Pixel-based Overlays for Navigating a Galaxy of Observations

Timothy Luciani+, Rebecca Hachey+, Daniel Q. Oliphant+, Brian A. Cherinka*, G. Elisabeta Marai+

University of Pittsburgh, +Department of Computer Science, *Department of Physics and Astronomy

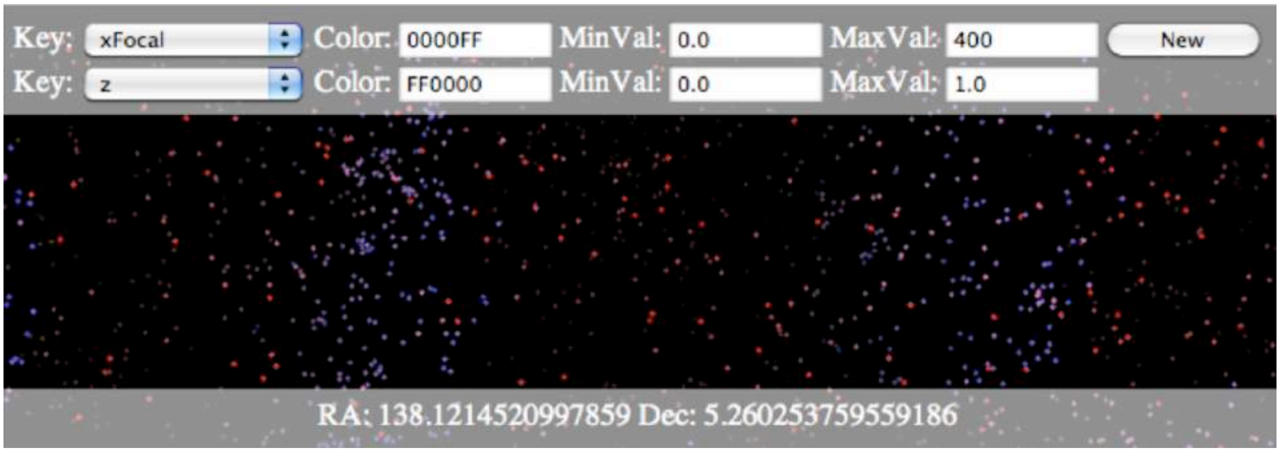

Figure 1. 831 points from the Sloan Digital Sky Survey database are visualized efficiently using pixel-based overlays. Two query results based on two different attributes are overlaid (red for redshift, blue for the focal ratio of the telescope; brighter intensities correspond to greater values), revealing spatial patterns in conjunction to attribute overlaps. The overlays are efficiently calculated and transferred through a WebGL client-server architecture. The interface allows for flexible control of the resulting visualizations.

1 Introduction

There is an incoming flood of information to the Astronomy community with the development of new telescopes. Over the next decade the amount of information available to the typical astronomer will grow by two orders of magnitude to the Peta scale, thanks to programs such as Pan-STARRS (Panoramic Survey Telescope and Rapid Response System) [4] and LSST (Large Synoptic Survey Telescope) [5]. Moreover, astronomy analyses often require pooling information from multiple data sources, such as a number of measurements from different telescopes at different wavelengths of light.

However, we lack an easy-to-use and scalable way to analyze anything beyond the most basic data on the thousands to billions of individual events and objects studied. Visually cross-registering astronomy data across multiple sources comes with its own challenges [7]; for example, images may be acquired at different resolutions and may only partially overlap with each other. The analysis process is further hampered by limited screen space and limited bandwidth (upwards of one minute may be required by large Sloan queries). For example, Fig. 2 shows a typical Google Sky [1] visualization of about 60 annotations (of different types) on objects in a small part of the sky. Even at such low numbers, the glyphs used to represent each piece of data quickly fill the screen space and become difficult to sift through.

Department of Computer Science, University of Pittsburgh, Pittsburgh, PA 15230
Email: {tbl8, reh59, dqo1, bac29, marai}@pitt.edu

IEEE Symposium on Large Data Analysis and Visualization
October 23 - 24, Providence, Rhode Island, USA
978-1-4673-0155-8/11/$26.00 ©2011 IEEE

To address the challenges of querying multiple sources, limited screen-space and limited bandwidth, we propose to use a pixel-based overlay approach, computed and implemented through a WebGL client-server architecture; our prototype was developed in close collaboration with astronomy experts. An interactive demo of the prototype is available.

2 Methods

To compute and transfer pixel-based overlays, we designed a client-server architecture, in which a web-browser client interacts with astronomy databases through a PHP-based web-server. For demonstration purposes we use a synthetically-generated MySQL database as well as the Sloan Digital Sky Survey (SDSS) [3] and the Faint Images of the Radio Sky at Twenty-Centimeters (FIRST) astronomy databases.

The PHP script on the server interacts directly with the astronomy databases, sending specific queries and then generating one or more pixel-based representations (PNG format images) based on the results of the query. The PHP script takes the client requirements --- desired resolution of the output image, the minimum and maximum right ascension and declination values (roughly, the latitude and longitude coordinates of an astronomical object), attribute thresholds, desired color-mapping, and any other optional filters on the other parameters present in the database --- and submits them as a query to the databases. Upon receiving a response from the databases, the PHP script creates one or more new PNG images and proceeds to draw on the image each tuple returned by the query. The right ascension and declination columns in each tuple are used to position the drawing within the image. The closer the value of the key attribute is to the maximum threshold, the brighter the color will be drawn at that point. All data tuples are added to the images, which are then compressed and returned over the network to the client application.

Figure 2. Traditional annotations in Google Sky. The picture contains only about 60 annotations, yet the information is almost illegible.

The client receives the images, cross-registers them, computes the pixel-based overlay and displays it. The cross-registration and overlay operation involves projecting each image from a rectangular grid onto the sky, zooming in and out to account for telescope parameters, changing transparency etc. The visualization is accomplished using WebGL -- a web standard that provides a 3D graphics API implemented in a web browser without the use of plug-ins [2]. The client finally renders the scene to an HTML Canvas element, where the visualization scenegraph consists of a sphere with the camera at the center looking out.

3 RESULTS

The prototype system was tested on both synthetic and real datasets. The tests were run using a server with Dual Dual-Core 3.6GHz Xeon processors and 12GB RAM. The client machine had a single 2.4 GHz Intel Core 2 Duo processor, 4GB RAM and an ATI Radeon HD 2400 128 MB graphics card.

The first, synthetic dataset was generated by specifying a redshift parameter (a measure of how far objects are away from us) that would result in a gradient in the final image. The purpose of this test was to ensure that the visualization accurately represented data patterns and did not introduce artifacts (Fig. 3). In terms of performance, the average time to query the database and generate the image was 3.51 seconds. The result mapped 10,000 data points to a 1200x800 pixel image that is 382,337 bytes in size in compressed PNG format.

The second dataset was taken from the Sloan Digital Sky Survey SpecObj astronomy database (Fig. 1). The average time to query the SDSS database and generate the image was 2.03 seconds. The result efficiently mapped 831 data points to a 1200x800 pixel image that is 110,185 bytes in size in compressed PNG format. Rendering to a canvas of 1024 by 768 pixels, the interface ran at 35 frames per second.

Finally, the third and most complicated dataset was taken from the Sloan and FIRST databases and showcases correct cross-registration of the two different sources at a 50/50 transparency (see additional image).

To further evaluate the benefits of this approach, we conducted informal interviews with four senior and junior astronomy researchers. Preliminary feedback shows they consider the approach "a promising beginning towards a tool for visualizing all-sky surveys. Many of the tools required have been implemented effectively." The expert users particularly appreciated the ability to combine separate sources of information without having to resort to cumbersome, external tools for image processing. The researchers are interested in applying this prototype to specific problems such as browsing large sets of objects and galaxy identification.

4 DISCUSSION AND CONCLUSION

These preliminary results show that pixel-based overlays have the potential to generate scalable, graphical representations of astronomy data. This approach may allow us to overcome bandwidth and screen-space current limitations in astronomy database visualization by following a WebGL - PHP client-server architecture. The advantages of this approach are its versatility and visual scalability (to the pixel level), enabling the visual analysis of large datasets. The resulting versatility allows for flexible control over the data and the client-side scripts. Accessing graphics hardware through WebGL further provides the users with a rich, graphics-accelerated web experience. Preliminary feedback from astronomy researchers emphasizes the benefits of visual analysis to this field.

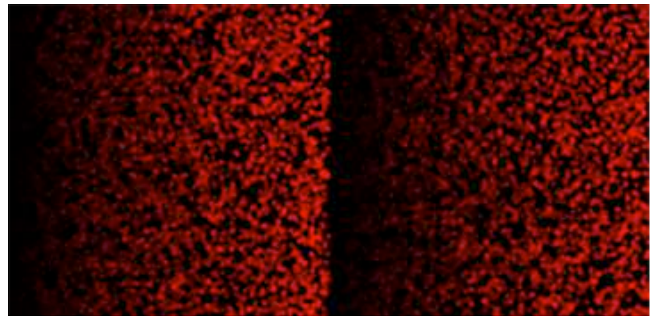

Figure 3. Overlay produced using synthetic data (for validation and performance evaluation purposes) that clearly shows the gradient pattern in the data. 10,000 data points are efficiently mapped to a PNG file.

ACKNOWLEDGMENTS

Research funded through NSF-OIA-1028162 and NSF-IIS-0952720. Many thanks to Andrew Conn, James Cavanaugh, Matthew Liegey, Matthew Seiler, Anja Weyant, Michael Wood-Vasey, Jeffrey Newman, Alex Labrindis, and Panos Chrysanthis.

REFERENCES

[1] *Google Earth*. http://Earth.google.com/sky.
[2] "WebGL - OpenGL ES 2.0 for the Web http://www.khronos.org/webgl/.
[3] *SDSS SkyServer DR7*. http://skyserver.sdss.org.
[4] PAN-STARRS www.pan-starrs.ifa.hawaii.edu
[5] LSST www.lsst.org
[6] FIRST http://sundog.stsci.edu/index.html
[7] Galaxy Zoo http://www/galaxyzoo.org