Software

◢ PixelLearn

PixelLearn is an integrated user-interface computer program for classifying pixels in scientific images. Heretofore, training a machine-learning algorithm to classify pixels in images has been tedious and difficult. PixelLearn provides a graphical user interface that makes it faster and more intuitive, leading to more interactive exploration of image data sets. PixelLearn also provides image-enhancement controls to make it easier to see subtle details in images. PixelLearn opens images or sets of images in a variety of common scientific file formats and enables the user to interact with several supervised or unsupervised machine-learning pixel-classifying algorithms while the user continues to browse through the images. The machine-learning algorithms in PixelLearn use advanced clustering and classification methods that enable accuracy much higher than is achievable by most other software previously available for this purpose. PixelLearn is written in portable C++ and runs natively on computers running Linux, Windows, or Mac OS X.

The program was written by Dominic Mazzoni, Kiri Wagstaff, Benjamin Bornstein, Nghia Tang, and Joseph Roden 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 Karina Edmonds of the California Institute of Technology at (626) 395-2322. Refer to NPO-42082.

◢ New Software for Predicting Charging of Spacecraft

The NASA/Air Force Spacecraft Charging System Analyzer Program (Nascap-2K) is a comprehensive update, revision, and extension of several NASA and Air Force codes for predicting electrical charging of spacecraft. Nascap-2K integrates the capabilities and models included in four independent programs: NASCAP/LEO for low-Earth orbits, NASCAP/GEO for geosynchronous orbits, POLAR for auroral charging in polar orbits, and DynaPAC (Dynamic Plasma Analysis Code) for time-dependent plasma interactions. While each of the earlier codes works well for the range of problems for which it was designed, by today's standards these codes are difficult to learn, cumbersome to use, and overly restrictive in their geometric modeling capabilities. Nascap-2K incorporates these models into a single software package that includes spacecraft surface modeling, spatial gridding, environmental specifications, calculating scripting, and post-processing analysis and visualization. The provided material properties database includes values from earlier programs as well as values from recent measurements. Development of Nascap-2K continues with future capabilities to include interactions with dense plasma such as those produced by electric propulsion.

This work was done by Myron Mandell and Victoria Davis of Science Applications International Corp. (SAIC), and Jeffrey Hilton, formerly of SAIC, for **Marshall Space Flight Center**. *For further information, contact Barbara Gardner at SAIC, Barbara.Gardner@saic.com.*
MFS-31939-1/2056-1

◢ Conversion Between Osculating and Mean Orbital Elements

Osculating/Mean Orbital Element Conversion (C version) (OSMEANC) is a C-language computer program that performs precise conversions between osculating and mean classical orbital elements. OSMEANC can be used for precise design of spacecraft missions and maneuvers and precise calculation of planetary orbits. The program accounts for the full complexity of gravitational fields, including aspherical and third-body effects. In comparison with prior software used for the same purposes, OSMEANC offers greater accuracy in conversion: By virtue of inclusion of high-order gravitational and third-body effects, variations in semimajor axes are calculated to meter-level accuracy. OSMEANC is delivered as a callable shared library. It can be built for any platform with a C compiler. The user interface is via a Python-language wrapper script that can be replaced by the user. OSMEANC is mature and is the product of a significant upgrade from a Fortran version that has been in use since 1991.

This program was written by Joseph Guinn, Min-Kun Chung, and Mark Vincent 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 Karina Edmonds of the California Institute of Technology at (818) 393-2827. Refer to NPO-41115.

◢ Generating a 2D Representation of a Complex Data Structure

A computer program, designed to assist in the development and debugging of other software, generates a two-dimensional (2D) representation of a possibly complex n-dimensional (where n is an integer >2) data structure or abstract rank-n object in that other software. The nature of the 2D representation is such that it can be displayed on a non-graphical output device and distributed by non-graphical means. The purpose served by this representation is to assist the user in visualizing and understanding the complex data structure or arbitrarily dimensioned object. This is the only known program that enables a programmer to map an n-dimensional data structure to a flat 2D space. This program does not depend upon the hardware characteristics of a particular output device, and can be executed on a variety of computers from different manufacturers. It can be distributed in source-code or binary-code form. It requires a Lisp compiler. It has no specific memory requirements and depends upon the other software with which it is used and application programs running in it. This software is implemented as a library that is called by, and becomes folded into, the developmental other software.

This work was done by Mark James 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 Karina Edmonds of the California Institute of Technology at (818) 393-2827. Refer to NPO-42076.