

# Animating Observation Geometries with Amphion

Steven Roach, University of Wyoming/Ames summer graduate student, Michael Lowry and Thomas Pressburger, Amphion Project, IC Division, Ames Research Center

Science opportunity visualizers (SOVs) are essential tools for analyzing mission designs, planning detailed science observations, and aiding the analysis of science data returned from a spacecraft's instruments. In the past, creating an SOV was a labor intensive software development process. This article describes an extension to the Amphion automatic programming system that facilitates the rapid generation of SOVs for understanding observation geometries. Although this extension is not yet mature, it has already been employed in developing a SOV that is actively being used by the planetary rings community, as well as dozens of prototype animation programs.

This article first reviews SPICE and Amphion, and then discusses the extensions that are used in generating SOVs.

## SPICE and Amphion

Jet Propulsion Laboratory's (JPL) Navigation and Ancillary Information Facility (NAIF)

Group constructed the SPICE ancillary information system to assist in planning observations and interpreting data from space borne instruments. It includes data sets for solar system bodies (e.g., planets, satellites, and spacecraft) and software in the form of a FORTRAN-77 subroutine library. The subroutines can be used to write programs that, by accessing SPICE data sets, calculate observation geometry parameters. A recent overview of SPICE was presented in the December 1994 issue of this newsletter on pp. 29-30.

Amphion is an automatic program synthesis system; that is, a system that writes source code programs given high-level problem specifications. Amphion was introduced in the February 1994 issue, pp. 22-25. It consists of a graphical user interface and a program synthesis subsystem. Amphion allows you to program at the level of abstract problem specifications instead of at the tedious level of programming language syntax. It is a general program synthesis system that is tailored to a particular domain by a domain



## Navigation Ancillary Information Facility

The goal of the Navigation and Ancillary Information Facility is to provide the planetary science community with data sets and transportable software tools, appropriate for computing, archiving, accessing and distributing the ancillary viewing geometry needed to interpret observations of solar system bodies.

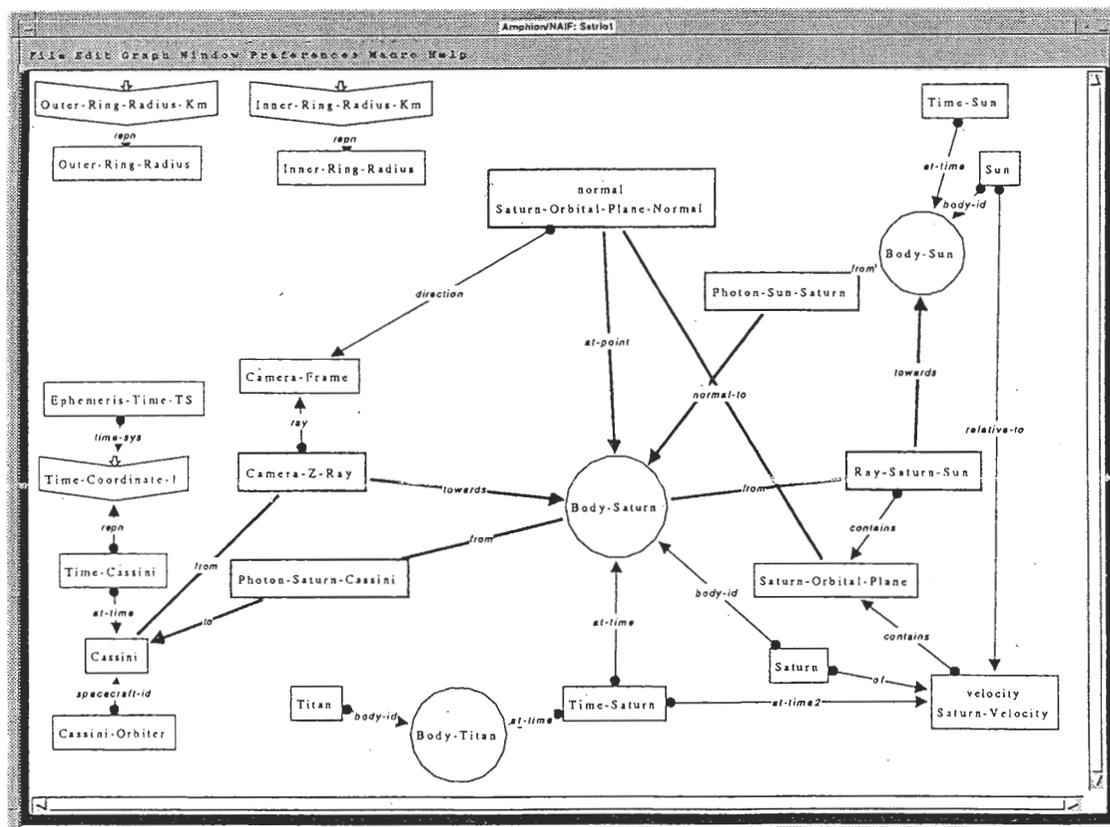


Figure 1. Graphical specification of Cassini observation geometry



The specification in Figure 1 includes Saturn, Titan, the Sun, and the Cassini spacecraft. In this specification, a "body" is the location of some object at a particular time. So the "Body-Saturn" icon is the location of the planet Saturn at Time-Saturn. The inputs to the program are the time of the observation at Cassini (Time-Cassini) and the inner and outer diameter of one of Saturn's rings to be displayed. To allow for the time light takes to travel, the time of Body-Saturn is the time a photon would have left Saturn to arrive at Cassini at Time-Cassini. The time of Body-Sun is specified in a similar fashion. Since they are relatively close, we assume that the time of Body-Titan is the same as that of Body-Saturn.

To create a visualization, a camera must also be specified. A camera has a coordinate system called a frame. The z-axis of the frame points along the camera boresight toward the objects being viewed, the x-axis points to the right, and the y-axis points up. In Figure 2, the z-axis is defined as a ray from Cassini toward Body-Saturn. The y-axis is orthogonal to the x-axis and is aligned with the vector normal to the orbital plane. The orbital plane is defined as the plane that contains two vectors: the instantaneous velocity vector of Saturn, and a vector from Saturn to the Sun.

In Figure 2 the specifications have been added for the objects to appear in the visualization and for the location of the source of light. This is done by creating display-objects. Each display-object consists of two components: an object in the specification and the camera from which it is

viewed. In Figure 2 there are display-objects for Body-Saturn, Body-Titan, and the inner and outer edges of the ring. Rings are visualized as pairs of simple ellipses depicting the inner and outer edges. The illumination source is Sun-Body.

From the specification in Figure 2 Amphion generates a FORTRAN program containing the appropriate calls to the SPICE and Euclid subroutines. These in turn call ESCHER and X-windows subroutines that open a window on a terminal and display a visualization. The program can automatically generate visualizations for a sequence of times, thus producing an animation. In contrast to many other animation programs, the program is efficient enough to generate the sequence of views in real time. Thus the entire Cassini four year tour is animated at any desired time compression without storing a buffer of images. Figure 3 is a frame from this animation at the time of the first Titan fly-by.

#### *Animation and ring crossing visualizer*

A fragment of the Cassini animation was extracted to create an MPEG file at the University of Wyoming. This MPEG file is a time-compressed animation of the first Cassini fly-by of Titan; it is about a minute long. This fragment can be viewed on Amphion's World Wide Web (WWW) Home Page at:

<http://ic-www.arc.nasa.gov/ic/projects/amphion>  
or downloaded through anonymous ftp from:

[ai.uwyo.edu/pub/nasa/](http://ai.uwyo.edu/pub/nasa/)

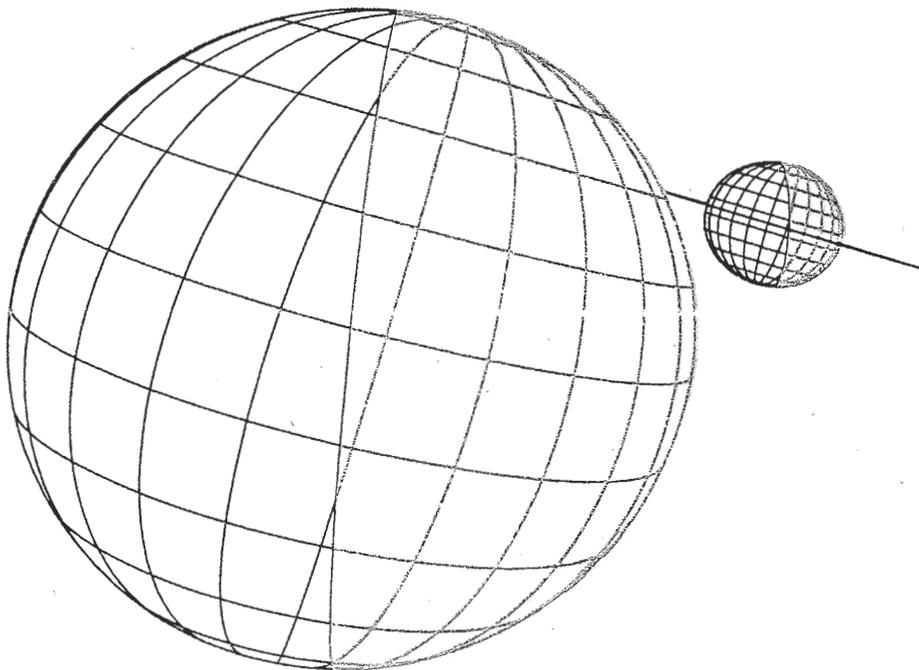
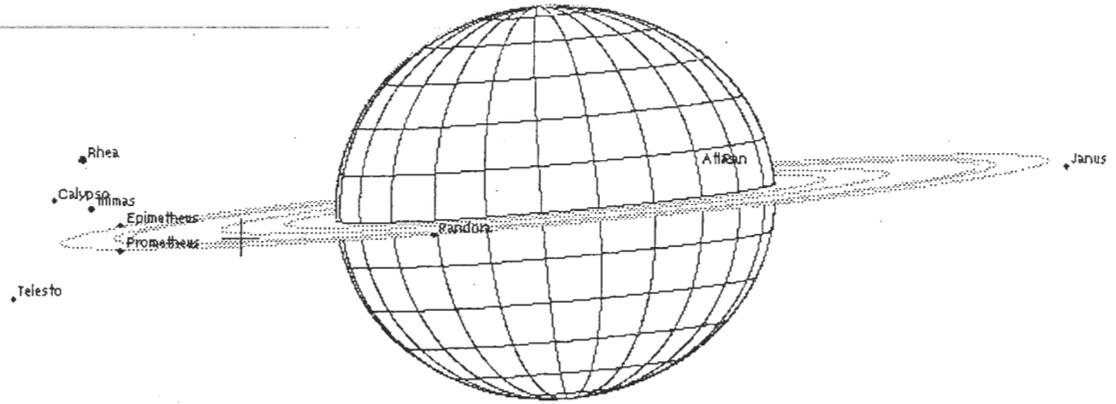


Figure 3. Freeze frame of first Titan fly-by in Cassini tour animation. Animation program was generated by Amphion.

Figure 4. Diagram of Saturn system generated by PDS Rings Node's Saturn Viewer program



The animation can be seen through any MPEG viewer, although some MPEG viewers might not display the colors correctly.

In December 1994 Mark Showalter, manager of the Rings Node for PDS, inquired about using Amphion to generate visualizations of the Saturn system as it will appear from Earth during the 1995-96 ring plane crossings. His objective was to provide a Web-based SOV to help astronomers plan their observations. Mark was able to adapt the specification in Figure 2 very quickly to produce a specification for his SOV. The program generated by Amphion was then further modified by Mark and is now accessible through the WWW at:

<http://ringside.arc.nasa.gov/www/rpx/rpx.html>

Figure 4 was generated using this Saturn Viewer program. It depicts Saturn, its rings, the inner moons, and an occulted background star just a few minutes after the Sun finishes crossing Saturn's ring plane at 12:05 UTC on November 21, 1995. The rings are shown in light gray because we will be seeing their unilluminated side at this time. The locations and names of the inner moons are shown (although the names are printed in a font too small to read in this reproduction). By coincidence, a 12th magnitude background star, shown as a plus, will be occulted by the rings at this time.

#### ***Current status of Amphion visualization capabilities***

Currently, Amphion generates a visualization subroutine containing calls to SPICE and Euclid subprograms. The user must build the main

program that calls this subroutine. Templates of main programs are available, and in practice it has been easy for programmers familiar with FORTRAN to build main programs from the templates. The main program templates have facilities for changing camera parameters at execution time (zoom in and out, pan side to side), and for starting and stopping the animations and stepping forward or backward through time.

#### ***Future plans***

The Amphion group is working to augment the animation domain theory and driver code to support multiple illumination sources, text, and multiple cameras. We plan to enhance the animation component to make it more compliant with the full features of the X-windows system and to make the execution environment more intuitive.

Amphion is also being extended to generate iterative code that searches for solutions to queries such as "When does a geometric configuration occur in a time interval" and "When does a geometric function take on a minimum value in a time interval." The NAIF group developed a set of iterative FORTRAN drivers call PERCY that form the components for this capability. Amphion is being extended to target PERCY, just as it was extended to target the elements of Euclid. Amphion's PERCY capability has already been prototyped. In combination with the animation capabilities, this will provide a basis for rapidly generating science opportunity analyzers.

For further information contact Michael Lowry via phone: 415-604-3369 or e-mail:

[lowry@ptolemy.arc.nasa.gov](mailto:lowry@ptolemy.arc.nasa.gov)