with reduced run times, as compared to existing rebinning approaches. This approach is able to take advantage of vectorized instructions such as Single Instruction Multiple Data (SIMD), to perform the rebinning operation.

The algorithm completely vectorizes the data rebinning operation, in the sense that a "single" arithmetic operation is applied simultaneously to multiply distinct data sets and is executed with the approximate run time of that operation applied to a single data set. For lower-level computer languages, such as C or assembly, vectorized operations can be implemented using central processing unit (CPU) single-instruction, multiple-data (SIMD) capabilities, such as streaming SIMD Extensions 3 (SSE3) on x86 computer architecture or AltiVec on PowerPC processors. Thus, although the algorithm has been implemented using MATLAB, it is not fundamentally tied to MATLAB, and can be implemented using other programming languages.

The vectorized data rebinning (downsampling) procedure offers a reduced run time when compared with standard rebinning algorithms. In general, algorithms are often optimized by trading decreased run time for increased memory, where the latter is needed for storing additional code, pre-computed results, or other ancillary data. However, the vectorized rebinning approach does not have increased memory requirements compared with conventional approaches. The underlying fundamental advantage to this technology is the utilization of vectorized instructions for the rebinning operation.

This work was done by Bruce Dean, David Aronstein, and Jeffrey Smith of Goddard Space Flight Center. Further information is contained in a TSP (see page 1). GSC-15949-1

Display Provides Pilots with Real-Time Sonic-Boom Information

The impact of sonic booms can be controlled over populated areas.

Dryden Flight Research Center, Edwards, California

Supersonic aircraft generate shock waves that move outward and extend to the ground. As a cone of pressurized air spreads across the landscape along the flight path, it creates a continuous sonic boom along the flight track. Several factors can influence sonic booms: weight, size, and shape of the aircraft; its altitude and flight path; and weather and atmospheric conditions. This technology allows pilots to control the impact of sonic booms.

A software system displays the location and intensity of shock waves caused by supersonic aircraft. This technology can be integrated into cockpits or flight control rooms to help pilots minimize sonic-boom impact in populated areas. The system processes vehicle and flight parameters as well as data regarding current atmospheric conditions. The display provides real-time information regarding sonic boom location and intensity, enabling pilots to make the necessary flight adjustments to control the timing and location of sonic booms. This technology can be used on current-generation supersonic aircraft, which generate loud sonic booms, as well as future-generation, low-boom aircraft, anticipated to be quiet enough for populated areas.

When fully deployed in real time, the display will leverage existing tools developed and enhanced by the U.S. Air Force and NASA to predict sonic boom parameters. The prediction data will be integrated with a real-time, local-area, moving-map display that is capable of displaying the aircraft's current sonic boom footprint at all times. The pilot will be able to choose from a menu of pre-programmed maneuvers such as accelerations, turns, or pushovers, and the predicted sonic boom footprint for that maneuver appears on the map. After fully developed and implemented, this will allow the pilot to select or modify parameters to either avoid generating a sonic boom or to place the sonic boom in a specific location. The system may also provide pilots with guidance on how to execute the chosen maneuver.

This technology will enable supersonic commercial flight without disturbing population centers on the ground.

This work was done by Ed Haering of Dryden Flight Research Center and Ken Plotkin of Wyle. Further information is contained in a TSP (see page 1). DRC-008-001

Onboard Algorithms for Data Prioritization and Summarization of Aerial Imagery

Clustering/machine learning methods are used to structure data for prioritization, mapping, and downlinking.

NASA's Jet Propulsion Laboratory, Pasadena, California

Many current and future NASA missions are capable of collecting enormous amounts of data, of which only a small portion can be transmitted to Earth. Communications are limited due to distance, visibility constraints, and competing mission downlinks. Long missions and high-resolution, multispectral imaging devices easily produce data exceeding the available bandwidth. To address this situation, computationally efficient algorithms were developed for analyzing science imagery onboard the spacecraft. These algorithms autonomously cluster the data into classes of similar imagery, enabling selective downlink of representatives of each class, and a map classifying the terrain imaged rather than the full dataset, reducing the volume of the downlinked data. A range of approaches was examined, including k-means clustering using image features based on color, texture, temporal, and spatial arrangement.

Several unique challenges influenced design decisions for automatic image analysis. First, onboard processing is limited in spaceflight applications. Avionics computers must satisfy strict radiation and energy constraints, and their resources are shared between continuous autonomous control and data processing. Computational constraints mandate a simple approach to image analysis in which statistical properties of the image serve as proxies for the actual content.

A major challenge is the diversity of surface features an aerobot might encounter. An aerobot would be in constant motion but difficult to control due to unpredictable atmospheric currents. It would be difficult to schedule image targets in advance or to anticipate the features of interest that will appear. This favors an "unsupervised" approach that makes few assumptions about image content but instead discovers interesting and representative samples based on the intrinsic properties of the data. Clustering is one common unsupervised approach; it classifies a dataset into discrete categories of items with similar properties.

Image features can be considered to fall into one of four groups, or themes, based on the properties they describe. These are color, edge, frequency, and time. The edge and frequency features correlate with image texture, color captures basic color statistics, and time describes the temporal order in which the images were collected.

The main feature of this innovation is the use of clustering/machine learning methods to structure data for prioritization, mapping, and downlink. The effectiveness of clustering rests on the quality of the feature vectors describing each set of data. Features that are redundant, or have little variance, will reduce the effectiveness of clustering. Dimensionality reduction techniques such as principal component analysis (PCA) can transform a high-dimensional feature space into a lower-dimensional space where the new, uncorrelated features have heightened variance. Ideal clusterings contain compact clusters that are spread far apart from one another.

This work was done by Steve A. Chien, David Hayden, David R. Thompson, and Rebecca Castano of Caltech for NASA's Jet Propulsion Laboratory. For more information, contact iaoffice@jpl.nasa.gov. NPO-47534

Monitoring and Acquisition Real-time System (MARS)

Marshall Space Flight Center, Alabama

MARS is a graphical user interface (GUI) written in MATLAB and Java, allowing the user to configure and control the Scalable Parallel Architecture for Real-Time Acquisition and Analysis (SPARTAA) data acquisition system. SPARTAA not only acquires data, but also allows for complex algorithms to be applied to the acquired data in real time. The MARS client allows the user to set up and configure all settings regarding the data channels attached to the system, as well as have complete control over starting and stopping data acquisition. It provides a unique "Test" programming environment, allowing the user to create tests consisting of a series of alarms, each of which contains any number of data channels. Each alarm is configured with a particular algorithm, determining the type of processing that will be applied on each data channel and tested against a defined threshold. Tests can be uploaded to SPARTAA, thereby teaching it how to process the data.

MARS was developed as a front-end GUI for setup, control, and plotting of data from SPARTAA. The system was designed to monitor spectral components in real time from instrumentation located on high-speed rotational hardware (primarily high-pressure turbopumps), and to issue cut commands to a facility if preset levels were violated. However, the system is not limited to rotational hardware, and can be used to monitor any level of frequency information from a myriad of instrumented test hardware. The control software allows the user to configure the system easily to support testing of various configurations with multiple alarms, voting logic, and sensor validation,

The uniqueness of MARS is in its capability to be adaptable easily to many test configurations. Test hardware measurement limits (i.e. vibration, pressure, temperature, etc.) can be predetermined, and MARS can be used to set up and support quickly any test configuration. Multiple alarms with various timings can be configured within minutes, as opposed to previous software modifications. MARS sends and receives protocols via TCP/IP, which allows for quick integration into almost any test environment. The use of MATLAB and Java as the programming languages allows for developers to integrate the software across multiple operating platforms.

This work was done by Corbin Holland of Marshall Space Flight Center. For more information, contact Sammy Nabors, MSFC Commercialization Assistance Lead, at sammy.a.nabors@nasa.gov. Refer to MFS-32905-1.