ALL-SKY IMAGE BEFORE FRAME SUBTRACTION
The ISS is concealed by glare from the horizon

ALL-SKY IMAGE AFTER FRAME SUBTRACTION
The ISS is clearly visible. The bright patch at the left is from trees moving in wind near light sources

IMAGE IN GIMBALLED NARROW-FIELD-OF-VIEW CAMERA

Figure 2. **Images of the International Space Station** (ISS) were acquired by the prototype apparatus and used to track the ISS as it moved across the sky.

manifests itself in the processed image as a bright spot on a dark background (see Figure 2). The moving target is detected computationally as a spot of pixels brighter than a set threshold level. The location of the target is determined, to within a fraction of a pixel, as a brightness-weighted average pixel location. By use of a straightforward transformation that utilizes the image-detector-plane cal-ibration, the target location is converted to azimuth and elevation coordinates, then by use of another calibrated transformation, the azimuth and elevation co-ordinates are converted to gimbal commands for initial aiming of the narrow-field-of-view camera.

Once the narrow-field-of view camera has been initially aimed and has acquired an image of the target, the apparatus switches into a tracking mode. In this mode, the gimbal commands are formulated to move the image of the target toward the center of the CCD image plane.

This work was done by Abhijit Biswas, Christopher Assad, Joseph M Kovalik, Bedabrata Pain, Chris J. Wrigley, and Peter Twiss of Caltech for NASA's Jet Propulsion Laboratory. Further information is contained in a TSP (see page 1). NPO-45237

Visual Data Analysis for Satellites

Stennis Space Center, Mississippi

The Visual Data Analysis Package is a collection of programs and scripts that facilitate visual analysis of data available from NASA and NOAA satellites, as well as dropsonde, buoy, and conventional *in-situ* observations. The package features utilities for data extraction, data quality control, statistical analysis, and data visualization.

The Hierarchical Data Format (HDF) satellite data extraction routines from NASA's Jet Propulsion Laboratory were customized for specific spatial coverage and file input/output. Statistical analysis includes the calculation of the relative error, the absolute error, and the root mean square error. Other capabilities include curve fitting through the data points to fill in missing data points between satellite passes or where clouds obscure satellite data. For data visualization, the software provides customizable Generic Mapping Tool (GMT) scripts to generate difference maps, scatter plots, line plots, vector plots, histograms, time-series, and color fill images.

This program was written by Yee Lau, Sachin Bhate, and Patrick Fitzpatrick of the GeoResources Institute at Mississippi State University for Stennis Space Center.

Inquiries concerning rights for its commercial use should be addressed to:

Mississippi State University
P.O. Box 6156
Mississippi State, MS 39762-5368
Phone No: (228) 688-1157
E-mail: fitz@gri.msstate.edu
Refer to SSC-00266-1, volume and number of this NASA Tech Briefs issue, and the page number.

A Data Type for Efficient Representation of Other Data Types

Some obstacles to programming of parallel computers are removed.

NASA's Jet Propulsion Laboratory, Pasadena, California

A self-organizing, monomorphic data type denoted a sequence has been conceived to address certain concerns (summarized below) that arise in programming parallel computers. ["Sequence" as used here should not be confused with "sequence" as the word is commonly understood or with "sequence" as used elsewhere to denote another, polymorphic data type that is also relevant to computer programming.] A sequence in the present sense can be regarded abstractly as a vector, set, bag, queue, or other construct. A sequence is defined in terms of the behavior of the operators that can be applied to it without any foreknowledge of the underpinnings of its representation or particular implementation.

Heretofore, in programming a parallel computer, it has been necessary for the programmer to state explicitly, at the

outset, what parts of the program and the underlying data structures must be represented in parallel form. Not only is this requirement not optimal from the perspective of implementation; it entails an additional requirement that the programmer have intimate understanding of the underlying parallel structure. Often, it is not possible to have such understanding because hardware and software are designed simultaneously. The present sequence data type overcomes both the implementation and parallel-structure obstacles. In so doing, the sequence data type provides unified means by which the programmer can represent a data structure for natural and automatic decomposition to a parallel computing architecture.

Sequences exhibit the behavioral and structural characteristics of vectors, but the underlying representations are automatically synthesized from combinations of programmers' advice and execution use metrics. Sequences can vary bidirectionally between sparseness and density, making them excellent choices for many kinds of algorithms. The novelty and benefit of this behavior lies in the fact that it can relieve programmers of the details of implementations.

The creation of a sequence enables decoupling of a conceptual representation from an implementation. In essence, a sequence is a fundamental extension of a vector. In the most general case, the length and internal structure of a sequence can be changed during run time, enabling the efficient addition and removal of elements around given positions. Because sequences are not subject to predefined limits in length, they can be used equally to store small and large collections of elements.

Sequences have efficient representations in both time and space for given patterns of use. When the use pattern of a sequence is simple, then the user has the option of causing its basic operations to be coded in line for maximal efficiency.

The underlying representation of a sequence is a hybrid of representations composed of vectors, linked lists, connected blocks, and hash tables. The internal structure of a sequence can automatically change from time to time on the basis of how it is being used. Those portions of a sequence where elements have not been added or removed can be as efficient as vectors. As elements are inserted and removed in a given portion, then different methods are utilized to provide both an access and memory strategy that is optimized for that portion and the use to which it is put.

This work was done by Mark James of Caltech for NASA's Jet Propulsion Laboratory.

Hand-Held Ultrasonic Instrument for Reading Matrix Symbols

All necessary functions would be performed within a compact package.

Marshall Space Flight Center, Alabama

A hand-held instrument that would include an ultrasonic camera has been proposed as an efficient means of reading matrix symbols. The proposed instrument could be operated without mechanical raster scanning. All electronic functions from excitation of ultrasonic pulses through final digital processing for decoding matrix symbols would be performed by dedicated circuitry within the single, compact instrument housing.

The instrument (see figure) would be placed on a selected area on an object of interest believed or suspected to contain a matrix symbol (hereafter denoted, simply, the target). Intimate contact for the purpose of coupling of low-energy ultrasound would be ensured by use of either a flexible membrane camera face or a replaceable gel pad. Ultrasound pulses would be transmitted from a transducer, through the membrane or gel pad, into the target. A portion of each ultrasonic pulse, as modified by any matrix symbol present in the target, would be reflected through the membrane or gel pad to an ultrasound-imaging integrated-circuit chip, which would convert the resulting spatial variation of ultrasound pressure to voltages that could be used to construct a video image of the matrix symbol (if any).

A set of circuit boards above the ultrasound-imaging chip converts the output of the chip into a useful video format and would coordinate timing between the transducer pulses and the acquisition and processing of image data. The system is fully portable and battery powered. The instrument includes the following other boards:

- A pulser board would control the current pulses that drive the acoustic transducer.
- A board comprising a liquid-crystal display unit and its driver circuitry would enable display of the video image in the future. It could include a decoder board that would translate the video image of a matrix symbol into a recog-
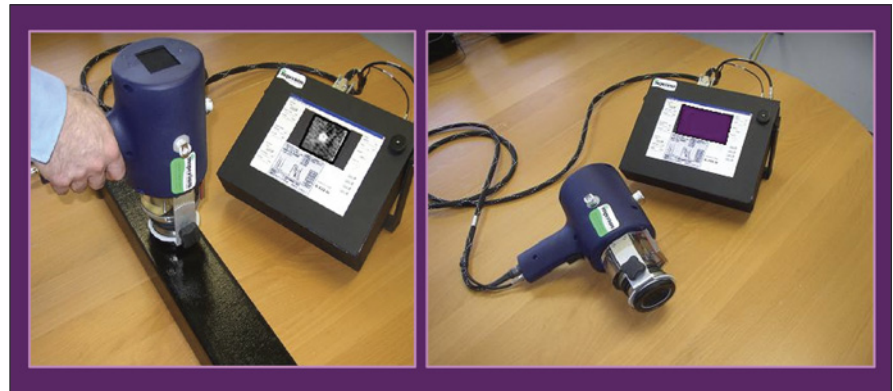
An **Ultrasonic Camera** and associated electronic circuitry would generate and decode a video image of a matrix symbol hidden below the surface of the target.