Software

Quantum Entanglement Molecular Absorption Spectrum Simulator

Quantum Entanglement Molecular Absorption Spectrum Simulator (QE-MASS) is a computer program for simulating two-photon molecular-absorption spectroscopy using quantum-entangled photons. More specifically, QE-MASS simulates the molecular absorption of two quantum-entangled photons generated by the spontaneous parametric down-conversion (SPDC) of a fixed-frequency photon from a laser. The two-photon absorption process is modeled via a combination of rovibrational and electronic single-photon transitions, using a wave-function formalism. A two-photon absorption cross section as a function of the entanglement delay time between the two photons is computed, then subjected to a fast Fourier transform to produce an energy spectrum. The program then detects peaks in the Fourier spectrum and displays the energy levels of very short-lived intermediate quantum states (or virtual states) of the molecule. Such virtual states were only previously accessible using ultra-fast (femtosecond) laser systems. However, with the use of a single-frequency continuous wave laser to produce SPDC photons, and QE-MASS program, these short-lived molecular states can now be studied using much simpler laser systems. QE-MASS can also show the dependence of the Fourier spectrum on the tuning range of the entanglement time of any externally introduced optical-path delay time. QE-MASS can be extended to any molecule for which an appropriate spectroscopic database is available. It is a means of performing an *a priori* parametric analysis of entangled-photon spectroscopy for development and implementation of emerging quantum-spectroscopic sensing techniques. QE-MASS is currently implemented using the Mathcad® software package.

This program was written by Quang-Viet Nguyen of **Glenn Research Center** *and Jun Kojima of the National Academy of Sciences. Further information is contained in a TSP (see page 1).*

Inquiries concerning rights for the commercial use of this invention should be addressed to NASA Glenn Research Center, Innovative Partnerships Office, Attn: Steve Fedor, Mail Stop 4–8, 21000 Brookpark Road, Cleveland, Ohio 44135. Refer to LEW-17830-1.

FuzzObserver

Fuzzy Feature Observation Planner for Small Body Proximity Observations (FuzzObserver) is a developmental computer program, to be used along with other software, for autonomous planning of maneuvers of a spacecraft near an asteroid, comet, or other small astronomical body. Selection of terrain features and estimation of the position of the spacecraft relative to these features is an essential part of such planning. FuzzObserver contributes to the selection and estimation by generating recommendations for spacecraft trajectory adjustments to maintain the spacecraft's ability to observe sufficient terrain features for estimating position. The input to FuzzObserver consists of data from terrain images, including sets of data on features acquired during descent toward, or traversal of, a body of interest. The name of this program reflects its use of fuzzy logic to reason about the terrain features represented by the data and extract corresponding trajectory-adjustment rules. Linguistic fuzzy sets and conditional statements enable fuzzy systems to make decisions based on heuristic rule-based knowledge derived by engineering experts. A major advantage of using fuzzy logic is that it involves simple arithmetic calculations that can be performed rapidly enough to be useful for planning within the short times typically available for spacecraft maneuvers.

This program was written by Ayanna Howard and David Bayard 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-41290.

Internet Distribution of Spacecraft Telemetry Data

Remote Access Multi-mission Processing and Analysis Ground Environment (RAMPAGE) is a Java-language server computer program that enables near-real-time display of spacecraft telemetry data on any authorized client computer that has access to the Internet and is equipped with Web-browser software. In addition to providing a variety of displays of the latest available telemetry data, RAMPAGE can deliver notification of an alarm by electronic mail. Subscribers can then use RAMPAGE displays to determine the state of the spacecraft and formulate a response to the alarm, if necessary. A user can query spacecraft mission data in either binary or comma-separated-value format by use of a Web form or a Practical Extraction and Reporting Language (PERL) script to automate the query process. RAMPAGE runs on Linux and Solaris server computers in the Ground Data System (GDS) of NASA's Jet Propulsion Laboratory and includes components designed specifically to make it compatible with legacy GDS software. The client/server architecture of RAMPAGE and the use of the Java programming language make it possible to utilize a variety of competitive server and client computers, thereby also helping to minimize costs.

This program was written by Ted Specht of Caltech and David Noble of Oak Grove Consulting 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-41168.

Semi-Automated Identification of Rocks in Images

Rock Identification Toolkit Suite is a computer program that assists users in identifying and characterizing rocks shown in images returned by the Mars Explorer Rover mission. Included in the program are components for automated finding of rocks, interactive adjustments of outlines of rocks, active contouring of rocks, and automated analysis of shapes in two dimensions. The program assists users in evaluating the surface properties of rocks and soil and reports basic properties of rocks. The program requires either the Mac OS X operating system running on a G4 (or more capable) processor or a Linux operating system running on a Pentium (or more capable) processor, plus at least 128MB of random-access memory.

This program was written by Benjamin Bornstein, Andres Castano, and Robert An-