk.cf`. It defines three "groups": "8pc", "30pc" and "600pc", also known as g1, g2, g3. Partiview creates a button for each – you can switch them separately on and off. It starts off with just the "8pc" group of nearest stars visible.

Note the label text (inside the starwalk/stars8.cf, .speck, etc. files), and the grid mesh and vertical spikes for each star, made from partiview "mesh" objects, generated by a program – the perl script "speckv2galmap".

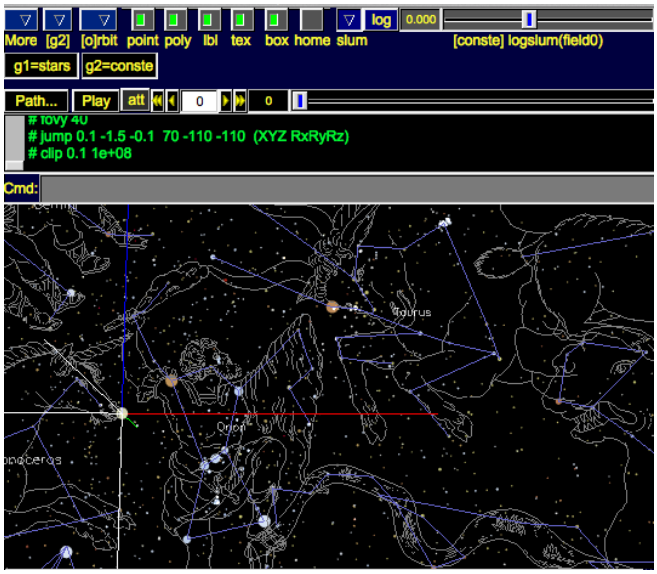
As seen from Earth, three stars – Deneb, Vega and Altair – look about equally bright. (Try flying to the origin crosshair, or click the "home" button.) But from elsewhere, we see that Deneb is a luminous supergiant – it's impressively bright over a large area, while Vega and Altair look small if we draw back a few tens of parsecs.

Star disk areas (polylumvar … area) are proportional to each star's intrinsic luminosity, which makes their apparent brightness fade in the proper way with distance.

See also the writeup in: **starwalk/handout.pdf**

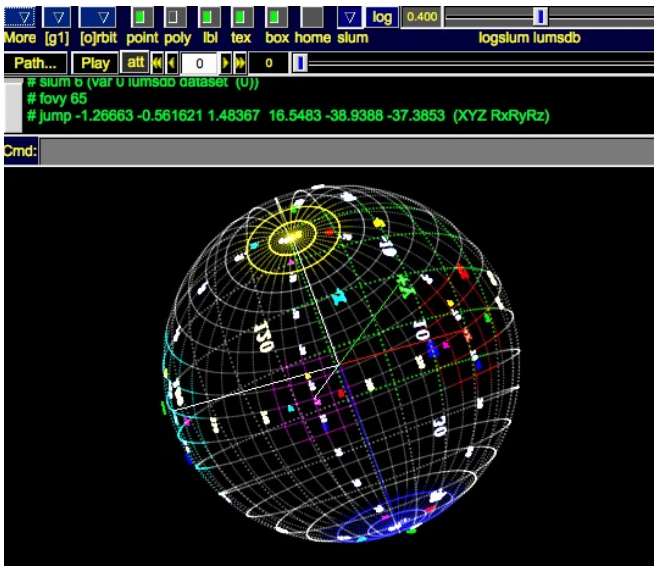
Guy Ottewell's book, *Astronomical Companion,* inspired this example.

constellations



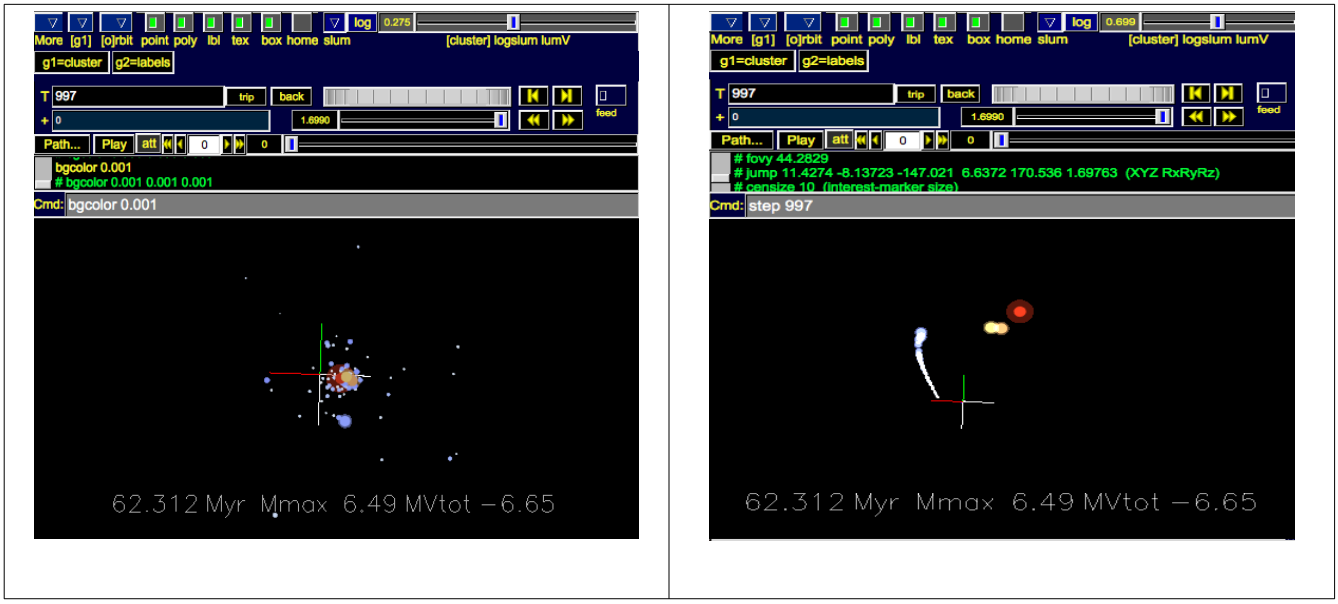
Pretty constellation pictures. Provided by Toshiyuki Takahei, National Observatory of Japan.

spheregrid



Handy spherical marker, with text made entirely of dots. Demonstrates .sdb binary particle file format. See spheregrid/mkspheregrid for the perl script which created the data.

cluster



An evolving cluster of stars. Two views of the same data: on the left, the stars' current positions; On the right, the "Hertzsprung-Russell" diagram, which plots the same stars on a 2-D graph of temperature (color) vs. luminosity. Press the >> key to watch the cluster evolve through 400 million years.

Uses the "warp" command to remap the positions of stars. Use "**warp on**" in the command box to see the temperature-luminosity view, or "**warp off**" to switch to the 3-D spatial position view.
Used this way, "warp" lets you replace any particle's coordinates with any linear combinations of its data attributes.

Uses the .pb binary particle format, with each timestep in its own file, and a .speck file (cluster/clu8f/stars.speck) which loads each one into its corresponding partiview timestep (pb -t …).

Also uses a second group to carry labels. The labels are made to stick to the screen, rather than moving in 3-D when you move the viewpoint, with the special "tfm camera" notation.

Stellar evolution was calculated with Bill Paxton's wonderful "EZ" code (its present-day descendant is MESA) and cluster dynamics with Peter Teuben's "NEMO".

galride

A very simplified model of stars moving in a galaxy, using epicycle orbits (!) and similar oscillatory motions rather than N-body gravity. Disk-star orbits were seeded with the distribution of velocities from HIPPARCOS stars near the Sun. (I just made up some bulge orbits somehow.)

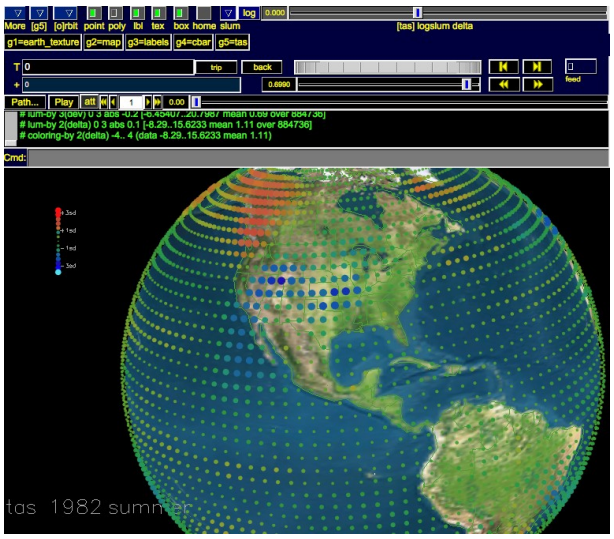
Inspired by an animation AVL made with Adler Planetarium about the IBEX satellite, which studies the boundary between the heliosphere and the galactic environment. The heliosphere's size varies as the Sun cycles through its merry-go-round galactic orbit, oscillating above and below the Galactic plane.

To start/stop the animation, click the >> or << buttons. Adjust speed with the adjacent slider.

There are two display modes: following the "Sun" (all motions relative to its orbit), or, in an inertial frame in which all stars orbit about the Milky Way center. To switch between modes, type in the partiview text box:

read rideOn.cf – ride along with the Sun
read rideOff.cf – inertial frame

gcm



Presentation of a century's worth of monthly data from a global climate model, run as of about 2001. The data came from Katharine Hayhoe (now of Texas Tech).

It's included here as a non-astronomical example, and to show that it's possible to include surfaces with images on them (the Earth) in a partiview scene. It's also demonstrating that visualizing renormalized data – subtracting out some trends in order to show others – can be a cool thing to do.

The dataset here is surface temperature (tas), averaged over “summer” (Jun-Aug). Each pixel corresponds (in color and brightness) to the average of that quantity over that interval forthe given year. But rather than having absolute brightness/color scales, each one is scaled by its deviation from the mean, at that point, over a given reference period -- generally 1981-1990 + 2001-2010.

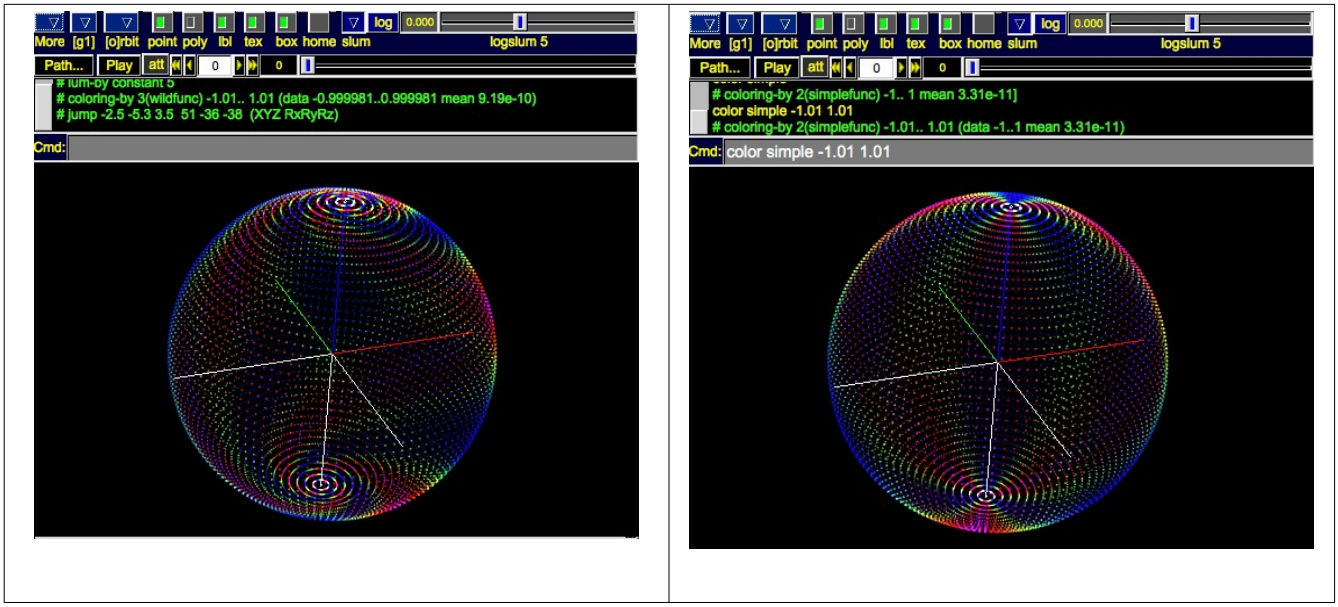
Data variables are

- val - actual value, of precip or temp or cloudiness
- dev - deviation from that point's distribution over reference time, as described above
- gdev - deviation from global mean
  (over all points during that season during reference period)
- tdev - deviation from mean for that point over all seasons of ref period

- meanXX - reference-period mean for that point
- sdXX - reference-period std dev for that point

For some things to try, see `partiview/examples/gcm/HOWTO.txt`

make-your-own-data



Two python scripts, “make-specks.py” and “make-pbs.py”, make equivalent datasets in two different formats – .speck (human-readable text) and .pb (a binary particle format, more efficient for large data). They might serve as examples for making your own data.

From a command line, run

```
partiview sphere-example.cf
```

or try re-creating the data by running

```
python make-specks.py    (or)
python make-pbs.py
```

Four variables, “lon”, “lat”, “simplefunc” and “wildfunc” are defined at each point. Use “color” or “lum” or “thresh” or etc. on one or another variable, e.g.:

```
datavar             # list variables (attributes), with value ranges
color lat           # color by latitude, over its full data range
thresh wild > 0.5 # select just those particles
see all             # undo thresholding, see everything again
```