Galaxy Supercluster Analysis and Visualization: Project Goals

Jameson Miller, Cory Quammen - UNC-CH, Computer Science
Matt Fleenor, Jim Rose - UNC-CH, Physics

May 10, 2005

1 Problem Domain

Determining the large-scale structure of galaxy distribution in the local universe is currently an active research area in cosmology. Features called *superclusters*, which are global overdensities of galaxies, are of particular interest. Superclusters are comprised of a collection of galaxy clusters, which are smaller-scale and more dense regions of galaxies, along with the regions between these clusters, known as inter-cluster regions. It is currently unknown if the galaxies between clusters, or inter-cluster galaxies, exhibit ordered substructure within the bulk overdensity. The potential inter-cluster structures, particular those in the Horologium-Reticulum supercluster, are the focus of our astronomer colleagues. Knowledge of supercluster substructure may help to confirm or refute current models of the universe's structural evolution.

2 Data Collection Methods

The data used for this visualization project was collected by the UK Schmidt Telescope at the Anglo Australian Observatory. Attached to the telescope is an instrument, called the 6dF (6-degree field), which is used to perform spectroscopic analysis of the light captured by the telescope from individual galaxies. The galaxies of interest are those that make up the Horologium-Reticulum supercluster.

Spectroscopic analysis is used to determine the redshift of light emitted from each of the galaxies under study. The redshift provides an estimate of the recessional velocity of each galaxy. Since the global trend in the cosmological motion is expansion proportional to the distance of objects from each other, the recessional velocity provides an estimate of each galaxy's distance from Earth.

Spectroscopic analysis is accomplished by projecting the light from the telescope onto the 6dF field plate. The location in the sky of each galaxy under study is known a priori, and a prism is placed upon the 6dF field plate at the locations where each galaxy's light is received. Optic fiber connects the prism to the spectrometer. For speed and accuracy, a robotic arm is employed to place the prisms on the 6dF field plate. This allows the redshift of light from up to 150 galaxies to be observed for a single plate setup, and for many plate setups to be used in a single night [6].

We hoped to use simulation data in this visualization project, but time limitations prevented that. Further details of the simulation data is provided in the section "Data Description."

3 Visualization Goals

3.1 Questions

Our astronomer colleagues have several basic questions they wish to answer with this visualization project. In order of importance relative to their research goals, the questions are:

1. Within Horologium-Reticulum, how does one characterize inter-cluster structure? Are there voids, are there spherical lumps, is the structure sponge-like with highly-connected overdensities?

2. How do observed structures compare with structures in simulated data from universe formation models? Are the same types of shapes found in the observed data also found in the simulated data? Are the relative distributions of structures the same in observed and simulated data?

3. Within the simulation data, how do true galaxy positions differ from those that would be calculated by an observer? Do the true and observed positions of galaxies contained in some structure vary more than those not contained in a structure? How does density affect observed galaxy position?

The first question has received little attention from the research community. By answering it, the astronomers hope to understand how the global overdensity affects member galaxy positions. The second question has also received surprisingly little attention due to lack of interaction among astronomers working with observational data and astronomers working on simulations. Answering it would have strong implications on the validity of current universe formation models.

3.2 Other Goals

Since determining the member galaxies of inter-cluster structures may be difficult to automate, and there is not currently a good definition of how to define inter-cluster structure, our colleagues requested the ability to manually create groups of galaxies. This would involve selecting individual galaxies and adding or removing them from groups. They also require the ability to export selected galaxies to a file format accepted by a particular plotting package with which they can make plots for inclusion in publications.

4 Data Description

There are two main types of data used in this project, observation data and simulation data. The observation data comes in two forms. The first form consists of 3D positional data of galaxies derived from the sky location and redshift data. Out of a total of 2403 total observed galaxies, 1708 galaxies whose recessional velocities were between 12,000 and 27,000 km/sec were chosen. This particular range of recessional velocities are those of the galaxies in and around Horologium-Reticulum. The second form of data consist of the 3D locations of identified cluster centers, derived in the same way as the individual galaxy position data, but averaged over the locations of cluster members. Data for 30 clusters are used in this project.

The clusters have been defined in a catalogue created by George Abell in 1958 and upated in 1989 [2]. The clustering criteria set forth in [2] considered only the two dimensions of the data that define where a galaxy appears in the sky. By taking into account the third dimension provided by redshift data, our colleagues have find-tuned the members of the clusters by rejecting galaxies whose recessional velocities vary by more than 2,000 km/sec from the average cluster membership velocities. The exact derivation of recessional velocity from redshift data is described later in this paper.

The galaxies in identified clusters are not of interest to the study at hand. In fact, their presence can actually interfere with determining inter-cluster structure. The interference is a result of each cluster galaxy's *peculiar* motion with respect to other members of the cluster, that is, motion caused by the local perturbations of the cluster's gravitational potential. This peculiar motion causes variations in redshift, making cluster galaxies appear to be either closer or farther than they truly are. Hence, galaxies within a cluster may have a recessional velocity not proportional to their distance from Earth. This results in those galaxies appearing to intermingle with the true inter-cluster galaxies. For this reason, the galaxies within the clusters are removed from the first data set. However, the known cluster locations can serve as positional signposts in the remaining data, so it is desirable to have this information available.

The simulation data that we were going to look at consists of 3D positions and velocities of dark matter particles, each of which represent a mass slightly smaller than a typical galactic mass. In addition to the position data, the data set has vector data representing particle velocities. The simulation data has 256^3 particles total [4].

As structure identification proceeds on both observation and simulation data, nominal data describing structure membership is created. This data may come from an automatic method or from human selection. In either case, this data is essential for characterizing inter-cluster structure.

In addition, the galaxy position data is averaged to produce a rectilinear grid with a ratio scalar density field. Density can be useful as another means to study structure in the data.

All told, we had available three non-field data sets (observed galaxy position, observed cluster position, and simulated galaxy position), two nominal data sets (observed and simulated), and two scalar fields (observed density and simulated density).

4.1 Observation Data

The observational data describes the locations of galaxies in a particular region of the sky in the southern hemisphere. The data files are formatted as 2D arrays of spatial values describing the angular positions of galaxies relative to Earth's orientation and position in space, plus a recessional velocity component, derived from redshift, which can be taken roughly as a third dimension. One of the two angular positions describes right ascension (RA) and the other describes declination (DEC). Each galaxy has associated with it a scalar value representing its recessional speed (cz). The data is stored in ASCII files consisting of floating-point data with between 2 and 6 decimal digits of precision.

RA describes the position of an object in the sky along the celestial equator, a conceptual plane that sweeps out into space from Earth's equator. It is akin to longitude in surface navigation. RA is measured in terms of hours, minutes, and seconds. With the exception of the two rotational poles, a given point on Earth sweeps through all RA positions in a 24-hour span [8].

DEC describes the position above or below the celestial equator. It is measured in terms of degrees, arcminutes, and arcseconds. The equatorial position corresponds to 0^o while the north pole is at $+90^o$ and the south pole is at -90^o. There are 60 arcminutes in a degree and 60 arcseconds in an arcminute [8].

The cz value consists of two components: c, the velocity of light, and z, the ratio between the observed wavelength of a galaxy and the emitted wavelength minus 1. Thus, $z = \frac{\lambda(observed)}{\lambda(emitted)} - 1$ [10].

4.2 Observational Data Issues

The set of galaxies chosen for redshift measurement represent approximately 70% of the known galaxies above a given apparent brightness ($b_J < 17.5$) in the observed field of view. To perform the sampling, a grid was overlayed on top of an equal-area plot of known galaxies. 70% of the samples within each grid box were chosen at random to determine the end random distribution. Sampling in this way reflected the distribution of galaxies with respect to RA and DEC.

As described above, galaxies that are members of known clusters are removed from the main galaxy data. They are not completely missing, however, as the positions of the clusters are kept.

The 6dF instrument provides a precise method of measuring galactic redshift data. Each fiber has a 6.7 arcsecond diameter and can be positioned to an accuracy of 1 arcsecond. Redshift data is accurate to within 50km/s [3]. Unfortunately, there is a great deal of uncertainty in deriving distance from redshift data. First, conversion from redshift data depends on the choice of Hubble constant. The Hubble constant is currently unknown and estimates of it vary. Our astronomer colleagues have chosen to set the Hubble constant at 70 km s^{-1}/Mpc, where Mpc stands for megaparsec (one million parsecs). Second, gravity affects the redshift of light as it travels through the universe in non-obvious ways, causing uncertainty in the redshift values. No attempt to correct for these uncertainties is undertaken in this project.

Redshift is also affected by the peculiar motion of galaxies, that is, motion within the supercluster. A galaxy's peculiar motion affects its observed redshift and hence the estimated position of the galaxy. For example, a peculiar motion of a galaxy away from Earth combined with the general expansion of the universe would make that galaxy appear farther away while a peculiar motion toward Earth would make the galaxy appear closer than its true distance. For the inter-cluster galaxies under study, the error induced by peculiar motion was assumed to be negligible, within 1% of the motion caused by the general expansion of the universe.

Finally, the quantization level of the data has no effect on its interpretation. Each data value has 6 or 7 significant digits, about as good as one could expect from observational data.

4.3 Simulation Data

The simulation data comes from computations carried out by the Virgo Supercomputing Consortium using computers at the Computing Centre of the Max-Planck Society in Garching, Germany. The data were generated by a low density cold dark matter model of universe formation. The lambda CDM model used 256^3 cold particles with mass on the order of a small galaxy, that is, 4.8×10^{10} solar masses [4]. The Hubble constant was assumed to be 70 km s^{-1}/Mpc, the same value as our astronomer colleagues are assuming. The simulation code HYDRA was run on a Cray T3D using 64 processors [7]. As in the observational data, the simulation data contains 3D positions of these cold particles. In addition, the velocities of each galaxy are available.

4.4 Simulation Data Issues

In the simulation data, all positions and velocities of the particles are known precisely. This is in contrast to the observational data, where only a galaxy's apparent position is known. This poses a challenge when trying to compare the simulated data with the observed data. To make a useful comparison, a reference observation point would need to be chosen in the simulation domain, and the simulation data would need to be converted to RA-DEC-cz values as seen from the observation point. Then, the RA-DEC-cz values would need to be converted back to the same 3D space to which the observation data were projected.

The number of particles, 256^3, is many orders of magnitude higher than the number of galaxies in the observed data. Therefore, data reduction would be required to make meaningful comparisons between the two. Our colleagues have typically used only one eighth of the simulation data, reducing the data to 128^3 particles. They further sample the particles down to 1/1000 the number of particles in the original eight of the data set, giving approximately the same number of samples as in the observational data.

In addition, the clusters are not removed from the simulation data, making useful comparisons with the observed data difficult. In the observed data, clusters would need to be determined by an automatic clustering selection algorithm such as the C4 algorithm in [5]. We would need to convert the data to RA-DEC-cz values and run this clustering algorithm on the simulated data, removing the clusters that it found from the data set.

5 Visualization Design

In this section, we detail our ideal visualization strategy for this data and, more importantly, for achieving our colleagues' research goals. We did not consider constraints, either time-wise or resource-wise, on implementation of the techniques we chose, but rather focused on what would be the best for reaching the research goals.

5.1 General Visualization Techniques

A common thread through the various questions we attempted to answer was the need to accurately perceive 3D shape. According to Ware, stereo depth and structure-from-motion are very important for 3D structure perception [9]. Taking this into account our system should provide support for stereoscopic displays. Users should be able to adjust eye-separation parameters to fit their particular preferences. In addition we should provide interactive view control. This would allow rotation, zooming, and panning. Users who have control over the view have a better understanding of 3D shape. Sometimes, however, we thought it might have been beneficial to automate a torsional rocking motion for showing structure-from-motion. This would allow the user to concentrate on the structure information he or she was seeing instead of on driving the interaction.

Another option useful for answering each of the questions would be the ability to limit the visualization volume. Only the galaxies located in the specified visualization volume would be viewable, allowing for the user's concentration on smaller scale structures. This would remove galaxies from the background that might distract the viewer from the local structure. The boundaries of the viewing volume should also be interactively adjustable.

5.2 Identifying and Characterizing Structure in Observed Data

There are several visualization techniques that could be applied to the problem of identifying and characterizing inter-cluster structure. We would chose to use isosurfaces with screen door transparency for exploring the density field, glyphs for representing galaxies and clusters, and to restrict the view volume. After averaging the point data to a structured grid, isosurfaces could be generated to visualize regions of a particular density. It is assumed that regions containing structures were of higher density than surrounding regions, and would show up when visualizing with isosurfaces. Since there is no clear density definition for structure, we allowed the isovalue to be adjusted by the user.

The ability to view and select individual galaxies is important to our colleagues. In order to provide the ability to view and select individual galaxies within an isosurface, a texture that provides screen door transparency should be mapped on the isosurface. The overall structure is expected to be of low frequency, so a screen door texture with large enough holes to allow viewing of the galaxies would not interfere with perception of the isosurface's shape. This combination would allow for observing both the overall structure of the density field and for viewing the individual galaxies within the isosurface. Furthermore the texture on the surface would aid in stereoscopic fusion. The alternative would be to use an opaque or translucent isosurface. An opaque isosurface would have occluded viewing of the galaxies contained within. A translucent isosurface would have interfered with shape and depth perception of the isosurface. In addition, a translucent isosurface would have made it difficult to discern if galaxies were contained within a particular connected component of the isosurface. If a galaxy glyph's colors had been blended with the isosurface, one could not know if the galaxy was behind only one isosurface boundary, making it inside the volume enclosed by the isosurface, or behind two isosurface boundaries, making it outside the isosurface.

Glyphs should be used to indicate specific objects in our data set, which include galaxies and previously-specified clusters. Spherical and cylindrical glyphs should be used to represent galaxies and clusters, respectively, and each should have a perceptually distinct color. Glyphs have two advantages over pixel-sized points in our visualization. First, glyphs have size and take up physical space which allows for scaling due to perspective projection. Second, glyphs occlude each other, helping with depth perception. If galaxies are represented as pixel points in space, both of these depth cues would be lost. To help with the characterization task, the user

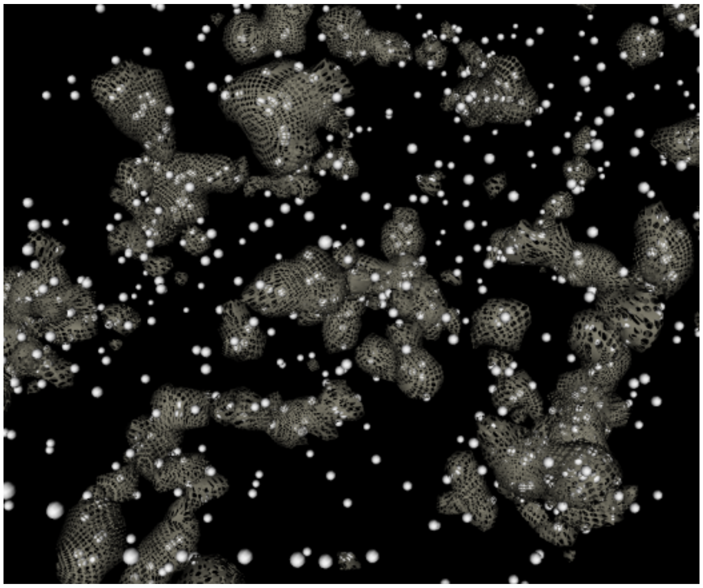


Figure 1: An example of screen door transparency applied to the density isosurface.

could select galaxies and assign them to different groups. These groups should be nominally color-coded. The color mapping scheme should be selected by the program to ensure a suitable selection of colors so that each group can easily be distinguished. The choice of colors should be based on the recommendation in Ware's book [9].

Another technique that was considered to answer this question was direct volume rendering. While this technique would give an overall impression of density throughout the volume, it would be difficult to perceive 3D structures, such as filaments, in the data set. If we chose to map opacity to density with, say, a linear ramp, there would be no sharp surface generated, so shape perception of structures in the data set would not be easily perceived. If, on the other hand, we chose to map opacity to density gradient, then edges of structures would be easier to see, but at the cost of some translucency in the volume. A final strike against direct volume rendering is the fact that the point data are not of sufficient density to justify the cloud-like effect direct volume rendering produces. In the end, an isosurface with screen door transparency will produce a readily perceivable surface to distinguish shapes in the data set while still allowing users to view galaxies within the isosurface.

5.3 Comparing Observed Data to Simulated Data

Viewing a single 3D data set is challenging. Comparing two 3D data sets in a single visualization is even more challenging, depending on the similarity of the two data sets. Comparing two data sets representing the same natural phenomena but which come from different sources, all within the same visualization is, to our knowledge, an untouched region of visualization research. Since we do not have a quantitative technique to measure the difference between observed and simulated data, we could not, for instance, simply visualize the difference magnitude between the two data sets. Instead, we can use a qualitative approach that involved viewing the two data sets side-by-side using the same techniques for identifying and characterizing structure. The camera orientation on each data set should be separately changeable, allowing exploration of each data set individually. Once the viewer finds a structure of interest in one data set, the viewer could then explore the other data set in search of a similar structure. Parameter settings for the isosurface, however, would be locked together. This would allow a qualitative observation of both data sets simultaneously.

5.4 Comparing True and Observation-derived Galaxy Positions in Simulated Data

As discussed in the data set description, galaxy positions derived from observational data have an associated error caused by the peculiar motions of galaxies as they move relative to each other. This error can be characterized by the distance between the true positions and those derived from a what a reference observer would see. To see this error, we would use motion of glyphs between the true and observation-derived galaxy positions. Greater motion would indicate greater error. The motion could either be automated or controlled by a slider widget in the user interface, allowing the user to view either the true or observed positions, or somewhere in between. This would allow the user to identify a structure in the observed positions and then slide the galaxies to their true positions to see if the structure existed in the true galaxy positions.

For automatic motion, we would vary the parameter of the interpolating function (which ranges over [0,1]) for each galaxy using a sinusoid. The zero-derivative at either endpoint of the range would give the galaxy motion a natural appearance. Additionally, we would calculate a density field from the true position data. The galaxy glyphs would be pseudo-colored according to the density at their true position with a black body radiation color map. This color map would be the best for comparing the density levels at various galaxy locations. It would also make identifying correlation between density and error magnitudes easier. If objects of the same color move by roughly the same amount, then that would imply a strong correlation between density and error. We would use a black background, so we would only use the portion of the black body radiation color map starting at pure red. We also would display a legend of the color map to allow estimation of density values.

The other techniques we considered for displaying the error in observation-derived galaxy positions could not achieve what motion and the black body color map could. We could have computed the error at each simulation point, averaged the error values onto a rectilinear grid, then used volume rendering of this scalar value to locate high error regions. This method, however, would not have admitted an easy way to correlate error to density. It also would not have allowed comparison of observation-derived positions in comparison to the true positions. We could also have created isosurfaces, one for error and one for density, but one isosurface could occlude another, making correlation between error and density difficult to discern. Moreover, isosurfaces would only display how *one* error value correlated with *one* density value. Finally, cutting planes would not be useful because of occlusion. Our choice of using two distinct visual channels that do not interfere with each other would allow the relationship between error and density to be conveyed well.

5.5 Keeping Viewers Oriented

One of the complaints our colleagues had about previous 3D visualizations of their galaxy position data was that it was easy to lose a sense of orientation to the data. We would alleviate that problem in each of our visualizations by displaying reference axes of the RA-DEC-cz coordinate system. The axes would appear as the outline of a truncated pyramid with a rounded base. Along each side of the pyramid would be a grid of reference coordinates. Every n-th line in the grid would be labeled with its coordinate value. The user should have the option of selecting which sides of the truncated pyramid are visible. The elements that comprise the axes are colored a just-noticeably-different color than the background so as to not distract the viewer from the primary information.

In addition to simply providing the grid as a reference, a user-interface option should be available to draw curves from selected galaxies to one of the sides of the axes pyramid. This direct connection between galaxies and their coordinates would provide a useful way to orient the viewer to the data, and allow for reasonably precise location of structures in the data. Other strategies, such as drop-shadows, would not have worked for two reasons. First, there were too many points, making identification of a galaxy's drop-shadow difficult at best. Second, the RA-DEC-cz coordinate system is essentially a spherical coordinate system, meaning a viewer would need to follow the shadow though a curved space, something that is not intuitive or, even if the user understands the coordinate space, easy to do. Drawing curves between selected galaxies and the reference axes would avoid these problems.

6 Design Implementation

We designed our software to address three separate goals, and each of these goals required different techniques to be applied. Because of the difference in techniques for each goal, we decided to build a separate application to address each of the three visualization goals. In order to optimize code reuse, each of the applications is set up to use the same core set of modules, while each has its own graphical user interface (GUI) specific to the visualization goal. The primary goal of identifying and locating inter-cluster structure by itself contained several subgoals, which required the use of

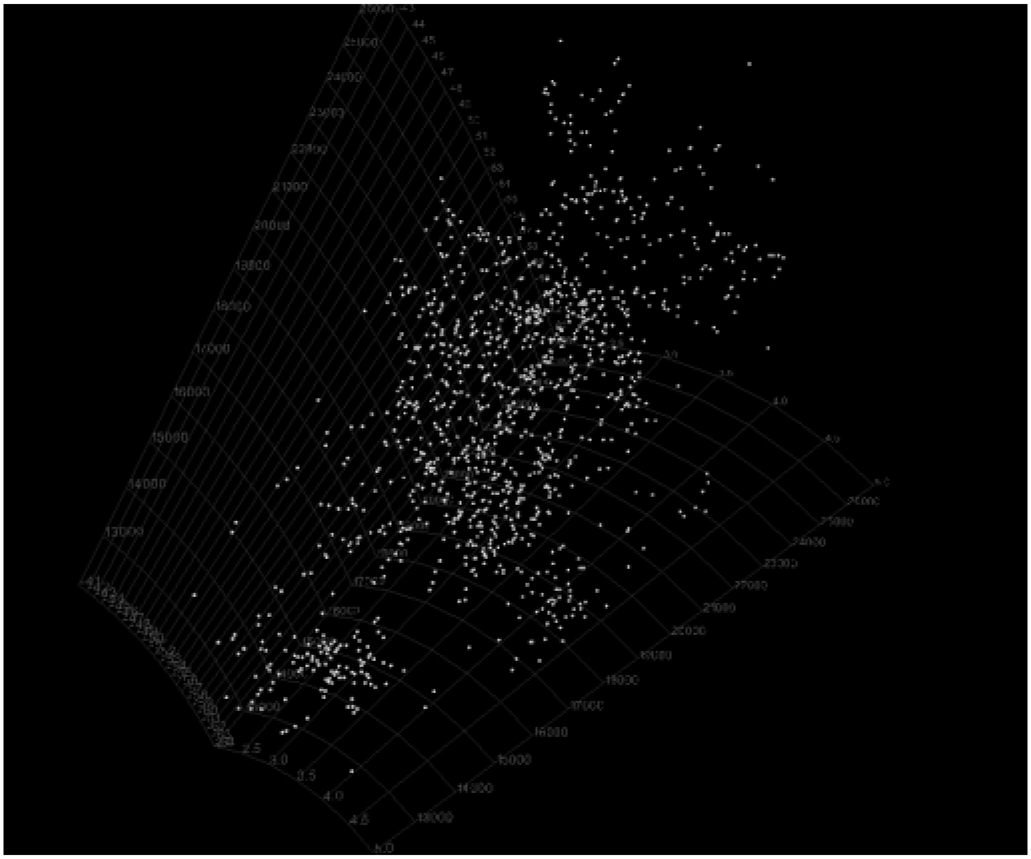


Figure 2: An example of labeled axes.

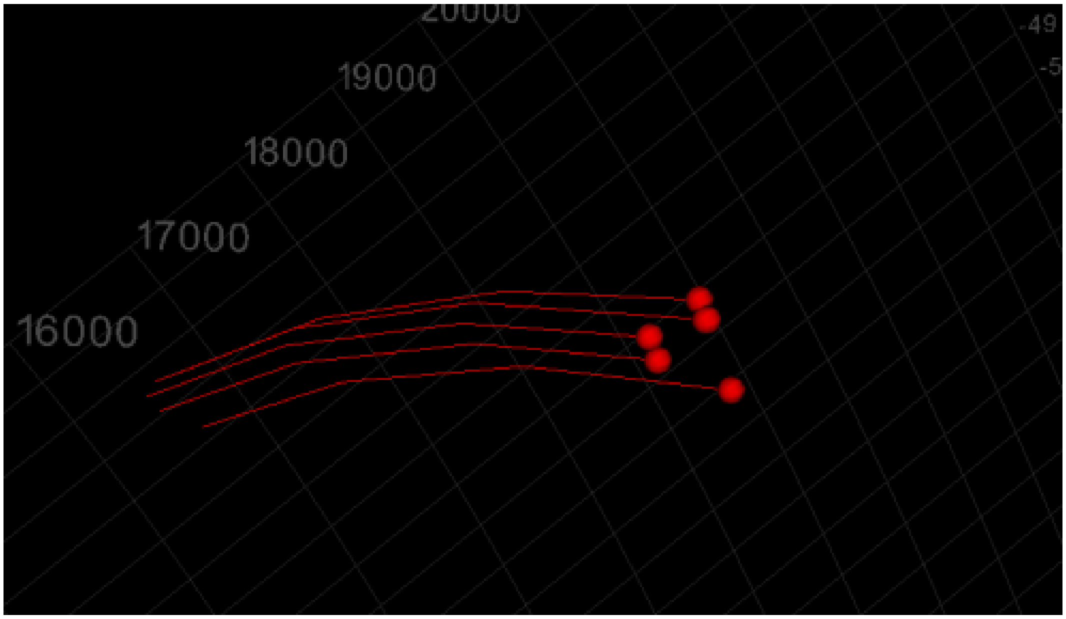
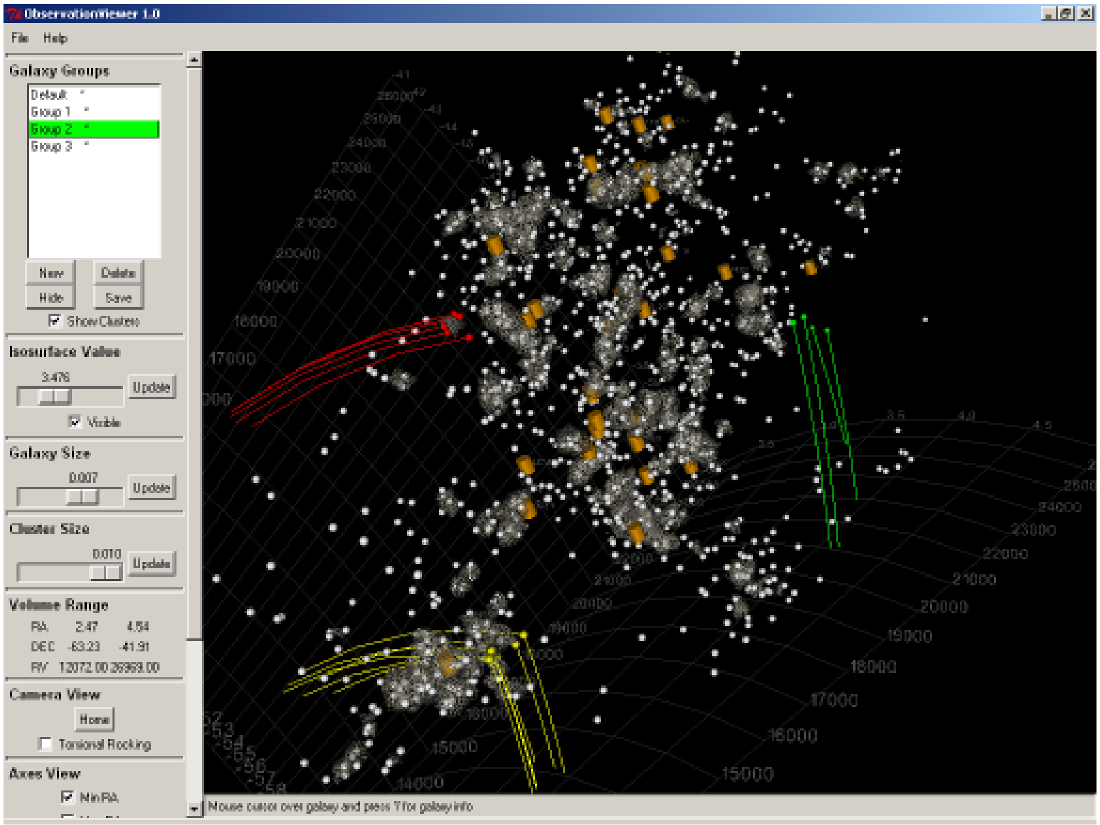


Figure 3: An example of lines connecting galaxies in space to a reference grid.

several visualization techniques. We worked closely with our clients in order to gain feedback on each version of our design, and incorporated their suggestions into successive implementations of the design. The end result is an implementation of all the techniques that we thought would provide the optimal visualization for the primary goal. The system provides an interactive interface in which to explore their data set. The secondary and tertiary goals were not completed, as fully implementing the techniques that we thought would be optimal for the first visualization took longer than expected. The issues of implementing these secondary and tertiary goals will be discussed later in the paper.

Our project did not involve extending a visualization system that was already in place, but rather was designed from the ground up. We first attempted to use Java to develop the software, but quickly changed to Python. We used Python because it seemed have better integration with VTK, especially under Linux and Mac OS X. Tkinter, a Python wrapper for TCL/TK, was used to generate the user interface. We chose to use VTK as the visualization library. Our choice of VTK and Python means that our system should run on any hardware that supports these two packages. Our current software runs against VTK 4.4, and works with Python 2.2 or 2.3. The current binary release of VTK (4.2) requires the use of Python 2.1, which is incompatible with our Python code. The system was developed and tested on both a Windows and Linux platform, and was also run under Mac OS X. Some portions of this program were only tested under the Windows environment, such as stereo using the CrystalEyes emitter and goggles. Our clients evaluated



ObservationViewer 1.0

Galaxy Groups

- Default *
- Group 1 *
- Group 2 *
- Group 3 *

☑ Show Clusters

Isosurface Value: 3.476

☑ Visible

Galaxy Size: 0.007

Cluster Size: 0.010

Volume Range

RA	2.47	4.54
DEC	-63.23	-41.91
RV	12072.00	26969.00

Camera View

☐ Torsional Rocking

Axes View

☑ Min RA

Mouse cursor over galaxy and press 'i' for galaxy info

Figure 4: Screen shot of the static visualization techniques present in the system.

the software on a Dell Precision system with a 2.0 GHz Pentium processor, 1 GB RAM, and an nVIDIA Quadro4 900 XGL graphics card. The monitor was a 21” Dell CRT capable of stereo update rates. Attached to the graphics card was an infrared emitter that synchronized the shutters of the CrystalEyes stereo goggles made by the StereoGraphics corporation.

6.1 Code Structure

The structure of the code is divided into several parts. The top level structures include the model, the view, and the GUI. The GUI is specific to each of the individual applications, while the model and view should be more or less portable across all three of our applications. We attempted to loosely follow the model/view/controller design pattern by dividing our code into three main categories. The ”model” part of our system contained data structures used for storing the astronomical data, including galaxy and cluster position data, along with transforms between the different coordinate systems being used (e.g. RA-DEC-cz and x-y-z). The ”view” part of our system is responsible for handling the visualization of the data. This component is responsible for changing the visibility of objects in the visualization. The GUI is responsible for the creation of the user interface, along with communicating user inputs to the view/model.

6.2 Visualization Algorithms

The data files that were read by this application are sets of point data for galaxy and cluster position. The data can be viewed as discrete objects, where each galaxy and cluster is represented by individual glyphs. An isosurface can also be generated using a particular isovalue in a density field that is computed from the galaxy position data. Each of the objects that can be visualized is created when the data set is loaded, but is not always visible by default. For instance, all of the lines from galaxy glyphs to a particular axis are created when the galaxy data set is loaded, but the visibility of these actors is turned off. When the user wants to view the lines, the visibility is turned on.

The first step in importing the data is to scale, rotate, and translate the galaxy and cluster positions to fit within a 2x2x2 box centered at the origin of the viewing volume. The positions are rotated so that the long dimension of the data set, the cz dimension, is aligned with the z-axis in x-y-z space. This is important when calculating the density field because it allows the data volume to more fully fill the viewing domain, improving the grid-size-to-data-frequency ratio.

6.2.1 Camera Controls

A home button is included to set the camera at a default location in case the user gets lost in the visualization and would like to get back to a familiar viewpoint. Another camera control is provided to turn on and off torsional rocking of the camera about the up vector. This rocking is accomplished by setting a timer in Python to rotate the image a certain angle every 1/30 of a second. The angle of rotation is determined by a sinusoidal function whose argument is incremented ($\pi/16$) for each camera rotation.

6.2.2 Galaxy Glyphs

The galaxies are represented by spherical glyphs. Galaxies are colored according to which group they have been assigned to by the user. Groups of galaxy glyphs are contained and controlled by the GalaxyGroupManager class. The GalaxyGroupManager class also controls the assignment of colors to galaxy groups. A vtkSphereSource is used to create the sphere glyphs. The coordinates of each galaxy in x-y-z space is passed to the vtkSphereSource. A vtkGlyph3D object is then created using the output of the vtkSphereSource object. A vtkPolyDataMapper takes the output from vtkGlyph3D object. A vtkActor is assigned to this vtkGlyph3D object, which is then assigned to the vtkRenderer. These glyphs are color coded according to which group they are assigned. The visibility of these glyphs can be toggled on and off, the size of the glyphs can be set by the user through a slider bar, and lines can be drawn from the glyph to its location on the axes.

In a second iteration of our implementation, we chose to assign the galaxy glyphs to their own actors. Our first implementation grouped all of the galaxy glyphs into a single vtkActor. This caused problems when we started to implement the ability to pick out individual galaxies and assign them to different groups. VTK includes functions to query the visualization to see what actors intersect with a line drawn from the mouse pointer through the view volume. Since we only had a single actor representing all of the galaxies, this did not tell us what galaxy we were trying to pick. Representing all of the galaxies as individual actors was the correct way to do this, though it adds some memory overhead.

6.2.3 GalaxyGroupManager

The GalaxyGroupManager is used to organize the galaxy groups, and is the main point for accessing the galaxies and galaxy glyphs. The Galaxy Group Manager is responsible for automatically assigning colors to each of the galaxy groups. The list of available colors is based on Ware's color selection for nominal coding [**?**, ware]

6.2.4 GalaxyRenderWidget

This class is a TK widget that provides a rendering window. It subclasses vtkTkRenderWidget, but adds extra key-bindings. The keys 'm','n','b','v' are bound to functions that handle the visibility of curves from glyphs to axes. The keys 'p' and 'i' are bound to functions that pick individual galaxies. Individual galaxies are picked through the use of vtkPointPicker. The 'p' key assigns the selected galaxy to the currently selected group while the 'i' key displays information about the galaxy in the GUI.

6.2.5 Cluster Glyphs

The locations of previously identified clusters are indicated through the use of cylindrical glyphs. These glyphs are colored orange, which is perceptually distinct from any other object in the visualization. The size and visibility of these glyphs can be easily adjusted by the user through a checkbox and slider bar, respectively. The pipeline for creating the cluster glyphs is similar to the pipeline for the galaxy glyphs, except that a vtkCylinderSource is used instead of a vtkSphereSource. The ID of each glyph is shown next to the cluster in the visualization. The text of the ID is set up to always face the camera through the use of a vtkFollower instead of a vtkActor.

6.2.6 Drop Curves

Another visualization technique we implemented is to have curved lines drawn from galaxy glyphs in a particular group to a particular axis. The lines are colored the same as the galaxy glyphs to which they are connected. They are curved to follow a galaxy's RA or DEC value as it varies in the other dimension to one of the reference axes. The commands for drawing lines from galaxies to an axis are bound to keyboard buttons. Keyboard buttons 'm', 'n', 'b', 'v' are bound to turning on or off curves from galaxies to the RA-min, RA-max, DEC-min, DEC-max axis.

One problem with using multiple lines to approximate the curves from the galaxies to the axes is the amount of memory it takes to store all the line segments. We attempted three different implementations. The first attempt was to draw a straight line from the galaxy to the axis. This required little memory but did not look very good, and could be confusing as lines could criss-cross. The second attempt used multiple vtkLineSources for each curve. The approach looked better and was less confusing but also took about 250 MB of memory! To see if the amount of memory used could be reduced, a third approach was tried using vtkPolyLine objects. The problem with this approach is that we also had to create vtkUnstructuredGrid objects to store each vtkPolyLine. These vtkUnstructuredGrid objects had to be mapped with the vtkDataSetMapper and then assigned to a vtkActor. This resulted in memory usage for curved lines alone ballooning to 400 MB! We decided to use the vtkLineSource approach as this provided curved lines at lower memory usage. The best way to implement this would have been to generate the curved lines on the fly as they are requested, but this might also slow down the interactivity of turning the lines on and off.

6.2.7 Density Field Generation

To generate the galaxy density field efficiently, we first create a vtkRectilinearGrid object, set its dimensions to 128^3, and assign evenly-spaced slice coordinate positions along each dimension. We then create a vtkPointLocator object and connect it to the vtkRectilinearGrid object. This sets up an octree structure to make locating the grid nodes within a certain radius of each galaxy position efficient. We set this radius to be the length of four grid spacings. For each galaxy, we find each grid node within this radius, calculate the distance from the galaxy to the grid node, and use this distance as a parameter to a 3D Gaussian function. The closer the galaxy is to a grid point, the higher its density contribution to that grid point.

6.2.8 Isosurface generation

To generate the isosurface, we use a vtkContourFilter applied to the density field. We then use the vtkPolyDataNormals filter to generate smooth normals at each vertex in the isosurface. To generate the screen door transparency effect, we use a texture map with circular regions of complete transparency. The 256x256 texture map is shown here. The white region is fully opaque while the black circles represent fully transparent areas. Using a white texture map allows the color of the isosurface to be modulated with a color of our choosing, a dull gray color that is perceptually distinct

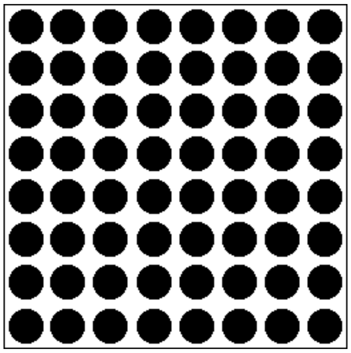


Figure 5: The texture pattern used for creating the screen door transparency effect on the isosurface. Black represents completely transparent regions while white shows the opaque regions.

from all the other objects in the visualization. The texture map is in the form of a VTK structured points data set and is read in with a vtkStructuredPointsReader object.

After the isosurface is created, we identify all its connected components using a vtkPolyDataConnectivityFilter. Then, for each connected component, we create a new vtkPolyData object and perform a deep copy from the connected component geometry to the new vtkPolyData object. This copies vertices, connectivity, and normals from the original isosurface to the vtkPolyData object.

Once the connected components are copied to individual vtkPolyData objects, we generate texture coordinates separately for each connected component. We use a vtkTextureMapToSphere object centered at the center of the bounding box of the connected component. This determines the texture coordinates of the connected component vertices by determining where each vertex projects onto the surface of a sphere whose texture coordinates are computed analytically. We also scale the texture coordinates relative to the size of the connected component to make the holes in the various connected components roughly the same size.

Our first attempt at screen door transparency used a vtkTextureMapToPlane object to generate texture coordinates. This was a quick method that worked reasonably well, but it suffered from the problem that the sides of the isosurface with tangents perpendicular to the texture map plane would either be completely solid or completely “chopped off.” Using the vtkTextureMapToSphere object eliminated these large holes or large unbroken pieces of isosurface.

One major downside of our approach to assigning texture coordinates is that it is an expensive process. Each update of the isovalue requires updating the isosurface, locating all the connected components, and performing expensive memory allocation and copying. This slowness prevents interactive manipulation of the isovalue. Another problem with this approach is evident when a connected component is long and narrow. The holes near the center are small while the holes on the far ends of the connected component are large. Using a more advanced algorithm for texture coordinate assignment would fix this problem.

6.2.9 Axes

Four different axes are available in the visualization. The four axes are positioned at the limits of the volume containing the galaxies. Two of the axes are oriented in the RA-cz coordinate plane, sweeping through the range of DEC values, and the other two are oriented in the DEC-cz coordinate plane, sweeping through the RA values. For each of the axes, a grid line is drawn every 1000 cz units. The axes are labeled with RA and cz or DEC and cz values, depending on the reference axes. The labels are set to always face the camera, so that they can be read no matter the camera orientation.

7 Visualization System Evaluation

As we were only able to complete the visualization for answering our colleagues' primary goal of finding structure in their observed data, we limited our evaluation to this system. Before performing our evaluation, we provided a brief overview of the various parameter controls our system allowed users to set. Since the goal of this evaluation was to gauge the effectiveness of our visualization technique and not our user interface design, we felt this introduction to the basic parameter controls of our application was appropriate.

As student and advisor, our colleagues Matt and Jim work closely on analyzing their data. Because of this, we felt they should work together on our evaluation tasks. During the evaluation, they each took turns as the primary manipulator of the system. The person not in control would often make suggestions as to the course of action to take. This was helpful in analyzing our system's effectiveness because each of their thought processes was communicated verbally to each other.

For evaluation of our system, we first asked our colleagues to perform three tasks with synthetic data sets. The first task involved identifying shapes and their locations in a data set. The second task involved identifying voids, or locations where no galaxies were present, and their locations. Finally, the third task involved counting the number of galaxies within a connected component of an isosurface. After our colleagues finished these tasks based on synthetic data, we observed them as they viewed their data for the first time with our fully-implemented visualization system.

7.1 Identifying and Locating Structures

For the first task, we inserted familiar shapes, such as spherical shells, cube shells, and cylindrical shells, into a data set consisting of galaxy positions uniformly distributed

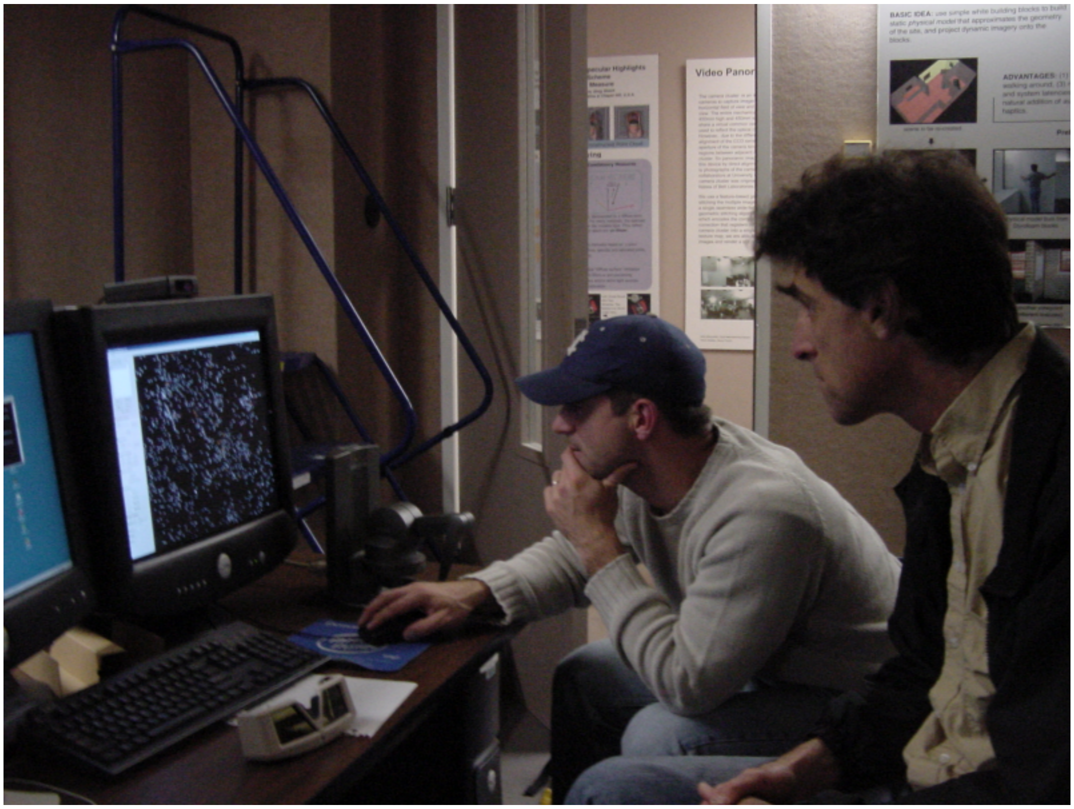


Figure 6: Matt and Jim viewing their data with our visualization system. Their communication during the evaluation task helped us better understand how they worked with the system.

in the RA, DEC, and cz dimensions. 1500 random galaxy locations were generated inside a volume ranging from 2.0 to 4.5 hours in the RA dimension, -5 to 5 degrees in the DEC dimension, and 12000 to 27000 km/sec in the cz dimension. We then asked our colleagues to identify the shapes we inserted into the random data set. Once the shapes were identified, we asked our colleagues to locate the centers of the shapes in the RA-DEC-cz coordinate system. We had them repeat this task, but with stereo goggles on the second time. This task was designed to see how effective our visualization system was for identifying and locating structure.

For the first task without the stereo goggles, we inserted a spherical shell, a cube shell, and a cylindrical shell. Our colleagues were able to identify the inserted shapes right away. They were a bit surprised to find such common shapes since they were coming in with the mindset that they would be looking for cosmological features. After this initial confusion subsided, they set about identifying the centers of the shapes. Right away, they turned some of the reference axes on. After that, they opted to draw curves from the galaxy positions to the reference axes. Since they only wanted a few curves drawn, they chose to create a galaxy group, pick some of the galaxies on the shape they were looking at, and draw curves from those galaxies to the reference axes. There was some confusion when they tried to determine which points would be the best for finding the center of the shapes. They thought that the shapes were filled, but after a few minutes, realized that the shapes were just shells. After a while, they decided to add some representative galaxies distributed around the shape to get an estimate of the center of the shape. Finding the center of the first shape they identified took roughly 15 minutes. This was, no doubt, because of unfamiliarity with the application and the task, in addition to noise in the lab in which we did the evaluation. In fact, Jim made a discouraging comment after that first attempt: “This should be easier.” Our colleagues were also very careful in their estimation, taking the time to be as precise as possible in their answers, more precise than we were expecting them to be. They even joked about providing a plus-or-minus error term in their answers. Identification of the second shape’s center went much faster. There were some hiccups due to a small bug in the program and duplicated galaxy position data in the test files, but our colleagues were able to find workarounds without us intervening and proceeded with the task.

The results for the first task were encouraging. They found the shapes in the random noise immediately, and their estimates of the shape positions were reasonably accurate. Table 1 shows a list of the shapes, their actual centers, and the estimated centers.

For the repetition of this task with stereo, we had another data set with three other shapes added in. This time, there were two cube shells at different locations and a single warped cylinder. Again, they found the shapes right away. They were a little confused because they thought this new data set was the same as the old one, and so they second-guessed their identification of a sphere in the first data set. They eventually figured out that this was a different data set. Our colleagues’ initial

Shape	RA	Est. RA	DEC	Est. DEC	cz	Est. cz
Cube shell	3.8	3.75	0	-0.2	18000	17800
Sphere shell	2.8	2.8	0	-0.2	14000	14100
Cylinder shell	4.02	4.0	0	0	22590	22500

Table 1: Comparison of true and estimated shape center positions without stereo

impressions of stereo were mixed. Jim said of the stereo when he first saw it, “That is pretty cool. That is pretty amazing.” When Matt put on the stereo goggles, he had trouble perceiving the stereo, especially when the camera was zoomed in. This difficulty arose from the camera model applying a scaling to the visualization actors to zoom in instead of actually moving forward along the view vector. The scaling of the actors effectively increased the eye separation parameter, causing the stereo effect to drop out at a certain zoom level. The eye separation parameter was adjustable, and both Matt and Jim took advantage of that fact.

One particular difficulty Matt and Jim encountered when using stereo was that the cursor was not displayed in stereo. This made selecting galaxies quite difficult, so they were unable to employ the strategy of selecting a few representative galaxies from the shapes and showing curves from those galaxies to the reference axes. Instead, they adopted a less precise strategy where they just used the reference axes to estimate the location of the shapes. Matt found the axes particularly useful, saying, “You guys did a really good job on the axes. You really outdid the CAVE in Holland... good job!” He was referring to the previous stereo visualization of their data where they had trouble with getting lost because they had no reference points.

The results for this task using stereo were also encouraging. Table 2 shows the actual and estimated coordinates of the shape centers.

Shape	RA	Est. RA	DEC	Est. DEC	cz	Est. cz
Cube shell 1	3.5	3.5	0	0.5	22000	23500
Cube shell 2	2.4	2.45	0	-0.5	17000	15750
Warped cylinder	2.26	2.3	0	0	23293.3	24500

Table 2: Comparison of true and estimated shape center positions with stereo

Overall, stereo did not seem to make the task of identifying shapes or their locations easier or more difficult. However, locating precise galaxy positions was more difficult because of the difficulty in selecting galaxy positions.

7.2 Identifying and Locating Voids

The second task we gave our colleagues was to identify voids in a synthetic data set. This task was designed to assess our visualization technique's ability to reveal structure through displaying "negative" structure, or regions where no galaxies are present. Finding these voids and characterizing their shape has important implications on the universe's topology.

To carry out this evaluation, we first created a random data set using a uniform distribution in each of the RA, DEC, and cz dimensions. The data set was initially seeded with 2500 galaxy locations and it spanned from 2.0 to 4.5 hours in the RA dimension, -5 to 5 degrees in the DEC dimension, and 12000 to 27000 km/sec in the cz dimension. Then, we created the voids by specifying multiple regions defined by minimum and maximum RA, DEC, and cz values, and then removing any galaxies from the random data set that existed in these regions. Two different data sets were used for this task, one for non-stereo and one for stereo. Three voids were created in the first data set. In the second data set, the dimensions of the voids were exactly the same, but their RA dimension was shifted by varying amounts. The other two dimensions were kept the same as this preserved volume from the voids in the previous data set. Table 3 shows the dimensions of the voids.

Void No.	RA span	DEC span	cz span
1	0.5 hrs	6 deg.	2500 km/sec
2	0.5 hrs	4 deg.	2000 km/sec
3	0.1 hrs	3 deg.	4000 km/sec

Table 3: Void dimensions used in void identification and location task

Our colleagues' initial strategy was to use interaction with the data set to find the voids via structure-by-motion. They looked at the data set from a variety of angles, both zoomed out to get a global overview and zoomed in to particular regions for closer examination. One strategy that may have made this task easier would have been to increase the size of the glyphs representing the galaxies. This would have filled the regions of uniform galaxy distribution with more glyph material, thereby revealing the regions without much glyph material as the void regions. Our colleagues did not pursue this strategy, however. They chose to use an isosurface instead. Their first choice of isovalue was low. This setting did not help as it produced few large connected components. They then increased the isovalue, and this was more effective in revealing the voids. Using galaxy grouping and displaying curves from representative galaxies just outside the void to the reference axes, our colleagues were able to get good position estimates of the voids they found. They were, however, only able to identify two of the three voids. This is due to the fact that the third void was too long and skinny to be noticed, especially at a large cz value where the random distribution strategy made the synthetic galaxy positions more sparse.

Void No.	RA	Est. RA	DEC	Est. DEC	cz	Est. cz
1	2.55	2.6	0	2	16250	16500
2	4.05	4	0.5	1	19000	19100
3	3.05	-	0.5	-	21000	-

Table 4: Comparison of true and estimated void center positions without stereo

Void No.	RA	Est. RA	DEC	Est. DEC	cz	Est. cz
1	4.25	4.25	0	0	16250	16800
2	2.35	2.8	0.5	0	19000	20500
3	2.75	-	0.5	-	21000	-

Table 5: Comparison of true and estimated void center positions with stereo

Table 4 shows the true void center positions and the estimated center positions of the galaxies our colleagues found.

Our colleagues repeated the task again on the second data set generated for this task, but this time with stereo goggles on. Stereo did not seem to assist in the task, which surprised Jim, who said, “Stereo is not helping with this at all, but I thought that it would.” Even so, using interaction, structure-by-motion, and the reference axes, our colleagues were able to identify two voids right away and quickly produced an estimate of the void center positions. Table 5 shows the true and estimated center positions.

The estimates were fairly good with stereo, but not as good as without stereo. This was probably because they were unable to group representative galaxies just outside the voids and therefore were unable to use the curves from the galaxies to the reference axes. Their estimate of Void 2 in Table 5 is off by quite a bit in the RA dimension, nearly half an hour (7.5 degrees) near DEC 0, the DEC value where a change in RA has the most significant effect on galaxy position.

With or without stereo, these tasks showed that our visualization system is effective for identifying and locating voids and assessing where their centers are. Estimation of the centers of voids can be difficult in three dimensions, especially with objects “occluding” the voids, but our techniques, particularly interaction, allow this to be carried out.

7.3 Viewing Galaxies within an Isosurface

To assess the effectiveness of our screen door transparency on isosurfaces, we used a simple test where our colleagues were required to count the number of galaxies within a connected component of the isosurface. We generated a small clump of galaxies, set up the isovalue to a predetermined setting, and let our colleagues adjust any other setting except the isovalue and the isosurface visibility. Two evaluations were performed, one without stereo and the other with. Two different test data sets were used, one with 18 galaxies in the clump and one with 13 galaxies. We wanted a large enough number to make the task challenging, but not so large as to make it impossible or overly tedious. The galaxy positions were determined randomly.

Without stereo, our colleagues made extensive use of interaction. Sometimes the galaxies would be occluded by the isosurface, so they would adjust the view to better see the galaxy through one of the holes in the isosurface. For their first count, they found 17 galaxies. After some more interaction, especially rotation, they found all 18. Eventually, they tried the torsional rocking motion. Our colleagues reported that this helped to reveal the galaxies "hiding" behind the isosurface, but it made it more challenging to remember the galaxies they had just counted. A complaint from our colleagues was that the isosurface was too opaque and hid some of the galaxies. We were surprised that they did not adjust the galaxy glyph size parameter, as this may have alleviated some of the occlusion problems they were seeing.

During the stereo part of this task, Matt tried to "cheat" and zoom into the isosurface. Since the galaxy clump and isosurface were centered at the origin, the zoom factor could not get large enough to let geometry culling cut away part of the isosurface. At a high enough zoom level, some of the math for the trackball-style camera goes haywire, and things start behaving strangely. This occurred during Matt's interactions, but he simply used the "Home" key to get himself back to the default camera view. This problem of scaling the geometry to zoom in could be alleviated by using a flight-style camera control where the camera actually moves along the viewing vector. We attempted to use VTK's flight-style camera, but it is incompatible with the TK rendering widget we use for display.

By Matt's account, stereo helped a "ton" in this task. He found it easier to keep track of previously counted galaxies because of the third dimension. In addition, if a galaxy was occluded from one eye, it was likely not occluded from the other eye. This helped with the galaxy hiding problem.

Our evaluation showed that the isosurface with screen door transparency was effective for revealing the galaxies within a connected component. A shortcoming of our evaluation was that it did not present galaxies outside the isosurface that might potentially interfere with counting the galaxies inside the isosurface. We feel that the same techniques as those applied to our test data sets would clearly reveal the galaxies inside and outside the isosurface.

7.4 Real Data Evaluation

Having suffered through our evaluation tasks, Matt said of viewing the real data, "This is much better than that crappy random distribution stuff!" Right away, Matt and Jim identified a particular void they knew well from their previous plotting techniques. At one point, Jim saw a line of galaxies and said, "Ooh, there's something right there!" Clearly, our colleagues were excited to see their actual data.

After viewing the data set for a while, Jim made this observation: "I don't see filaments so much as bubbles." This observation is exciting because it says something about the topology of the universe and therefore the formation of the universe. Further follow up indicated his excitement at comparing the observed data to the simulation data. Prior visualization work of the simulation data reveals that the galaxies in simulations tend toward filamentary structures [1]. This is in disagreement with the observational data, according to Jim: "We were impressed by what looked like evacuated 'bubbles' with galaxies defining the surfaces of these void areas, rather then the 1D filaments that *seem* to be so prevalent in the numerical simulations of structure in the universe."

Matt had some observations of his own to offer. He said, "Viewing it in 3D makes it look kind of weird, in a sense." He has been used to plotting his data in two-dimensional plots, so it will take a while to get to know his data in 3D. About stereo, he said, "[Stereo] would be helpful on a small scale, where no stereo would be helpful on a larger scale."

Matt had two requests for us. He requested the ability to display galaxy information when a particular galaxy is queried. We have since implemented his request as a TK label widget that displays the RA, DEC, and cz, coordinates, the name of the survey that collected that galaxy data, and a galaxy ID. Matt also requested that we increase the scale of the cluster glyph relative to the galaxy glyphs to make it take up more volume. We decided to make the cluster size a separate adjustable parameter and have implemented it as such.

8 Lessons Learned

There were several interesting points that we learned from this project. It was interesting to watch scientists actually use the application and hear their comments. The scientists were a little disoriented the first time they looked at their data in 3D, as they were so used to viewing their data in 2D plots. The cues that were included in the visualization, such as the cluster glyphs and axes, definitely helped them navigate through the visualization, but they were so used to the 2D version that the 3D looked weird. Hopefully this new perspective on their data will allow them to understand it better. Another interesting lesson was how little they used isosurfaces when visualizing their data. While we were only able to observe them for a short time while they looked at their own data, they just wanted to look at the discrete galaxy glyphs. Perhaps when they have more time to explore their data they will use the isosurface visualization more extensively.

9 Future Directions

While we were only able to complete the first goal of our visualization design, the other two are still very interesting to our colleagues. The second goal, the comparison of simulation data to observational data is an area that does not seem to have been paid much attention. The third goal is interesting as it addresses how redshift might affect their observation data. The modules provide much of the base framework required to build visualizations to answer these other questions. For the second goal, the simulation data would have to be adjusted so that it matched with the observation data, and a GUI that sets up two render windows would need to be created. For the third goal, the extra visualization techniques would need to be implemented in the visualization code.

By our colleagues' preliminary findings from our visualization system, we would consider adding user-controlled widgets that could help locate and define the bubbles in the galaxy distribution. These widgets would most likely be in the form of sphere glyphs that the user could position and scale interactively to fill the voids.

Our colleagues are interested in continuing this work in the future, and we are interested in working with them. Hopefully, with continued interaction, they will be able to leverage our visualization system to find new knowledge about the formation and structure of the universe from their data.

References

[1] Cosmological n-body simulations. http://www.mpa-garching.mpg.de/GIF/#pics.

[2] G. O. Abell, Jr. H. G. Corwin, and R. P. Olowin. A catalog of rich clusters of galaxies. *Astrophysical Journal Supplement Series*, 70:1–138, May 1989.

[3] M. Colless and Q. Parker. 6df survey plan. http://www2.iap.fr/users/gam/6dF/6dF_survey_plan.html, 2000.

[4] A. Jenkins, C. S. Frenk, F. R. Pearce, P. A. Thomas, J. M. Colberg, S. D. M. White, H. M. P. Couchman, J.A. Peacock, G. Efstathiou, and A.H. Nelson. Evolution of structure in cold dark matter universes. *Astrophysical Journal*, 499(1), May 1998.

[5] R. C. Nichol, C. J. Miller, and T. Goto. The interplay of cluster and galaxy evolution. *Astrophysics and Space Science*, 285:157–165, 2003.

[6] Q. Parker. 6df: The new automated multi-object fibre-optic spectroscopy system on the ukst. http://www.roe.ac.uk/ifa/wfau/6df/6df.html, 2001.

[7] F. R. Pearce and H. M. P. Couchman. Hydra: a parallel adaptive grid code. *New Astronomy*, 2:411–427, November 1997.

[8] J. H. Simonetti. How to do astrometry. `http://www.phys.vt.edu/~jhs/SIP/astrometry.html`.

[9] Colin Ware. *Information Visualization: Perception for Design*. Morgan Kauffman, 2nd edition, 2004.

[10] E. L. Wright. Doppler shift. `http://www.astro.ucla.edu/~wright/doppler.htm`, 2002.

10 Appendix A: Evaluation Survey

The following was presented to our colleagues during the evaluation of our visualization system:

10.1 Evaluation Survey

We have shown you how to manipulate various parameters in GalaxyViewer. Now we must assess how useful our visualization techniques are for answering some of the questions you wish to answer with this visualization project. We will present a series of questions for you. Please answer the best that you can. Don't worry about getting any questions "wrong". You are not being tested, rather, our visualization design is being tested.

Please feel free to vocalize your thought process. An example statement might be, "I am attempting to determine X, so my strategy is to Y. I wish I could do Z to help me in this goal." In order to provide a fair assessment of our visualization techniques, we cannot help you during the evaluation.

At the end of each task, please notify the observers that you are finished. They will set up the next task, after which you may proceed.

11 Task 1

Your first task is to identify as many familiar shapes in the data set as you can. Please do not use the stereo goggles for this task. Please list the familiar shapes you find below, leaving room to the right of each listed shape for Task 2:

12 Task 2

For the shapes you found in Task 1, please locate, as best you can, their centers in RA-DEC-rv coordinates. Please write these coordinates next to the shapes you listed above.

13 Task 3

Please put on the stereo goggles and repeat Task 1.

14 Task 4

Please put on the stereo goggles and repeat Task 2.

15 Task 5

Please take off the stereo goggles. For this task, your goal is to locate voids, or global underdensities. For each void you find, please estimate the RA-DEC-rv coordinates of the center of the void.

16 Task 6

Please put on the stereo goggles and repeat Task 5.

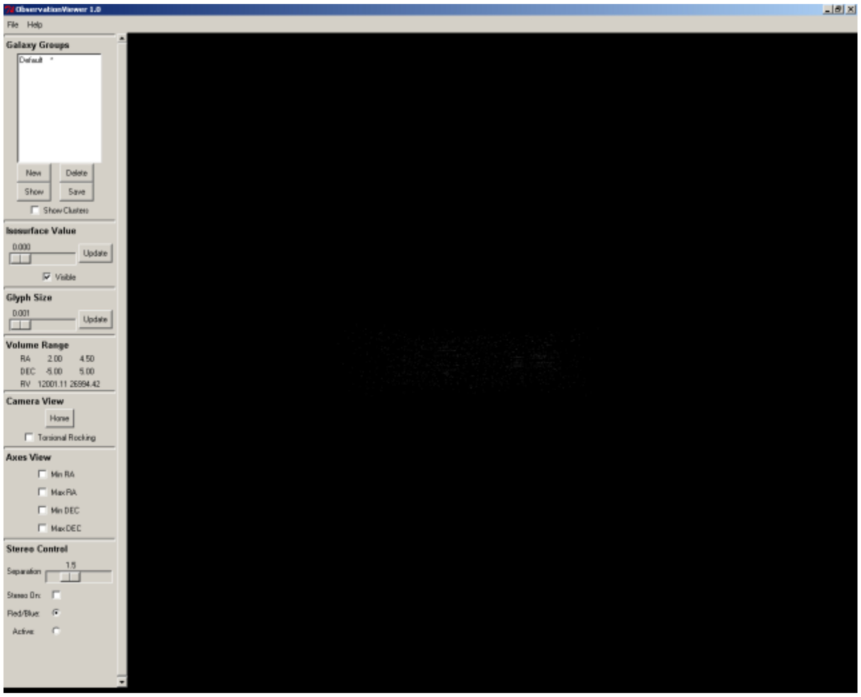
17 Task 7

Please take off the stereo goggles. For this task, we have loaded up a data set and set the isosurface to enclose a set of galaxies. Without turning the isosurface off, how many galaxies are inside the isosurface?

18 Task 8

Please put the stereo goggles on and repeat Task 7.

19 Appendix B: Evaluation Screenshots



Galaxy Groups

Default *

□ Show Clusters

Isosurface Value

0.000

☑ Visible

Glyph Size

0.001

Volume Range

RA 2.00 4.50

DEC -5.00 5.00

RV 12001.11 26994.42

Camera View

□ Torsional Rocking

Axes View

□ Min RA

□ Max RA

□ Min DEC

□ Max DEC

Stereo Control

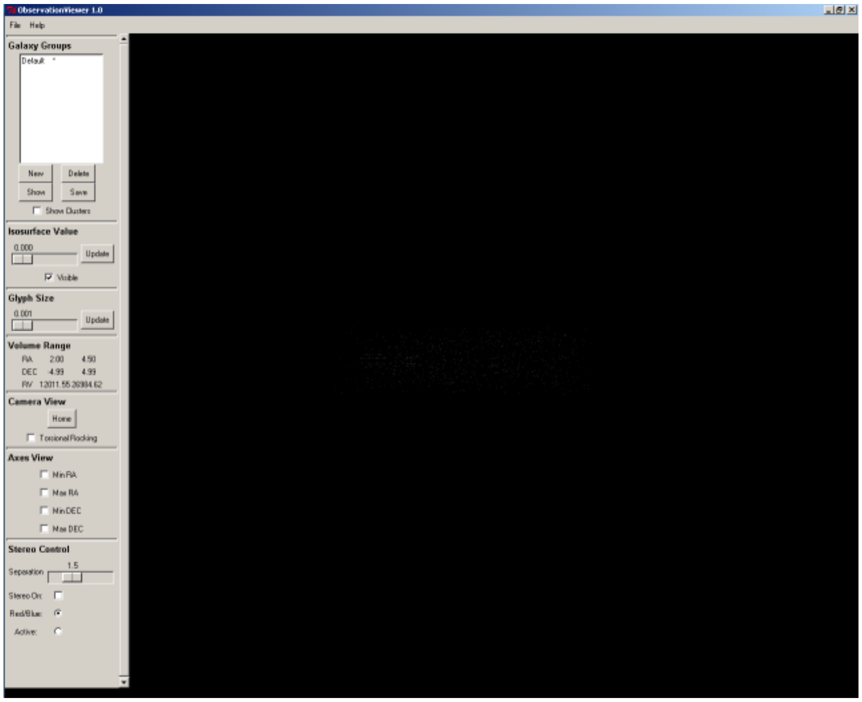
Separation 1.5

Stereo On: □

Red/Blue: ●

Active: ○

Figure 7: Our application as it appeared for the first evaluation task. Note that in this and the next three figures, the camera zoom and galaxy glyph size settings are set such that the galaxies are difficult to discern.



Default *

☐ Show Clusters

Isosurface Value 0.000

☑ Visible

Glyph Size 0.001

RA	2.00	4.50
DEC	-4.99	4.99
RV	12011.55	26984.62

☐ Torsional Rocking

☐ Min RA

☐ Max RA

☐ Min DEC

☐ Max DEC

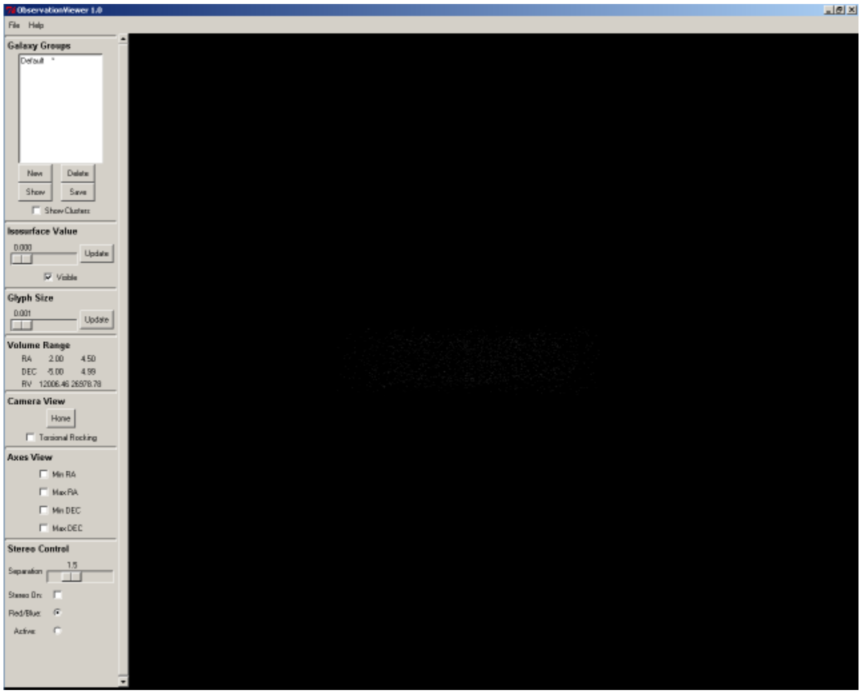
Separation 1.5

Stereo On: ☐

Red/Blue: ●

Active: ○

Figure 8: Our application as it appeared for the second evaluation task.



Default *

☐ Show Clusters

Isosurface Value 0.000

☑ Visible

Glyph Size 0.001

RA	2.00	4.50
DEC	-5.00	4.99
RV	12006.46	26978.78

☐ Torsional Rocking

☐ Min RA

☐ Max RA

☐ Min DEC

☐ Max DEC

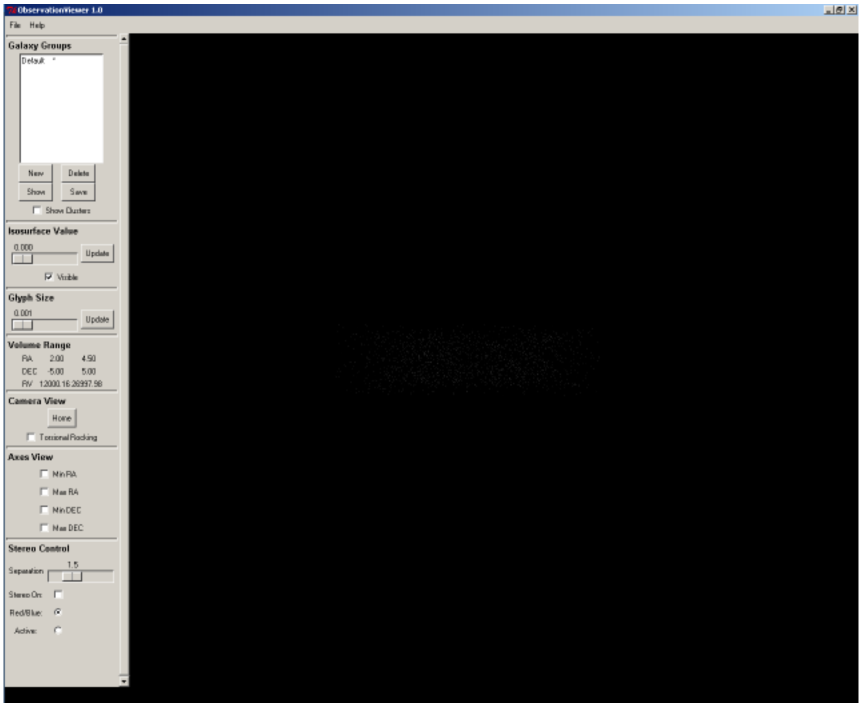
Separation 1.5

Stereo On: ☐

Red/Blue: ●

Active: ○

Figure 9: Our application as it appeared for the third evaluation task.



Default *

□ Show Clusters

0.000

☑ Visible

0.001

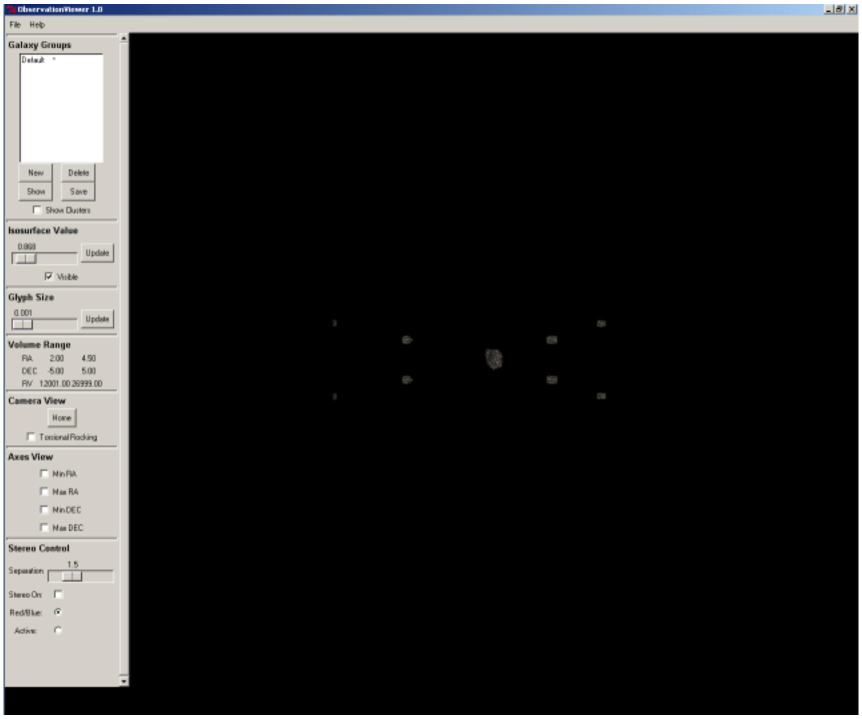
RA 2.00 4.50
DEC -5.00 5.00
RV 12000.16 26997.98

□ Torsional Rocking

□ Min RA
□ Max RA
□ Min DEC
□ Max DEC

Separation 1.5
Stereo On: □
Red/Blue: ●
Active: ○

Figure 10: Our application as it appeared for the fourth evaluation task.



Default *

□ Show Clusters

Isosurface Value: 0.868

☑ Visible

Glyph Size: 0.001

RA	2.00	4.50
DEC	-5.00	5.00
RV	12001.00	26999.00

□ Torsional Rocking

□ Min RA

□ Max RA

□ Min DEC

□ Max DEC

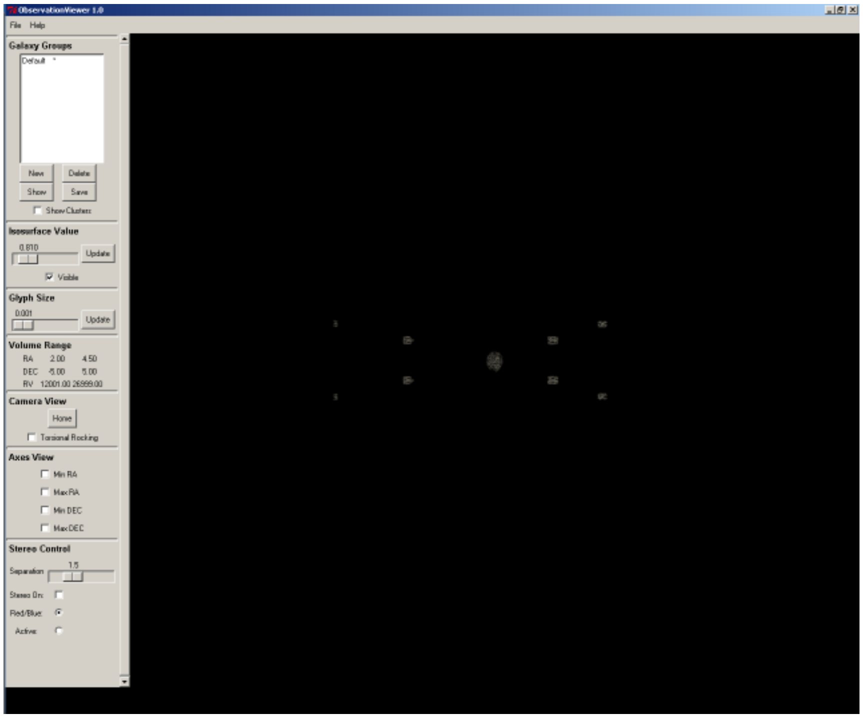
Separation 1.5

Stereo On: □

Red/Blue: ●

Active: ○

Figure 11: Our application as it appeared for the fifth evaluation task.



Galaxy Groups

Default *

□ Show Clusters

Isosurface Value

0.810

☑ Visible

Glyph Size

0.001

Volume Range

RA 2.00 4.50

DEC -5.00 5.00

RV 12001.00 26999.00

Camera View

□ Torsional Rocking

Axes View

□ Min RA

□ Max RA

□ Min DEC

□ Max DEC

Stereo Control

Separation 1.5

Stereo On: □

Red/Blue: ●

Active: ○

Figure 12: Our application as it appeared for the sixth evaluation task.