Very Large Scale Visualization Methods for Astrophysical Data

Andrew J. Hanson, Chi-Wing Fu, and Eric A. Wernert

Indiana University, Bloomington, IN 47405, USA,
{hanson, cwfu, ewernert}@cs.indiana.edu,
WWW home page: http://www.cs.indiana.edu/

Abstract. We address the problem of interacting with scenes that contain a very large range of scales. Computer graphics environments normally deal with only a limited range of orders of magnitude before numerical error and other anomalies begin to be apparent, and the effects vary widely from environment to environment. Applications such as astrophysics, where a single scene could in principle contain visible objects from the subatomic scale to the intergalactic scale, provide a good proving ground for the multiple scale problem. In this context, we examine methods for interacting continuously with simultaneously active astronomical data sets ranging over 40 or more orders of magnitude. Our approach relies on utilizing a single scale of order 1.0 for the definition of all data sets. Where a single object, like a planet or a galaxy, may require moving in neighborhoods of vastly different scales, we employ multiple scale representations for the single object; normally, these are sparse in all but a few neighborhoods. By keying the changes of scale to the pixel size, we can restrict all data set scaling to roughly four orders of magnitude. Navigation problems are solved by designing constraint spaces that adjust properly to the large scale changes, keeping navigation sensitivity at a relatively constant speed in the user's screen space.

1 Introduction

We study the problem of supporting interactive graphics exploration of datasets spanning dozens of orders of magnitude in space and time.

The physical universe is precisely such a data set, and, as our theoretical and experimental understanding of physics has improved over the years, the development of appropriate techniques for making these vast scales accessible to human comprehension has become increasingly important. From the scale of galactic super-clusters down to the quarks in atomic nuclei, we have approximate sizes ranging from 10^{25} meters down to 10^{-15} meters. To look closely at cosmological scales corresponding to the distance that far-away light has traveled since near the beginning of the Big Bang [10], we may need to go to even larger scales, while to visualize the tightly wound Calabi-Yau spaces intrinsic to the "hidden dimensions" of modern string theory [6, 9], we need to delve down to the Planck length of 1.6×10^{-35} meters, giving a potential requirement for 60 or more orders of magnitude.

Large scale data sets are common in visualization, but the focus of the visualization literature has generally been on data sets with massive amounts of information (see, e.g., [13]). As increasingly detailed data have become available on distant astronomical structures (see, e.g., [2, 5]), the combined problems of large scale in the size and large scale in the extent of astronomical data have received increasing attention (see, e.g, [16]).

The concept of creating pictures and animations to expose these vast realms has a long history. As early as 1957, a small book by Kees Boeke entitled “Cosmic View: The Universe in Forty Jumps” [1] had already appeared with the purpose of teaching school children about the scales of the universe. The effort to turn the ideas in this book into a film were pursued by the creative team of Charles and Ray Eames for over a decade, starting with a rough draft completed in 1968 and culminating in the classic film “Powers of 10” in 1977 [3]. Five years later, Philip and Phylis Morrison annotated the movie in a richly illustrated Scientific American book [12]. The Eames Office continues to develop educational materials based on the original film, including a CD-ROM [14] that adds many additional details. Recently, the IMAX corporation sponsored the production of “Cosmic Voyage,” a feature-length film [15] that reframes many of the features of “Powers of 10” using modern cosmological concepts combined with current large scale 3D data sets and simulations to provide additional richness and detail.

What all these efforts have in common is that they are relatively static, with no easy way to let the student of the universe ask his or her own questions and wander through the data choosing personal viewpoints. Even given the general availability of many interesting astronomical data sets and the computer power to display portions of them in real time, myriad problems confront the visualizer who wishes to roam freely through the whole range of scales of the physical universe. In this paper we present an elegant approach to solving this problem.

2 Strategies

Our experience with brute force attempts to display data sets of many orders of magnitude is that major flaws in precision occur between scales of 10^{13} and 10^{19}; the problems can actually be worse on more expensive workstations, regardless of the apparent uniformity of the promised hardware support for OpenGL transformation matrices. One can surmise that the onset of problems traces to normalizing vectors by taking the inverse square root of the scale squared; with an available exponent range in single precision arithmetic up to about 10^{38}, our experimental observations are theoretically plausible.

One of our chosen tasks is therefore to find a way that would allow many more orders of magnitude to be represented without significant arithmetic errors. We accomplish this with three major strategies: (1) the replacement of ordinary homogeneous coordinates by “log scale” homogeneous coordinates, with the fourth coordinate representing the $\log$ of the current scale in the chosen base, typically 10; (2) the use of a small number of cycling scale frames, covering approximately three orders of magnitude, that are reused so that all data are rendered at or near unit scale, regardless of the actual scale; (3) pixel-level replacement of enormous but distant visible data sets by hierarchically scaled models reducing to environment texture maps whenever the current scale of local user motions would result in negligible screen motions of distant objects.

We then combine these techniques with a scale-intelligent implementation of the constrained navigation framework [7, 17] to keep user navigation response constant in screen space; this is essential to retain an appropriate level of user response universally across scales. A crucial side benefit of constrained navigation is somewhat similar to the advantages of carefully constraining the camera motion in Image-Based Rendering: one allows the viewer a great deal of freedom, but restricts access in such a way that the expense of data retrieval is reduced compared to that needed for full six-degree-of-freedom "flying."

In the following sections, we begin our work by deriving the conditions needed to represent a large range of scales in spatial data. Next, we describe our hierarchical approach to object representation, ways to support many scale magnitudes within a single object, and the concept of variable-representation geometry adapted from classical cartography. Finally, we treat several examples in the context of constrained navigation with intelligent scaling built into the user interface.

3 The Geometry of Scale Uniformization

Navigation through a virtual environment and the placement of objects in the environment utilize the standard six degrees of freedom: three orientation parameters, which are independent of scale, and three position parameters, which must be adapted to our scaling requirements. In order to support local scales near unity, we introduce one additional parameter, the power scale, which is effectively an index into an array of local, mipmap-like unit-scale environments whose scales are related by exponentiating the indices.

3.1 Homogeneous Power Coordinates

We define the homogeneous power coordinate representation of a point as

$$p = (x, y, z, s) \tag{1}$$

where s is typically an integer and the physical coordinates in 3D space corresponding to p are

$$\mathbf{X}(p) = (xk^s, yk^s, zk^s) = \mathbf{x} \times k^s .$$

Here k is the chosen scale base, typically $k = 10$. The homogeneous power coordinates p are thus equivalent to the ordinary homogeneous coordinates $X = (x, y, z, w)$ with

$$s = -\log_k w \ .$$

Thus if we choose $k = 10$ and units of meters, we see that $s = 0$ corresponds to the human scale of one meter, and $s = 7$ to about the diameter of the Earth (1.3×10^7m). Table 1 gives a rough picture of some typical scales in these units.

We note that s can be generalized to a vector, $\mathbf{s} = (s_x, s_y, s_z)$, such that s_x, s_y and s_z are the power scales applied to the x, y and z-components of p independently.

Object	Power of 10	Object	Power	Object	Power	Object	Power
Planck length	-35	virus	-7	Earth	7	Local galaxies	23
proton	-14	cell	-5	Solar system	14	Super-cluster	25
hydrogen atom	-10	human	0	Milky Way	21	Known Universe	27

Table 1. Base 10 logarithms of scales of typical objects in the physical universe in units of meters. To convert to other common units, note that 1 *au* = 1.50×10^{11} m, 1 *ly* = 9.46×10^{15} m, and 1 *pc* = 3.26 *ly* = 3.08×10^{16} m, where *au* = astronomical unit, *ly* = light year, and *pc* = parsec.

3.2 Homogeneous Power Coordinate Interpolation

Since each coordinate $p = (x, y, z, s)$ may have a different power scale, our first requirement is the formulation of interpolation methods that may be used to transition smoothly among scales. To interpolate between two homogeneous power coordinates p_0 and p_1 using the parameter t, where $t \in [0, 1]$, we may immediately write $p(t) = (1-t)p_0 + tp_1$. Since $\mathbf{X}(p) = \mathbf{x} \times k^s$ is the ordinary space representation of p, the interpolation in ordinary space becomes

$$\mathbf{X}(p(t)) = [(1-t)\mathbf{x}_0 + t\mathbf{x}_1] \times k^{(1-t)s_0 + ts_1} \ .$$

When $s_0 = s_1$, this reduces to ordinary interpolation within a single scale domain. When we fix $\mathbf{x}_0 = \mathbf{x}_1$, we can implement a “Powers of [Base k]” journey by interpolating directly in power-scale space alone.

In order to adjust display positions of objects with different scales appearing in the same scene, we need scale-to-scale correspondence rules. Thus, if we let p_0 and p_1 be the positions of a single object expressed with respect to power scales of s_0 and s_1, respectively, we require equivalence of the physical coordinates:

$$\begin{aligned} \mathbf{X}_0(p_0) &= \mathbf{X}_1(p_1) \\ \mathbf{x}_0 \times 10^{s_0} &= \mathbf{x}_1 \times 10^{s_1} \ . \end{aligned}$$

That is, $\mathbf{x}_1 = \mathbf{x}_0 \times 10^{\delta s}$, where $\delta s = s_1 - s_0$ is the change in the power scale.

3.3 Environment Map Angular Scale Criteria

Next, we need to set up the framework for viewing large scale objects that intrude upon the current scale of the user's display.

Let $\mathbf{x}_1$ and $\mathbf{x}_2$ be the limiting boundaries of a family of $3D$ observation positions spanning a distance $|\mathbf{x}_2 - \mathbf{x}_1| = d$, and let O be an object at a radial distance r from the mid-point of the line joining $\mathbf{x}_1$ and $\mathbf{x}_2$, as shown in Figure 1. As we move along the line between $\mathbf{x}_1$ and $\mathbf{x}_2$, O will have the greatest angular displacement (as perceived by the viewer) when O is located on the perpendicular bisecting plane of $\mathbf{x}_1$ and $\mathbf{x}_2$. Thus if α is the maximal angular displacement of O with respect to a motion connecting $\mathbf{x}_1$ and $\mathbf{x}_2$, then $\tan(\alpha/2) = d/2r$, so

$$\alpha = 2 \tan^{-1} \frac{d}{2r} \ . \tag{2}$$

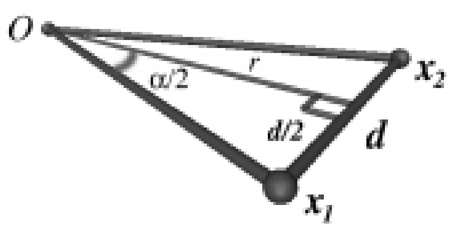

Fig. 1. Geometry for the calculation of an object's maximum subtended angle α.

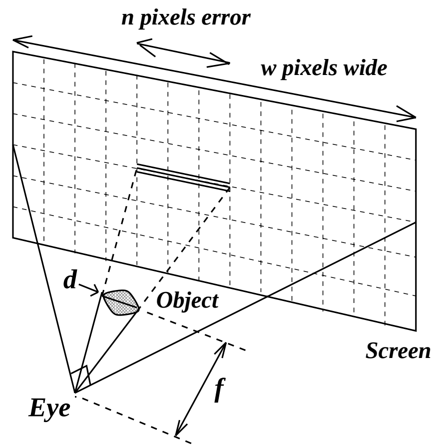

Fig. 2. Schematic diagram for angular limits deciding whether to omit an object.

Next let w be the maximum resolution of a desktop display screen, or perhaps one wall of a CAVE$^{\mathrm{TM}}$. Assuming a 90-degree field of view, the typical angle subtended by one pixel is $\pi/2w$. We impose the assumption that whenever an object is displaced by fewer than n pixels as we move our viewpoint across its maximal local range, we can represent that object by a static texture map (an *environment map*) without any major perceptual error. The condition becomes

$$\alpha < \frac{n\pi}{2w} \ .$$

Thus we conclude that $\tan^{-1}(d/2r) < n\pi/4w$, so

$$r > \frac{d}{2\tan(n\pi/4w)} \ . \tag{3}$$

Thus we may replace an object O by its pixel-level environment map representation whenever its distance r from the worst-case head-motion displacement axis with $d = |\mathbf{x}_2 - \mathbf{x}_1|$ satisfies Eq. (3).

Local User Motion Scales. Let s be the scale of our current environment. This means, e.g., that we typically restrict local motions to lie within a virtual cube of side 10^s accessible to the user without a scale change. Assuming the average extreme motion is given simply by the cube edge length, and letting the distance of the object in question be of order $r = 10^l$, we find

$$10^l = (\frac{10^s}{2})\frac{1}{\tan(\frac{n\pi}{4w})}$$
$$\delta s(n) = l - s = -\log_{10} 2 - \log_{10} \tan(\frac{n\pi}{4w}) \ .$$

Plugging in typical values for n and taking $w = 1024$, we find the following results:

$$\delta s(1) = 2.81, \quad \delta s(2) = 2.5, \quad \delta s(3) = 2.34, \quad \delta s(4) = 2.21 \ .$$

Therefore, to represent the Milky Way with individual star data from the Bright Star (B-SC) catalog [8] or Hipparcos [4], we need a hierarchical strategy: whenever the current scale differs from that of the star data scale by about 3 units, we can just use an environment map. As we move within the scale of a single local environment, this guarantees that no object position will differ by more than n pixels from the full 3D rendering.

3.4 Object Disappearance Criteria

When will an object's scale be so small that we do not need to draw it? In this section, we derive the estimates needed to decide when to "recycle" a particular scale representation so that we can adhere to our requirements that nothing be represented too far from unit scale in *actual* calls to the graphics hardware. This of course can only be accomplished if the transition between successive scales is smooth and has the appearance of a large scale change. To carry out this illusion, we need to eliminate small scale representations in favor of new unit scale representations as the camera zooms out (and also the reverse).

The Smallest Visible Object. We begin by assuming that any object subtending less than n pixels when projected to the screen or CAVE wall can be ignored and does not need to be drawn.

As before, the resolution of the screen is described by the average angle subtended by one pixel, i.e., $\pi/2w$, where w be the maximum screen resolution in pixels; thus, if the projected size of anything is smaller than $n\pi/2w$, we can ignore it.

Largest Angle Subtended by an Object. Given an object of size d, we need to calculate the worst-case projected size on the screen in order to decide when it can be omitted.

If f is the distance from the viewpoint to the near viewing plane, the largest angle that could be subtended by this object is d/f.

Note that the limits of f depend on the human visual system: when the object is too near, the eye cannot focus on it — the near limit of f is roughly 0.3 feet (10 cm). An estimate of the required precision for a CAVE follows from the CAVE wall dimension of 8–10 feet (2.4–3.0 m), so f is about $1/24$ to $1/30$ the screen size.

Next, we have to convert f to the units of our virtual environment. When the current scale of the virtual environment is s, one side of the wall is 10^s units. For example, an 8-foot (2.4 m) CAVE would have f roughly equal to $10^s/24$ in wall units.

Combining the Numbers. We can now put everything together. When the visual angle of an object is smaller than the visual limit, it does not need to be drawn. The condition is that the angle in radians subtended by the object at distance f must be less than the angle subtended by the desired limit of n screen pixels (see Figure 2):

$$\frac{d}{f} = \frac{24d}{10^s} < \frac{n\pi}{2w}$$

Defining $l = \log_{10} d$, we find $24(10^{l-s}) < n\pi/2w$, so that

$$\delta s(n) = l - s < \log_{10}\left(\frac{n\pi}{48w}\right)$$

Note that $n\pi/48w$ is smaller than 1 in general, so δs is normally negative. Plugging in typical values for n and letting $w = 1024$, we find:

$$\delta s(1) = -4.194, \quad \delta s(2) = -3.893, \quad \delta s(3) = -3.717, \quad \delta s(4) = -3.592\,.$$

Therefore, if the scale of an object is smaller than the current scale by around 4.2, we can ignore it for rendering purposes without entailing more than a single pixel worth of perceptual error.

4 Implementing Multi-scale Blending

The conceptual goal of keeping all objects drawn at unit scale as the graphical viewing volume sweeps through each object's natural scale is an appealing ideal. However, in practice, very few objects live "alone" in their assigned neighborhood of the order-of-magnitude scale space. From the Earth, at scale 10^7 m, we can see elements of the solar system, at scale 10^{13} m, as well as clouds of stars in our own galaxy, the Milky Way, at scale 10^{21} m. If we wish to show the Milky Way as seen from Earth, we must have *multi-scale* representations of objects at far different scales that can be rationally displayed alongside the Earth without requiring huge scale factors.

In this section, we describe several different issues, problems, and solutions to the mixed scale rendering problem. Our actual implementation involves an elaborate scripting language that supports each of the features described below. In particular, we provide definitions for multiple navigation manifolds, lists of alternate object representations indexed by scale, and animation parameters for hierarchical object motion. Further details will be omitted here for lack of space.

Symbolic interpolators: In many circumstances, certain objects serve as anchors or flags that define a special user context in the visualization. In such circumstances, it is appropriate to render objects at scales that are unconnected to their actual size, but are dictated by their semantic importance. In ordinary cartography, landmarks or objects of legal significance or liability to the map maker are drawn using symbols out of proportion to the conventional size of an object. One example we have implemented keeps the Earth itself visible by using a constant size globe starting above some particular scale threshold.

A related technique (see, e.g., [11]) maintains an entire virtual library of depictions to be used for an object in different scale environments. At one scale, a fully rendered, illuminated 3D object may be appropriate, at another a textured 2D billboard may be correct, while at another one might use fixed-scale text if the object should never completely disappear.

Multi-resolution data structures and octrees: Typical astronomical objects may extend over many orders of magnitude, with a variety of natural representation scales for individual subregions. Such data are normally stored in a hierarchy of resolutions such as an octree. However, to have a fully detailed set of data at all possible points of an enormous astronomical object would be prohibitively expensive. We can sidestep this problem and make the missing detail less obvious to the user by restricting the navigation paths and permitted view directions to permit detailed examination of only certain selected areas. Thus one can in principle design an environment in which the highest levels of detail are typically be limited to only a few regions of the octree representation, permitting substantial economy in data storage. If all the detail is actually required, more sophisticated methods such as those suggested in [16] are needed.

Environment maps: We have already derived a series of formulas determining the level at which a 3D object can be replaced by an asymptotic texture map without requiring an expensive 3D rendering. Such maps are equivalent to so-called "environment maps" that are used to represent distant fixed objects, such as the fixed stars themselves, without rendering them. A spherical texture map, or a series of six orthogonal images with fields of view corresponding to the faces of a cube, will accomplish the desired result. All we need to do is to keep track of the camera motion so we can change the environment map (or switch to 3D detail) if the viewpoint changes significantly.

5 Examples of Scaled Constrained Navigation

We define constrained navigation to be the assignment of a mapping between a controller space and a general field of viewing control parameters [7]. Our typical implementation involves the design of one or more 2D "sidewalks" that define the 3D spatial motion of the user and the view parameters in response to inputs in a limited controller space such as that of a mouse or the CAVE thumb joystick.

Since the response of the user displacement to a unit of controller motion can be completely controlled and customized by the fields stored with the navigation manifold, we can easily adapt the motion response to meet the user requirements of large-scale navigation. In particular, it seems obvious that a "Powers of [Base k]," i.e., logarithmic, scaling of the response is the natural one to use: the farther we get from the Earth's surface, the lower the density of detailed observational data, and the larger the scale of the visible structures that are interesting to depict.

In addition to scaling motion control with the constrained navigation framework, we have studied a number of constraint manifold designs that are well-suited to this work: among these, we describe below the "pond ripple," the "wedge," and the "twist."

Multi-centered Pond Ripple Navigation. In Figure 3, we show a very special disk-shaped navigation manifold. This manifold has not one, but multiple centers of attention, corresponding to the evolving changes in the centers of the rings at each scale level. Tangential motion takes place in the lateral ring direction, centering the viewer's gaze on the current center.

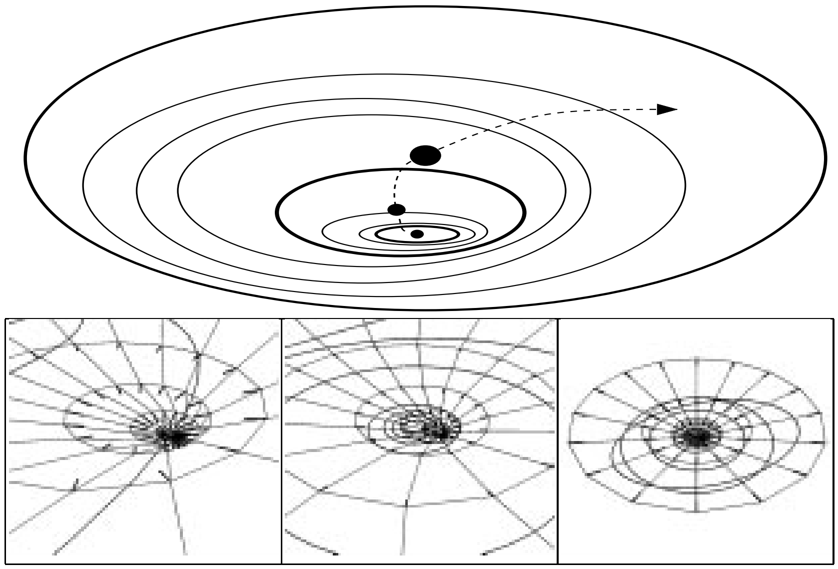

Fig. 3. "Pond ripple" navigation environment with continuous recentering as the viewer orbits around the central object of interest at each scale. The bottom row of diagrams illustrates the fact that the viewpoint directions (short heavy bars) at three different positions focus on three different centers.

Wedge Exponential Journey. In Figure 4, we show a symbolic journey on a wedge-shaped navigation manifold; the width of the wedge as well as the actual distance traveled per unit controller motion expand exponentially with distance from the origin. This allows the viewer to pursue a "Powers of [Base k]" interactive exploration of the space, and to travel between viewpoints in logarithmic time instead of the much less appropriate constant velocity.

Twisting Reorientation. In Figure 5, we show the advantage of the constrained navigation approach for huge scales with smooth evolution between mismatched orientation frames natural to each scale. Beginning with a "wedge" manifold, we twist the frames to get custom orientations. Beginning with an orientation suitable for viewing the Earth, we move out seven orders of magnitude, while twisting so the solar system appears horizontal instead of the Earth's equator; moving out seven more orders of magnitude to the galactic scale, we twist again to orient ourselves to the galactic plane.

6 Conclusions

We have addressed the problem of effective interactive navigation across huge ranges of scales appropriate for spanning the entire physical universe. This was accomplished with a combination of methods, including the following: a systematic treatment of all data at unit scale with (hardware) transformation matrices never exceeding four orders of magnitude in scale range; systematic blending of both iconic (fixed symbolic scale) and very large data groups using multi-scale representations; merging environment maps with mipmaps and octrees to make very distant data visible without violating the unit scale requirement; supporting multi-resolution maps to allow switching to layered unit-scale data representations as we zoom in and out; scaling both spatial navigation control and time-scale of simulations to the scale of the current local data representation; constrained multi-resolution navigation. Future work will focus on additional requirements of time scaling, the multi-resolution problem with very large data sets, techniques for designing navigation constraints to control the required data set sizes, analyzing user responses to the environment, and incorporating the effects of special and general relativity in ways that are intuitive and qualitatively correct without being obtrusive.

Acknowledgments

This research was supported by NASA grant number NAG5-8163. This research was also made possible in part by NSF infrastructure grant CDA 93-03189 and the generosity of the Indiana University Advanced Visualization Laboratory. We thank P.C. Frisch for many useful interactions and contributions.

References

1. Kees Boeke. *Cosmic View: The Universe in Forty Jumps*. John Day, 1957.
2. Arthur N. Cox. *Astrophysical Quantities*. Springer-Verlag New York, Inc., New York, 1999.
3. Charles Eames and Ray Eames. Powers of Ten, 1977. 9 1/2 minute film, made for IBM.
4. M. A. C. Perryman et al., editor. *The Hipparcos and Tycho Catalogues*. ESA Publications Division, The Netherlands, 1997.
5. M. Geller and J. Huchra. Mapping the universe. *Science*, 246:897–910, 1989.
6. Brian Greene. *The Elegant Universe: Superstrings, Hidden Dimensions, and the Quest for the Ultimate Theory*. W.W. Norton, 1999.
7. A. J. Hanson and E. Wernert. Constrained 3D navigation with 2D controllers. In *Proceedings of Visualization '97*, pages 175–182. IEEE Computer Society Press, 1997.
8. Dorrit Hoffleit and Carlos Jaschek. *The Bright Star Catalogue: Fourth Revised Edition*. Yale University Observatory, New Haven, 1982.
9. Michio Kaku. *Introduction to Superstrings and M-Theory*. Springer Verlag, 1999.
10. David Layzer. *Constructing the Universe*. Scientific American Books, 1984.
11. Paulo W. C. Maciel and Peter Shirley. Visual navigation of large environments using textured clusters. *1995 Symposium on Interactive 3D Graphics*, pages 95–102, April 1995.
12. Philip Morrison and Phylis Morrison. *Powers of Ten*. Scientific American Books, 1982.
13. G.M. Nielson, H. Hagen, and H. Mueller. *Scientific Visualization: Overviews, Methodologies, Techniques*. IEEE Computer Society, 1997.
14. The Office of Charles and Ray Eames. Powers of Ten, 1998. CD-ROM educational materials.
15. Bayley Silleck. Cosmic voyage, 1996. 35 minute film, a presentation of the Smithsonian Institution's National Air and Space Museum and the Motorola Foundation.
16. D. Song and M. Norman. Looking in, looking out: Exploring multi-scale data with virtual reality. *Comput. Sci. Eng.*, 1(3):53–64, 1994.
17. E.A. Wernert and A.J. Hanson. A framework for assisted exploration with collaboration. In *Proceedings of Visualization '99*, pages 241–248. IEEE Computer Society Press, 1999.

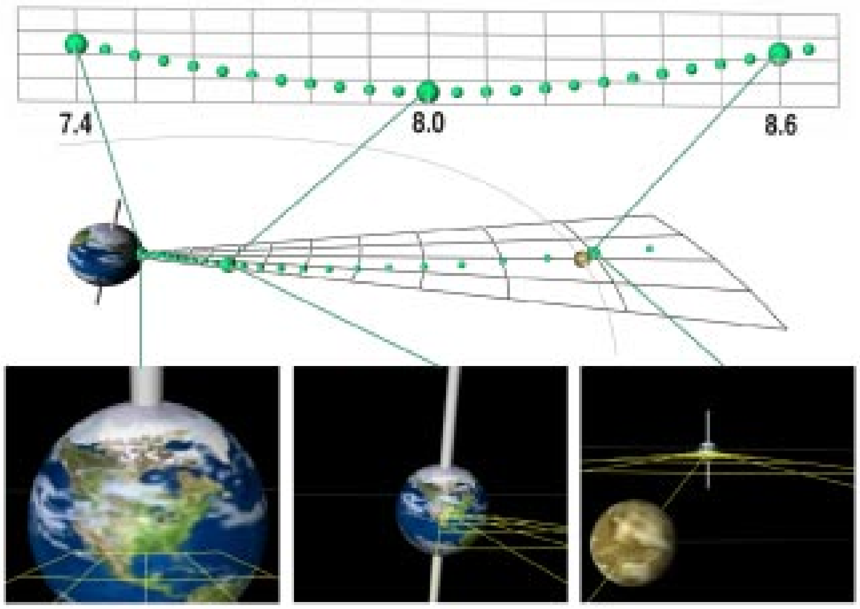

Fig. 4. Images corresponding to three viewpoints on the logarithmically scaled “wedge” path from the Earth to the Moon.

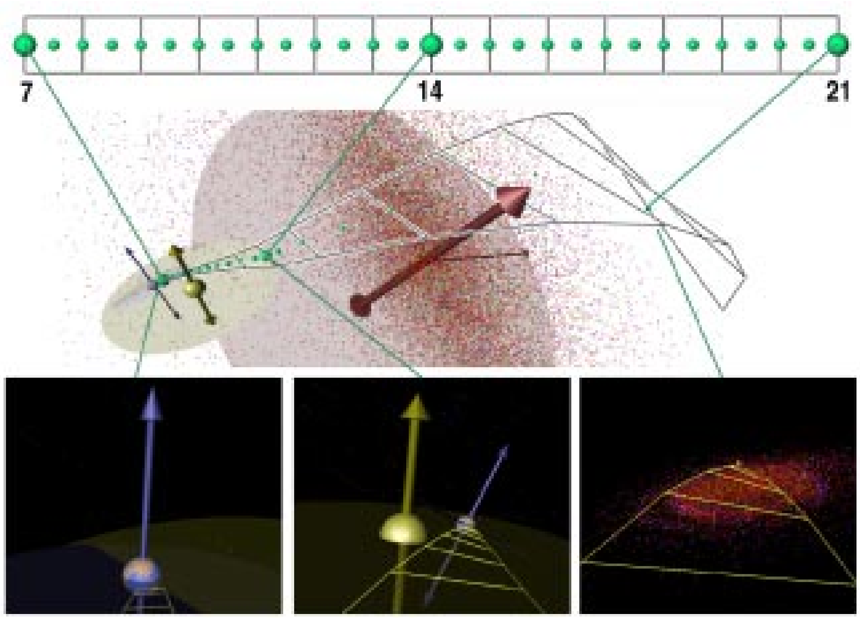

Fig. 5. Images corresponding to three viewpoints on the adaptive twisted-orientation constraint manifold encompassing the Earth, solar system, and the Milky Way galaxy.