

using the 3D landmark positions. Finally, it extracts image patches for each landmark from auto-selected key frames and constructs the landmark database. The landmark database can then be used to estimate future camera poses (and therefore localize a robotic vehicle that may be carrying the cameras) by matching current imagery to landmark database image patches and using the known 3D landmark positions to estimate the current pose.

This work was done by Michael Wolf, Adnan I. Ansar, Shane Brennan, Daniel S. Clouse, and Curtis W. Padgett of Caltech for NASA's Jet Propulsion Laboratory. Further information is contained in a TSP (see page 1).

The software used in this innovation is available for commercial licensing. Please contact Daniel Broderick of the California Institute of Technology at danielb@caltech.edu. Refer to NPO-47845.

➤ **Adaption of G-TAG Software for Validating Touch and Go Asteroid Sample Return Design Methodology**

A software tool is used to demonstrate the feasibility of Touch and Go (TAG) sampling for Asteroid Sample Return missions. TAG is a concept whereby a spacecraft is in contact with the surface of a small body, such as a comet or asteroid, for a few seconds or less before ascending to a safe location away from the small body. Previous work at JPL developed the G-TAG simulation tool, which provides a software environment for fast, multi-body simulations of the TAG event. G-TAG is described in "Multibody Simulation Software Testbed for Small-Body Exploration and Sampling," (NPO-47196) *NASA Tech Briefs*, Vol. 35, No. 11 (November 2011), p.54. This current innovation adapts this tool to a mission that intends to return a sample from the surface of an asteroid.

In order to demonstrate the feasibility of the TAG concept, the new software tool was used to generate extensive simulations that demonstrate the designed spacecraft meets key requirements. These requirements state that contact force and duration must be sufficient to ensure that enough material from the surface is collected in the brushwheel sampler (BWS), and that the spacecraft must survive the contact and must be able to recover and ascend to a safe position, and maintain velocity and orientation after the contact.

This work was done by Lars James C. Blackmore, Behcet Acikmese, and Milan Mandic of Caltech for NASA's Jet Propulsion Laboratory. Further information is contained in a TSP (see page 1).

This software is available for commercial licensing. Please contact Daniel Broderick of the California Institute of Technology at danielb@caltech.edu. Refer to NPO-47193.

➤ **3D Visualization for Phoenix Mars Lander Science Operations**

Planetary surface exploration missions present considerable operational challenges in the form of substantial communication delays, limited communication windows, and limited communication bandwidth. A 3D visualization software was developed and delivered to the 2008 Phoenix Mars Lander (PML) mission. The components of the system include an interactive 3D visualization environment called "Mercator," terrain reconstruction software called the "Ames Stereo Pipeline," and a server providing distributed access to terrain models. The software was successfully utilized during the mission for science analysis, site understanding, and science operations activity planning.

A "terrain server" was implemented that provided distribution of terrain models from a central repository to clients running the Mercator software. The Ames Stereo Pipeline generates accurate, high-resolution, texture-mapped, 3D terrain models from stereo image pairs. These terrain models can then be visualized within the Mercator environment. The central crosscutting goal for these tools is to provide an easy-to-use, high-quality, full-featured visualization environment that enhances the mission science team's ability to develop low-risk productive science activity plans. In addition, for the Mercator and Viz visualization environments, extensibility and adaptability to different missions and application areas are key design goals.

Mercator is a cross-platform, adaptable, extensible, interactive 3D visualization software tool that enables users to manipulate and interrogate a simulated 3D environment. It is implemented in the Java programming language to be compatible with Ensemble, a NASA-developed ground data systems software component framework based on the Eclipse open source platform.

The Mercator User Interface (UI) is

divided into a number of tiles or "elements," presenting control panels and views into the 3D scene. The central UI element is an interactive 3D viewer with site interrogation and analysis capabilities. Each UI element can be repositioned, resized, iconified, or dragged out of the window frame.

In an effort to achieve simple, natural interactions, object-oriented, direct manipulation techniques were chosen where practical, and persistent user interface modes were minimized. For example, to measure distances, the user manipulates a 3D representation of a measuring tool in the scene. There is no explicit mode of measurement, and the user can continue to interact with the 3D environment (e.g., changing the viewpoint) as usual.

This work was done by Laurence Edwards, Leslie Keely, David Lees, and Carol Stoker of Ames Research Center. Further information is contained in a TSP (see page 1). ARC-16434-1

➤ **RxGen General Optical Model Prescription Generator**

RxGen is a prescription generator for JPL's in-house optical modeling software package called MACOS (Modeling and Analysis for Controlled Optical Systems), which is an expert optical analysis software package focusing on modeling optics on dynamic structures, deformable optics, and controlled optics.

The objectives of RxGen are to simplify and automate MACOS prescription generations, reducing errors associated with creating such optical prescriptions, and improving user efficiency without requiring MACOS proficiency. RxGen uses MATLAB (a high-level language and interactive environment developed by MathWorks) as the development and deployment platform, but RxGen can easily be ported to another optical modeling/analysis platform.

Running RxGen within the modeling environment has the huge benefit that variations in optical models can be made an integral part of the modeling state. For instance, optical prescription parameters determined as external functional dependencies, optical variations by controlling the in-/exclusion of optical components like sub-systems, and/or controlling the state of all components.

Combining the mentioned capabilities and flexibilities with RxGen's optical abstraction layer completely elimi-