Improving the Visualization of Galactic Events Using Pixar's RenderMan

Paul Cassidy[1]
ptc6340@rit.edu

Tyler Kilburn[2]
kilburn_tyler@yahoo.com

Vincent Salemink[1]
vincentsalemink@msn.com

Reynold Bailey[1]
rjb@cs.rit.edu

Hans-Peter Bischof[1]
hpb@cs.rit.edu

[1]Department of Computer Science, Rochester Institute of Technology
[2]Department of Computer Science, Saint John Fisher College

ABSTRACT

Pixar's PhotoRealistic RenderMan® is a powerful graphics rendering system with underexplored potential for visualizing scientific data. We utilized PhotoRealistic RenderMan® to visualize simulations of galaxy mergers which are dynamic and contain large volumes of multidimensional data. Our goal was to emulate existing astronomical imagery such as images captured by the Hubble Space Telescope. In order to accomplish this we developed techniques for rendering individual stars and also took into account factors such as galaxy density. PhotoRealistic RenderMan® was used to create a movie from the provided galaxy merger data. In this paper we describe our approach and present the results. We also discuss how the rendering framework can be extended to incorporate the effects of light bending due to gravity.

Keywords
Visualization, RenderMan®, Simulation, Galactic events.

1. INTRODUCTION

Humans are extremely limited in their ability to process and understand raw scientific data. This is especially true for large datasets. Simulations of galactic events such as black hole mergers for example, generate huge volumes of complex multidimensional data. Scientific visualization is needed to help bring meaning to this information. Computer Generated Imagery (CGI) can be used to display such phenomena. Typically existing tools are used or new ones must be developed. Powerful tools such as Pixar's PhotoRealistic RenderMan® [Kau00] have strong roots in computer animation and in the film industry. However, PhotoRealistic RenderMan® is often viewed by the scientific community as a means of creating fantasy worlds and special effects and has largely been ignored as a resource for visualizing scientific data.

PhotoRealistic RenderMan® has several important characteristics that make it a viable and effective tool for scientific visualization. These include numerous advanced features for creating visual effects and the ability to use distributed processing to take advantage of large multi-core networks of computers to increase overall throughput. In recent years, several planetariums have begun using rendering packages such as PhotoRealistic RenderMan® to create immersive experiences for their visitors [Mol10]. In this paper we demonstrate how PhotoRealistic RenderMan® can be used to visualize simulations of galaxy mergers. Our aim was to create compelling imagery similar to the stunning images captured by the Hubble Space Telescope (see Figure 1).

Figure 1. Galaxy NGC 4414. Courtesy of NASA and the European Space Agency. Image generated by the Hubble Space Telescope.

The remainder of this paper is structured as follows: section 2 discusses background and related work; section 3 presents our visualization framework and rendering techniques; results are presented in section 4; we conclude and describe future work in section 5.

2. BACKGROUND AND RELATED WORK

The Center for Computational Relativity and Gravitation (CCRG) is a research group that utilizes mathematical modeling, supercomputing and data visualization to help further our understanding of astrophysical phenomena [RIT10]. Researchers at CCRG have developed a visualization framework called Spiegel, which is equipped with a convenient graphical interface for creating programs to extract, analyze, and render different views of the data [Bis05]. Spiegel was originally designed to utilize Java3D or JOGL to create the final images. Recently, Espinal et al. extended the Spiegel framework to utilize PhotoRealistic RenderMan® [Esp10]. They demonstrated the capabilities of the improved framework by creating renderings of galaxy mergers. However they utilized relatively simple shaders, which were not very realistic. Our aim was to create more realistic images that were closer in appearance to imagery captured by the Hubble Space Telescope. To accomplish this goal, we consider the appearance of individual stars and overall star density.

3. VISUALIZATION FRAMEWORK AND RENDERING TECHNIQUES

Individual frames of the final video sequence were produced using simulation files provided by the Center for Computational Relativity and Gravitation. The files contain positional data of every star and black hole for each timestep in a galaxy merger simulation. The Spiegel graphical interface was used to establish viewing parameters and a RenderMan® Interface Bytestream (RIB) file was then generated for each frame. RIB files are simply text files that contain a sequence of commands that RenderMan® compliant renderers such as PhotoRealistic RenderMan® can interpret and execute.

In reality, black holes are not visually perceptible because the gravitational pull of black hole is so strong that not even light can escape from its center. Since our focus was on realism, we opted to ignore the presence of black holes in our renderings and instead focused on creating more realistic stars.

Figure 2 shows an overview of our visualization framework. Prior to this work, the stars were typically rendered as simple spheres. For our approach, each star is actually a combination of three spheres with separate shaders. Using a modified version of RenderMan's built in Perlin-based noise function [Per85] and the built in spline function, the main surface color across the star was created. For the spline, five colors are used: a base color, usually white or black, and four colors similar to the desired star color. The color palette is shown in Figure 3. To simulate sunspots, we modified the built in noise function and used the RenderMan® mix function to add this into the overall surface color.

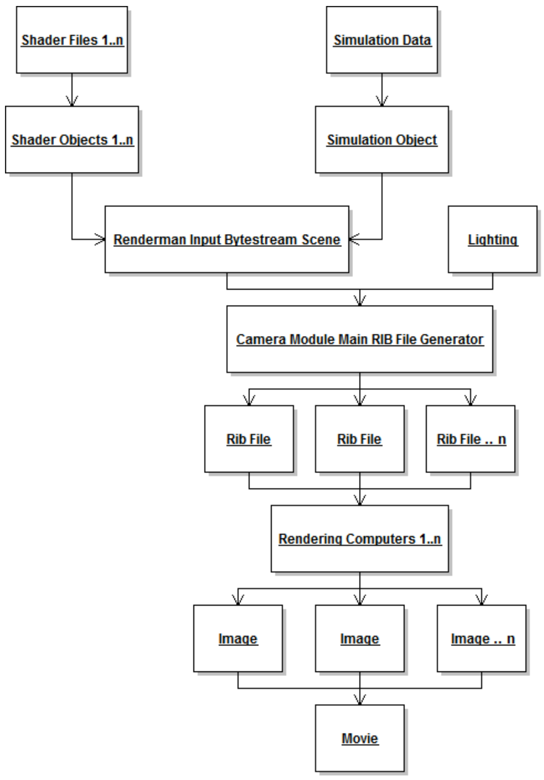


Figure 2: Overview of visualization framework.

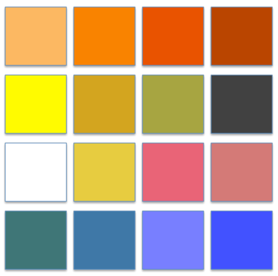
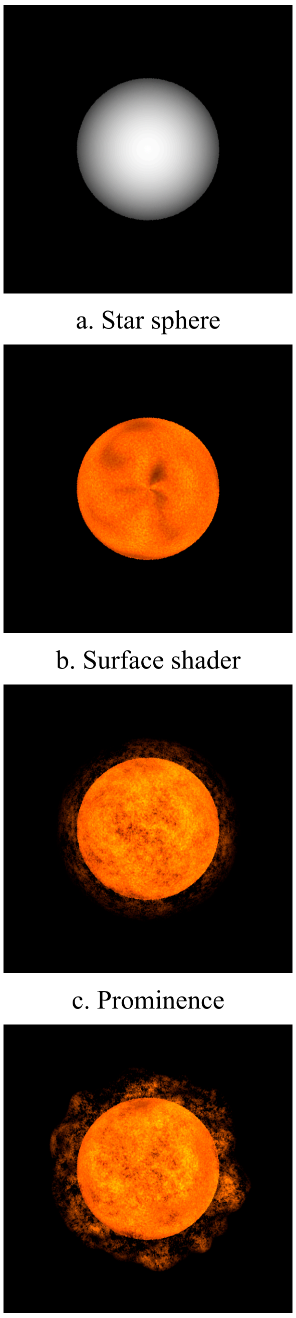


Figure 3: Palette of possible colors for stars.

To simulate the prominences of the stars, i.e. a cloud of gas elevated above the star surface, a second sphere is applied over the main sphere and an opaque, fog-like surface shader of the appropriate color is applied. The specific color of the prominence surface shader does not have to be the same as the surface color of a star. For example, a red star can have a yellow prominence and it would still look acceptable. Additionally, a displacement shader is applied to the prominence to further enhance the effect. Figure 4 shows the effect of each of these steps.


a. Star sphere

b. Surface shader

c. Prominence

d. Prominence with displacement

Figure 4: Combining various shaders to create more realistic looking stars.

The galaxy as a whole was imaged using subtle techniques that cannot be seen when looking at a single star. Every star in the simulation was bound to a transparent surrounding sphere with a radius 25 times larger than itself. The RenderMan® Shading Language's noise function was used to generate a believable regular pattern. The noise function was given the surface point of the sphere as a parameter and subtracting a constant value of 0.4 for all red, green blue (RGB) colors. The magnitude of the color was further determined by five octaves. Each of the octaves enclosed the noise function where noise was multiplied by a value of 3.6 lacunarity. After color values were calculated for each surface point, opacity was defined using the values in Table 1.

	Red	Green	Blue
Opacity	0.02	0.02	0.03

Table 1: Red, green, and blue opacity values

Rendering of the movies was performed using RenderMan Pro Server 15.1 on iMac 9.1 machines with 2.66 GHz Intel Core 2 Duo processors, and 4 GB 1066 DDR3 memory running OS X version 10.6.4. RIB files were generated from Spiegel visualization system for input to PhotoRealistic RenderMan®. The RIB files were dispersed among 10 machines where PhotoRealistic RenderMan® was used to render the frames. A simple Bash shell script was used to feed a single RIB file at a time to reduce overhead and memory consumption.

4. RESULTS

Rendering time for an average frame containing 14000 stars took approximately 4 to 6 minutes across workstations. The movies generated are on average 1 minute, 40 seconds in length at 25 frames per second. All frames had a 768 x 768 pixel resolution. Transitions between frames appear smooth but a careful viewer can sometimes perceive subtle discontinuities. The discontinuities only occur with slow moving stars at the edges of the simulation where they do not change positions enough between frames for the difference to appear. These transitions were hard to perceive and vary in severity depending on the input data, frame rate, and number of times interpolated between positions.

The effects of the final galaxy shader are most visible in the center of galaxy where many stars contribute to the final pixels. For a simulation of two super dense galaxies colliding with a larger galaxy we noted that as stars are pulled away from their home galaxies that this results in a visible reduction in the total brightness of their home galaxy mimicking what happens when real galaxies collide. Figure 5 shows a galaxy rendered using our approach.

5. CONCLUSION AND FUTURE WORK

We found that PhotoRealistic RenderMan® could be used to generate realistic looking visualizations of galaxies and galactic events. Our technique takes advantage of noise algorithms, which mimic reoccurring patterns found in nature. This, combined with the fluid motion of the galaxy, can be used to create believable movies that are based on scientific data as opposed to an artist's depiction. Movie quality appeared similar to that of the false color astronomical photos. More realistic images are possible with diligent shader design.

PhotoRealistic RenderMan®, while useful, does have limitations. When rendering large sets images we observed considerable slow down as size increased. Using a simple UNIX shell script we were able to reduce overhead and memory consumption by feeding one RIB file at a time to the renderer.

In the future we would like to visualize the effects created by the bending of light around black holes. However, RenderMan's ray-tracer is not set up to trace light along curves; it only traces light rays in straight lines from the camera and between objects. Another issue with this idea is that there is no explicit formula for the curves that the light follows around black holes, especially in the case where there are multiple black holes. In general the differential equations describing the shape of the space-time continuum cannot be solved analytically. The geodesics that the light rays follow need to be generated by integrating these equations numerically. Zink discusses an approach to general relativistic volume raytracing that accurately visualizes the matter around a single black hole [Zin08]. His approach involves generating geodesics by numerical integration. As part of our future work we plan to extend his technique to handle multiple black holes. Since the RenderMan® ray-tracer only works along straight lines it should be possible to make this algorithm work by dividing the curves into short straight line segments.

Finally, although RenderMan® has some limitations, it is a definitely a powerful and viable solution for visualizing scientific data. It readily provides a rich collection of shading and rendering techniques, with large libraries of shaders publicly available. A scientist with limited programming knowledge can easily modify them to suit their needs. RenderMan® should be used when appealing and realistic visualizations need to be created. However, for real time rendering it is not recommended. One final benefit to using RenderMan is that it runs on standard commodity hardware and can be distributed to multiple computers to take advantage of whatever a scientist has available.

6. REFERENCES

[Bis05] Hans-Peter Bischof and Jonathan Coles, A Movie Is Worth More Than a Million Data Points, Lecture Notes in Computer Science Publisher: Springer-Verlag GmbH, ISSN: 0302-9743 Subject: Computer Science Volume 3514/2005, Title: Computational Science ICCS 2005: 5th International Conference, Atlanta, GA.

[Esp10] Julio Espinal, Virginia Allen, Kwesi Amable, Reynold Bailey, and Hans-Peter Bischof, RenderMan's Power to Visualization's Rescue. 18th International Conference on Computer Graphics, Visualization and Computer Vision, 2010.

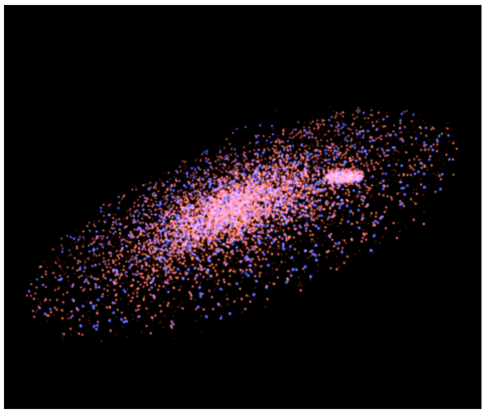
[Kau00] Kaufmann, M. (2000). Advanced RenderMan. San Diego, California: Academic Press. Nobel, S. (n.d.).

[Mol10] Karen Moltenbrey, Scientific Visualization, Computer Graphics World, Volume 33, Issue 7, (July 2010).

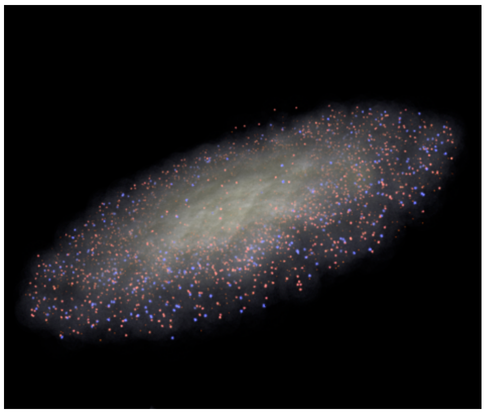
[Per85] Ken Perlin, An Image Synthesizer, Computer Graphics, Volume 19, Number 3, 1985.

[RIT10] Rochester Institute of Technology (2010). Retrieved October 20, 2010. Center for Computational Relativity and Gravitation, website: http://ccrg.rit.edu/.

[Zin08] Zink, B. (2008). Ray-tracing Black Holes. Saarbrucken, Germany, Germany: VDM Verlag Dr. Muller.



a. Galaxy shown without effect of galaxy shader.



b. Galaxy shown with effect of galaxy shader

Figure 5: Images rendered with and without our galaxy rendering technique enabled. This particular dataset consisted of 14000 stars.