Uniview - Visualizing the Universe

Staffan Klashed[1], Per Hemingsson[1], Carter Emmart[2], Matthew Cooper[3] and Anders Ynnerman[3]

[1]SCISS AB, Stockholm, Sweden
[2]American Museum of Natural History, New York, USA
[3]C-Research, Linköping University, Sweden

Abstract

This paper describes the development of the software system, Uniview, the motivation behind some of the most prominent features of the system and the strengths and challenges of running a development project in such close collaboration with the users. Uniview is a sophisticated system for the visual display and exploration of the enormous and complex data which the human race has gathered about the universe. This beautiful, fascinating data, with it's sheer size both in terms of data elements and the distances between the objects in the known universe, presents challenges to the developer at all levels: from basic rendering through representation and to data managment.

Categories and Subject Descriptors (according to ACM CCS): I.3.8 [Computer Graphics]: Applications—

1. Introduction

Humans have always been fascinated by the stars of the night sky and much of human scientific endeavor has focused on the exploration and mapping of space, leading to a progressively improved understanding of the physical processes governing astronomical phenomena.

The desire to convey the knowledge obtained by astronomers and astrophysicists to the general public has led to the construction of increasingly sophisticated planetariums, in which the night sky is projected onto a dome surface to create the illusion of stars and other astronomical objects. The history of the planetarium stems back to ancient Egypt and Greece, but in its modern shape the planetarium concept has developed rapidly over the past 100 years. In the late 1990's the concept took another step forward with the creation of the first digital planetariums, which opened up a new dimension of possibilities for the creation of content, making use of the techniques developed in computer graphics and visualization.

The first digital planetariums relied on high-end computers with advanced graphics capabilities, such as SGI multipipe parallel graphics systems. The rapid development of hardware for rendering has accelerated this development and domes, large screen theatres and digital planetariums are now, almost by default, equipped with the hardware capabilities to accommodate live and interactive presentations.

The introduction of digital domes and the desire to show interactive content has created an urgent need for software tools that enable the interactive navigation of curated databases, containing both observed and simulated data, while also providing high quality, near photo-realistic, rendering capabilities. This paper describes the development of Uniview, one of the first software packages tailored to provide such high quality, real-time and interactive graphics for exploring astrophysical data in digital domes. The paper will also discuss the future direction of Uniview, and elaborate on how SCISS, the company behind the software, will develop the platform to further support the creation of content.

The main contributions of this paper are:

1. A description of the Uniview platform.
2. Sample examples of usage of Uniview.
3. Presentation of a tool for content creation using Uniview.
4. Presentation of a tool for networked presentations connecting domes and flat screens world-wide.

The paper is organized as follows. We first give an account of related work, focusing on other similar software projects. We then provide an overview of the technical foundation and the recent development of production platforms. We conclude the paper with a summary and future outlook.

2. Background

Uniview relies heavily on the set of scientific databases it is designed to interface with. The primary database is the Digital Universe [Ame] atlas from the American Museum of Natural History (AMNH), a database that, in turn, is the product of a curation of over 40 separate datasets ranging from planets, through stars, exoplanets, galaxies and other phenomena all the way to the oldest, and most distant objects known, the quasars. The massive distances involved, combined with the enormous range of scales, causes major problems in producing usable images from such data and it was to accommodate the Digital Universe, together with a custom solar system, that was the very first goal of the Uniview development, see figure 2 and section 3.1.

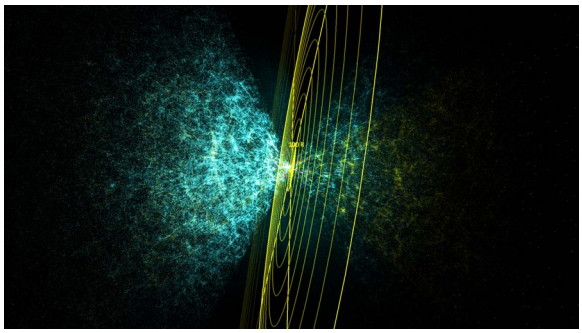

Figure 1: *The Digital Universe data atlas produced by the American Museum of Natural History, shown here rendered in Uniview. The concentric yellow rings indicate the plane of our galaxy, neatly dividing the two halves of the currently observable extragalactic space.*

Other important databases often used for work of this kind, and available in Uniview, include the 'OnEarth', 'OnMars' and 'OnMoon' servers [Jeta], containing satellite photographs and hosted by the JetPropulsion Laboratories (JPL) at NASA. Other key databases come from the Minor Planet Centre (MPC), North American Aerospace Defence Command (NORAD), and European Space Agency (ESA).

While Uniview is fast becoming a popular product, it is certainly not the only real-time software available in the planetarium market. Other competing products are offered by Evans & Sutherland, SkySkan and Spitz (now a part of E&S), to name the most prominent ones.

3. The Uniview Software Platform

Uniview is, today, a state-of-the-art visualization software with high-specification and high-performance modules for data processing, rendering and interaction. Reaching this point, however, has not been a quick or easy process. Instead the software has gone through a number of distinct design and development phases. This evolutionary process has been driven by improving hardware and software capabilities, the appearance of new and interesting datasets, and the requirements of other parts of the vision for Uniview in the dome planetarium market but, most importantly, by user needs and desires.

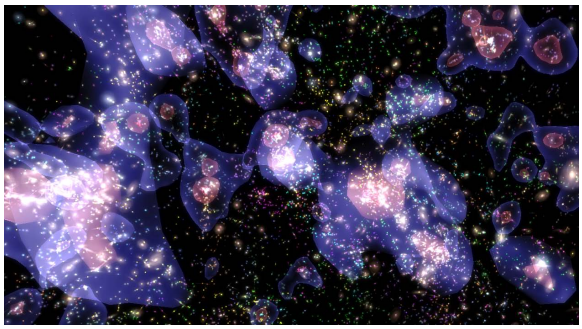

Figure 2: *Uniview supports a multitude of visual representations including high quality rendering of textured and transparent objects.*

One of the most fundamental and critical components of Uniview is the rendering engine for points, lines and grids. This is what is used to turn the Digital Universe data into pixels and textures in 3D. The first version of the engine was based on the OpenGL Performer API provided to application developers by SGI. The rendering engine was developed to accommodate a specific approach - to make points look like stars. The technique used had previously been developed in the Partiview software [Nat, Lev03] which had introduced a two-pass rendering scheme for stars: one pass with a point and one pass with a textured polygon. The texture was an exponential 'glow' and was scaled with the apparent magnitude of a star and then rendered without perspective projection to give the proper look from any vantage point. The rendering engine placed heavy demands on the graphics hardware of the time and proper culling was required. In the most complex datasets, with a few hundred thousand points, frame rates could drop to just 12-15 fps on an SGI Onyx2 system.

Over the years, the Uniview rendering engine has been refined to fully make use of the rapid increases in rendering performance of the modern GPU in today's graphics cards. The massive increase in performance for lower end computer systems soon, as in other application areas, led to a more or less complete change in hardware platform from SGI supercomputers to PC clusters for Uniview. An increase in rendering quality and performance came from the introduction of programmable vertex and pixel shaders which meant that advanced effects could be created that were previously not possible without significantly increasing the complexity of geometry rendered. The load balancing between the CPU and GPU has changed dramatically as vertex and pixel throughput has increased on the GPU. With lower ver-

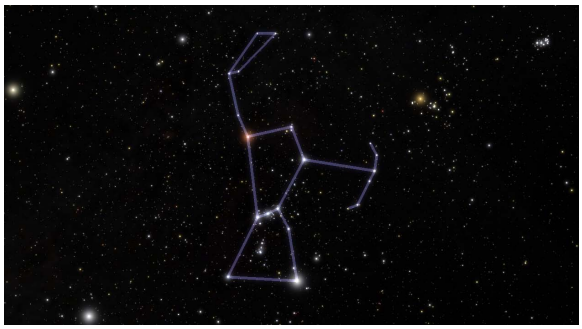

Figure 3: *The stars of the night sky rendered as points and halos using Uniview, and here outlining the constellation of Orion.*

tex and pixel throughput, a feature like culling was primarily a job for the CPU, whereas now it is a lot less important or even handled by the GPU itself. Physical synchronization of GPUs through the 'genlock' feature then allows clusters of PC boxes to swap frames at the same time, overcoming the major problem of 'tearing' in multi-channel displays.

Uniview currently supports single box systems with one or more GPUs and one or more graphics outputs, as well as high performance PC cluster-based, multi-channel systems. Techniques have been developed to deal with a wide range of problems in the rendering of astronomical phenomena. In the following subsections we describe some of the underlying ideas and the features in Uniview that have been developed over the past 5 years.

3.1. The Scalegraph

One of the fundamental ideas behind Uniview was to merge a solar system representation with a representation of deep space databases like the Digital Universe. Such a representation had been developed in the Japanese project called the Solar System Simulator [Sol]. One of the main challenges faced when integrating a solar system simulator with the Digital Universe is dealing with the enormous range of scales present, on computers with finite numerical precision, currently 32 or 64 bit floating point arithmetic. This was the motivation behind the development of the *Scalegraph*, which integrates the changes of scale into the scenegraph changes.

As an example, consider a representation of the solar system with an artificial satellite, such as the International Space Station (ISS). As modellers, we want to use an appropriate unit of scale when we model the ISS mesh, such as centimetres, but at the same time we need to be able to show the Andromeda galaxy in the night sky behind the ISS. Andromeda is 2.5 million light years away from the ISS, a distance of approximately 2.365×10^{24} centimetres. The images in figures 5 and 7 show these two scenes from the the two viewpoints, respectively. The depth buffer of the graphics pipeline is not capable of dealing with floating point numbers on that scale. Also the modelview matrix, with any available precision, is not capable of representing distances of this scale. So problems arose in many areas, such as depth sorting, z-buffering, navigational precision, and jittering of distant objects.

The Uniview Scalegraph employs a solution that divides the entire representation of the universe into different scenes. Each scene has an origin, a unit and a radius. As such, all scenes are spheres inside other spheres. The exact definition of a single scene can be chosen more or less arbitrarily. The definition of a scene relies on a choice of unit that gives the optimal numerical range for the data set to be displayed. Other scenes can then be expressed with the correct relative scale to one another, given the correct relation of units. For example, consider a scene called 'earth' with its origin at a position within another scene called 'solarsystem'. The earth scene uses kilometres as unit, while the solarsystem scene uses astronomical units. At each frame, Uniview identifies which scene currently contains the camera. It also knows, by default, the transformation matrix between the earth scene and the solarsystem scene. Then all objects in other scenes, such as the planet Mars in the 'mars' scene for example, are drawn using a new modelview matrix: $M = (C + P \cdot |D|) \cdot S \cdot T$, where C is the translation from the origin to the camera in the scene that contains it, D is the vector from the camera to the object, P is the radius of the scene that currently contains the camera, S is the scale difference between the scene containing the camera and the scene of the projected object, and T is the transformation matrix from the scene of the object to the closest common parent with the scene containing the camera. This projection of objects using the modelview matrix is called the Scalegraph, and is the fundamental feature of how Uniview handles the constraint of finite numerical precision (figure 4).

Another important aspect of the scalegraph is how to define the radius of each scene. Since this radius is fundamental to how close objects are projected to the camera, it is essential to ensure that all objects fit inside the frustum of the near and far clip planes. The radius is calculated for each object as $R = B + MAX(C) - CR$, where B is the bounding sphere of the scene containing the camera, $MAX(C)$ is the bounding sphere of the largest object to be projected, and CR is the bounding sphere of the object currently projected. The radius can also be used to define the far clip plane, and the near clip plane can be defined as the radius divided by the depth buffer precision.

Since the fundamental work on the Uniview platform took place, the *scale* problem has been addressed from different perspectives and by a number of different research and development groups. One of the early efforts being the SGI project that had the same name as the database from AMNH - Digital Universe. It can also be noted that several papers on

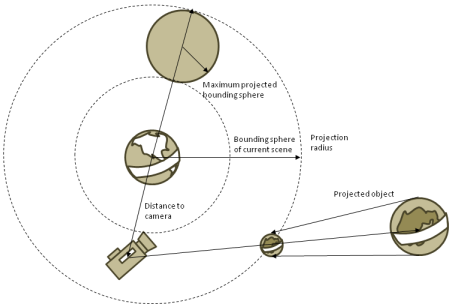

Figure 4: *The scalegraph projection. Objects are rendered with a modelview calculated as the distance from the origin of the current scene to the camera, plus direction from camera to object times the optimal projection distance.*

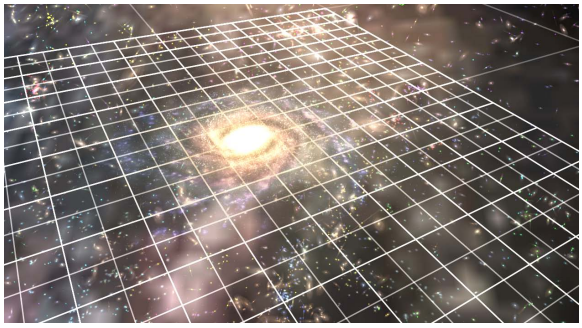

Figure 5: *The ScaleGraph provides a seamless visual representation across the huge range of scales required to visualize such diverse data.*

how to solve the scale issue have been presented. See [FH07] for a useful review.

Numerical precision in the graphics pipeline is becoming substantially higher with new generations of GPUs but, given the extraordinarily wide range of scales in the astronomical universe the problem is unlikely to ever completely disappear, meaning that some approach to this issue will always be necessary.

3.2. The Uniview Navigation Model

Another fundamental feature of Uniview is the camera model (figure 8). An inertia-based exponential drift model was developed for Uniview that ensures that the navigation remains smooth, regardless of the level of skill of the user. Based on the experience from several trials with live audiences, the importance of keeping camera motion smooth

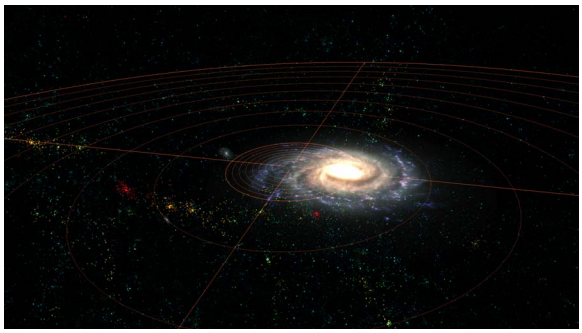

Figure 6: *The use of distance measures embedded in the visualization is an important feature in Uniview, here shown as the distance from the earth in the galactic plane.*

Figure 7: *The International Space Station in its low earth orbit. The scattering effects of the atmosphere are clearly visible at the horizon. The weather patterns are derived from KML files fetched in real time from a server at JPL and updated frequently.*

and cinematic at all times was taken into account in the design of the interaction scheme. One of the keys is the exponential camera drift, which is an automatic transmission that increases camera speed with the distance from a point of interest, the camera's pivot point. Camera inertia makes the camera come to halt gradually rather than suddenly, and also makes acceleration and deceleration happen as if the camera has a mass and is exposed to a force. Camera velocity is calculated based on the distance to an object of interest, chosen by the user, according to the formula $V = D \cdot DT \cdot (1 - cos(\pi/2 \cdot T)^3$, where T is the time from when user input starts to a user-defined later time, when inertia is fully overcome. DT is the time step for each loop in the application, to overcome sensitivity to frame-rate, and D is the distance to the object of interest.

Another important navigational feature is the ability to fade in and out the individual datasets when the camera is within a certain zone. This is a very useful feature as it saves the user from having to manipulate dataset parameters, switching them on and off, while navigating interactively.

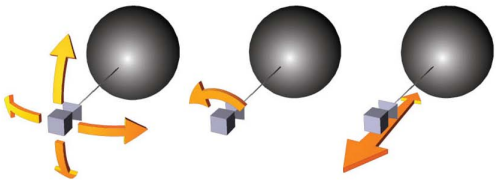

Figure 8: *The camera model. The user navigates around a pivot point, a centre of interest. The camera velocity scales with the distance to that point, enabling smooth and easy control as the distance changes.*

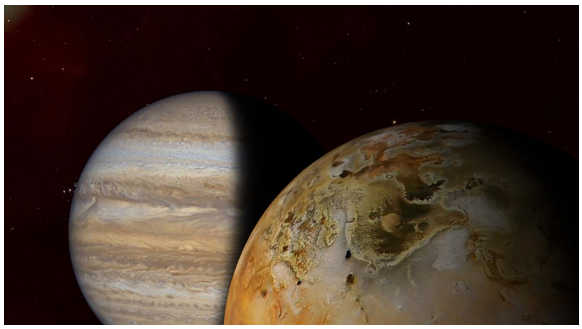

Figure 9: *The navigation model allows for smooth navigation and makes it easy to explore all of the data bases. This is of critical importance when moving close to large objects which cover large areas of the field of view as these can be very disturbing to the audience.*

Together, these described features are packaged by SCISS into a feature set called 'Flight Assist', which is still today an important part of Uniview for live presentations.

3.3. The Planet Renderer

The representation of planets in Uniview has been essential from the very first software prototype. Initially, the *clipmapping* feature that was available in OpenGL on SGI machines was used to texture planet surfaces. Clipmapping was built on the idea of image pyramids, originally developed at Xerox PARC and used in the Map Viewer application. As a consequence of the development of alternative platforms for multi-channel rendering, Uniview had to support clipmapping on other systems as well. A migration away from clipmapping on the SGI platform to PC clusters was made later.

An implementation of the ROAM 2.0 [Duc] technology for planetary rendering was chosen as the solution. The system is able to pre-slice a high resolution image and store it in a local cache from which the real-time system reads at runtime. ROAM 2.0 supports both texture and height-map information and is a fully functioning replacement for clipmapping. Later WMS and KML [The] were identified and implemented as crucial components for planetary rendering within Uniview.

Figure 10: *Multiple visual effects and terrain modelling work together to create realistic representations of earth and other planetary surfaces.*

The representation of planetary atmospheres is a key component in the generation of realistic looking images of planets as seen from low orbits. An atmospheric scattering simulation has been implemented to address this need. The simulation approximates both Mie and Rayleigh scattering of light. This approach allows for the creation of an atmosphere that is controllable through physical parameters rather than user-adjusted rendering parameters, and has good visual appearance both from a distance and from within the atmosphere. Figure 10 shows an image, captured from Uniview, showing the quality of the atmospheric rendering. Similar atmospheres can also be produced for other planets. This work was partly inspired by a similar project implemented by the work of O'Neil from NVIDIA [O'N]. The atmospheric simulator also enables the creation of realistic sunrises and sunsets, and also supports shadowcasting with simulation of the umbra and penumbra effect of eclipses. This last feature has been used to produce images showing solar eclipses both on earth and on other planets where, in the cases of the gas giants with their many satellites, eclipses can be a very frequent occurrence.

With the implementation of a KML renderer, the complete planetary rendering feature set was packaged and given the name "Geoscope", inspired by the vision of Buckminster Fuller to provide a vehicle to visualize *geostories*.

3.4. Spice and Orrery

Initially Uniview had relied on an analytical solution to position planets and moons on the night sky. While good enough for a general explanation model of the solar system, this solution does not suffice when it comes to simulating something like the Cassini mission to Saturn, the dynamics of the Galilean moons of Jupiter, or lunar and solar eclipses.

This increased need to display spacecraft on planetary

missions required the use of the NASA Navigation and Ancillary Information System (NAIF), the engine from which had previously been used by the Denver Museum of Nature and Science (DMNS) in their Cosmic Atlas project. Collaboration with DMNS helped incorporate this functionality, which uses publicly available data packets from NASA called SPICE kernels [Jetb] for all planetary bodies and spacecraft, within Uniview. These kernels are not designed for real time use, however, having been developed with the goal of stable and optimized behaviour over long periods of time, such as when running a simulation over several decades. The results of our reimplementation of the SPICE analyser has been a system which provides accurate and smooth tracks for the various natural and artificial objects in the real-time display. One very clear example of this use is shown in figure 11.

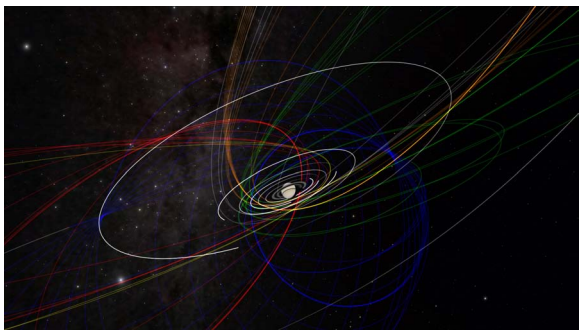

Figure 11: *NASA SPICE kernels are required to accurately plot positions of planetary bodies and spacecraft such as Cassini, here shown with its orbit colour-coded to indicate the specific mission phases in its four year exploration of the Saturnian system.*

3.5. Uniview Producer

Even though Uniview has primarily been designed to be used as a real-time and interactive application, there is also a need to be able to pre-record entire shows or parts of shows which can be incorporated into live, interactive sessions. This requirement has been the motivation for the development of the 'producer' toolset. It was the production of the AMNH show "Field Trip to the Moon" which pioneered the need for timeline editing and frame recording within Uniview.

The producer software is a timeline editor that enables users to control properties of Uniview over time, interpolate between pre-recorded segments using appropriate mathematical methods and add narrative elements such as audio, text and images to quickly create a polished and effective show on a topical subject. In figure 12 a screen capture of the timeline editor is shown.

Turning Uniview into a production tool, however, involves a lot more than just timeline editing. The producer tool also

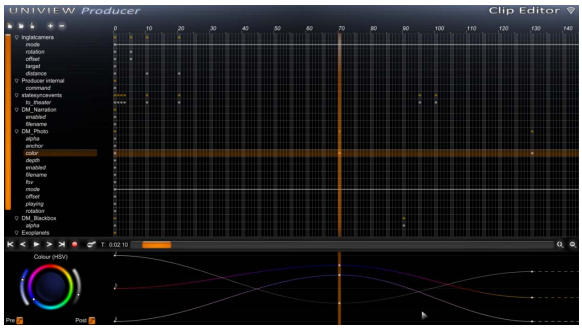

Figure 12: *A screen capture of the Uniview producer graphical user interface*

supports rendering output, to accommodate features like 16 bit colour output, sub-frame motion blur and some level of interoperability with standard tools such as Maya. thus video recordings of pre-defined shows can easily be produced and these elements incorporated into more elaborate renderings.

3.6. Remote Presentation

Another feature request that has come from many in the Uniview user community, and which has been a part of the development agenda, is remote and multi-user presentation capabilities. The Uniview remote presentation tool, known as *Octopus*, is based on the exchange of render messages containing navigation and state information between peers, and interpolation of render messages to avoid network spikes.

The remote collaboration feature is included in the 1.3 release of Uniview Theater and has proven to be an extremely powerful feature for communicating science from one part of the world to another, from domes to classrooms, and from presenters wherever they may be on-line. Octopus allows an expert anywhere in the world to become a part of the exhibition of astrophysics and astronomical information to an audience anywhere else on the global internet. They can use a Uniview instance on their own computer, even a relatively low-powered laptop machine where the graphics performance may mean that their own display is of quite low quality, to control a fully-fledged multi-node, multi-channel dome or theatre experience with superlative graphics performance. Uniview regards the two instances as completely symmetric.There is currently a network of Octopus users running several shows each week and it is estimated that several tens of thousands of individuals, both adults and school children, have already experienced the universe through Uniview and an Octopus connection.

4. Summary and Conclusions

Uniview is the manifestation of work done to address the needs of a particular institution, by a team of software developers who have listened very carefully to their users and

who have tried to collaborate with the leading institutions in the field. By deploying student thesis projects focusing on particular areas of scientific visualization and coherently incorporating their results into the software package, we have been able to carefully capture the vision and intentions of our user base. The seven feature areas described in this paper are only some of the major efforts undertaken, and even these seven do steer in dramatically different directions.

The future development of Uniview will take place in close collaboration with the key partners and continue to address the important challenges in interactive visualization for large scale theatres, such as the mix of scripting and free navigation. We will also explore the embedding of simulations into the observed data. On the rendering side we have already begun to explore large scale volumetric rendering of astrophysical phenomena, which has the capability to provide excellent results, merging the results seamlessly into the Uniview scene graph as objects which can be manipulated and navigated around as easily as the present geometric representations.

References

[Ame] AMERICAN MUSEUM OF NATURAL HISTORY: Digital universe. http://www.haydenplanetarium.org/universe/.

[Duc] DUCHAINEAU M.: ROAM 2.0. http://www.cognigraph.com/ROAM_homepage/.

[FH07] FU C.-W., HANSON A. J.: A transparently scalable visualization architecture for exploring the universe. *IEEE Transactions on Visualization and Computer Graphics 13*, 1 (January/February 2007), 108 – 121.

[Jeta] JET PROPULSION LABORATORY: Onearth. http://onearth.jpl.nasa.gov/.

[Jetb] JET PROPULSION LABORATORY: The Planetary Science Division's Ancilliary Information System. http://naif.jpl.nasa.gov/naif/.

[Lev03] LEVY S.: Interactive 3d visualization of particle systems with partiview. *IAU Symposium 208* (2003), 343–348.

[Nat] NATIONAL CENTER FOR SUPERCOMPUTING APPLICATIONS: Partiview. http://dart.ncsa.uiuc.edu/partiview/.

[O'N] O'NEIL S.: Atmospheric Scattering. http://www.gamedev.net/columns/hardcore/atmscattering/.

[Sol] SOLAR SYSTEM SIMULATOR PROJECT: Solar system simulator. http://www.sssim.com/.

[The] THE OPEN GEOSPATIAL CONSORTIUM: WMS and KLM. http://www.opengeospatial.org/standards/wms, http://www.opengeospatial.org/standards/kml.