Dspace: Real-time 3D Visualization System for Spacecraft Dynamics Simulation

Marc I. Pomerantz and Abhinandan Jain

Jet Propulsion Laboratory, California Institute of Technology,

4800 Oak Grove Dr., Pasadena, CA 91109

***Abstract*— The multi-mission Dshell simulation framework has formed the basis for high-performance, physics-based simulations for a large variety of space mission simulations including cruise and orbiter spacecraft, Entry, Descent, Landing (EDL) missions as well as for planetary surface rovers. The Dspace interactive, reusable 3D visualization system has been developed to support the diverse visualization needs of such complex real-time simulations. A key aspect of Dspace's design is that it can be configured at runtime to support different simulations with no changes to source code, thereby minimizing the cost and time required to develop accurate and real-time visualizations tailored to these simulations. This paper describes the Dspace object-oriented open architecture and the various challenges it addresses for real-time simulation visualization.**

During a simulation run, the initialization process is responsible for registering it's simulation objects (with associated tags) with Dspace in a scenegraph hierarchy. The scenegraph tags allow the simulation to update the status of the objects in real-time within Dspace's 3D scene. The Dshell simulations manage this efficiently at run-time using watch handlers that trigger only when the state of a simulation object changes within the simulation. For example, for articulated rover vehicle simulations, joint motions for vehicle components such as suspension systems or camera mast parts trigger simulation callbacks that cause Dspace to search for and modify the position, orientation and other properties of scenegraph nodes continuously in real time. To reduce search time, scenegraph nodes are cached in Dspace for fast subsequent retrievals. This process keeps the visualization state in sync with the simulation state at all times.

Because users rely significantly on the visualization to interpret the simulation state, it is essential that the visualization model correspond accurately to the simulation model geometry and kinematics. Keeping graphics models consistent with simulation models can be a challenging and expensive proposition as the vehicle structure and design evolves especially so during the early phases of projects. To mitigate this, Dshell auto-generates an invisible skeletal scenegraph fragment that is faithful to the underlying physical model within the simulation. The 3D graphics parts of the vehicle, such as wheels, masts, arms, etc., are attached to this skeleton to ensure that the simulation and visualization models maintain close correspondence to each other.

Rendered scenes typically contain "real" simulation objects together with additional "annotation" objects (eg. axes) that help users interpret the state of the simulation. The Dspace framework provides ways to arbitrarily organize and group simulation objects to allow run-time selection of the visualization content by selective rendering of the groups. An important use case for selective rendering is that of rover camera modeling, where only physical-based objects, such as terrain, planetary bodies or rover components, are allowed in the rendered image, and ornamental objects such as goal markers, viewing frustum pyramids or coordinate axes need to be excluded.

The Dspace module is also in use to support simulation modeling needs that require graphics hardware acceleration. An important example of this is the generation of synthetic imagery for real-time camera simulation, for machine vision modules (eg. stereo processing). Simulated camera images are required to accurately capture the camera optics non-idealities prescribed by camera calibration parameters. The Dshell camera models simulate camera lens radial and fish-eye distortion in real-time by making use of Dspace's offscreen rendering within the simulation pipeline. Real-time, shader-based, shadows and ephemeris-based lighting enhance the realism for this rendered synthetic imagery.

Because the Dshell simulations for surface and near-surface planetary simulations involve terrain models, a Dspace extension called DspaceTerrain has been developed specifically for terrain visualization. The DspaceTerrain rendering engine can manage and render DEM or mesh based terrain using OpenGL fixed functionality rendering or, for very large datasets, using a GLSL clipmap shader. Terrain surface property information in the form of texture images can be applied and terrain geometry deformation can be performed for vehicle wheel tracks.

Dspace development is ongoing and future plans include the implementation of an advanced lighting model for enhanced scene realism for use in Mars or Lunar surface simulations, an interaction mode that will allow users to directly modify simulation states at runtime and an improved single-frame animation process.

Dspace has been used at the JPL's DARTS Laboratory by the ROAMS, DSENDS, LSOS, ATHLETE, AEROBOT and SOOPS projects and for generating products for JPL outreach.

R Madison, M Pomerantz, and A Jain, "Camera Response Modeling and Verification in ROAMS," i-SAIRAS 2005, Munich, Germany, September 2005.

A. Jain, J. Balaram, J. Cameron, J. Guineau, C. Lim, M. Pomerantz, G. Sohl, "Recent Developments in the ROAMS Planetary Rover Simulation Environment", IEEE 2004 Aerospace Conf., Big Sky, Montana, March 6-13, 2004.

H. Nayar, J. Balaram, J. Cameron, A. Jain, C. Lim, R. Mukherjee, S. Peters, M. Pomerantz, L. Reder, P. Shakkottai, S. Wall, A Lunar Surface Operations Simulator ", Proc. International Conference on Simulation, Modeling and Programming for Autonomous Robots (SIMPAR 2008), Venice Italy, Nov. 3-7, 2008.