vides a mechanism to establish trust among deployed AIUs based on recombining shared secrets, authentication and verify users with a username, X.509 certificate, enclave information, and classification level. AIU achieves data protection through (1) splitting data into multiple information pieces using the Shamir's secret sharing algorithm, (2) encrypting each individual information piece using military-grade AES-256 encryption, and (3) randomizing the position of the encrypted data based on the unbiased and memory efficient inplace Fisher-Yates shuffle method. Therefore, it becomes virtually impossible for attackers to compromise data since attackers need to obtain all distributed information as well as the encryption key and the random seeds to properly arrange the data. In addition, since policy can be associated with data in the AIU, different user access and data control strategies can be included.

The AIU technology can greatly enhance information assurance and security management in the bandwidth-limited and ad hoc net-centric environments. In addition, AIU technology can be applicable to general complex network domains and applications where distributed user authentication and data protection are necessary. AIU achieves fine-grain data access and user control, reducing the security risk significantly, simplifying the complexity of various security operations, and providing the high information assurance across different network domains.

This work was done by Edward T. Chow, Simon S. Woo, Mark James, and George K. Paloulian 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-48224.

Vehicle Detection for RCTA/ANS (Autonomous Navigation System)

This algorithm can be applied to semi-autonomous vehicles for driver assistance, and to military robots.

NASA's Jet Propulsion Laboratory, Pasadena, California

Using a stereo camera pair, imagery is acquired and processed through the "JPLV" stereo processing pipeline. From this stereo data, large 3D blobs are found. These blobs are then described and classified by their shape to determine which are vehicles and which are not. Prior vehicle detection algorithms are either targeted to specific domains, such as following lead cars, or are intensity-based methods that involve learning typical vehicle appearances from a large corpus of training data.

In order to detect vehicles, the JPL Vehicle Detection (JVD) algorithm goes through the following steps:

1. Take as input a left disparity image and left rectified image from JPLV stereo.
2. Project the disparity data onto a two-dimensional Cartesian map.
3. Perform some post-processing of the map built in the previous step in order to clean it up.
4. Take the processed map and find peaks. For each peak, grow it out into a map blob. These map blobs represent large, roughly vehicle-sized objects in the scene.
5. Take these map blobs and reject those that do not meet certain criteria. Build descriptors for the ones that remain. Pass these descriptors onto a classifier, which determines if the blob is a vehicle or not.

The probability of detection is the probability that if a vehicle is present in the image, is visible, and un-occluded, then it will be detected by the JVD algorithm. In order to estimate this probability, eight sequences were ground-truthed from the RCTA (Robotics Collaborative Technology Alliances) program, totaling over 4,000 frames with 15 unique vehicles. Since these vehicles were observed at varying ranges, one is able to find the probability of detection as a function of range. At the time of this reporting, the JVD algorithm was tuned to perform best at cars seen from the front, rear, or either side, and perform poorly on vehicles seen from oblique angles.

This work was done by Shane Brennan, Max Bajracharya, Larry H. Matthies, and Andrew B. Howard of Caltech for NASA's Jet Propulsion Laboratory. Further information is contained in a TSP (see page 1). NPO-47569

Image Mapping and Visual Attention on the Sensory Ego-Sphere

This technology can be used to map a robot's environment and direct its attention.

Lyndon B. Johnson Space Center, Houston, Texas

The Sensory Ego-Sphere (SES) is a short-term memory for a robot in the form of an egocentric, tessellated, spherical, sensory-motor map of the robot's locale. Visual attention enables fast alignment of overlapping images without warping or position optimization, since an attentional point (AP) on the composite typically corresponds to one on each of the collocated regions in the images. Such alignment speeds analysis of the multiple images of the area.

Compositing and attention were performed two ways and compared: (1) APs were computed directly on the composite and not on the full-resolution images until the time of retrieval; and (2) the attentional operator was applied to all incoming imagery. It was found that although the second method was slower, it produced consistent and, thereby, more useful APs.

The SES is an integral part of a control system that will enable a robot to learn new behaviors based on its previous experiences, and that will enable it to recombine its known behaviors in such a way as to solve related, but novel, task problems with apparent creativity. The approach is to combine sensory-motor data association and dimensionality reduction to learn navigation and manipulation tasks as sequences of basic behaviors that can be implemented with a small set of closed-loop controllers. Over time, the aggregate of behaviors and their transition probabilities form a stochastic network. Then given a task, the robot finds a path in the network that leads from its current state to the goal.

The SES provides a short-term memory for the cognitive functions of the robot, association of sensory and motor data via spatio-temporal coincidence, direction of the attention of the robot, navigation through spatial localization with respect to known or discovered landmarks, and structured data sharing between the robot and human team members, the individuals in multi-robot teams, or with a C3 center.

This work was done by Katherine Achim Fleming and Richard Alan Peters II of Vanderbilt University for Johnson Space Center. Further information is contained in a TSP (see page 1).

In accordance with Public Law 96-517, the contractor has elected to retain title to this invention. Inquiries concerning rights for its commercial use should be addressed to:

Vanderbilt University, School of Engineering
Department of Electrical Engineering and Computer Science
VU Station B 351824,
2301 Vanderbilt Place
Nashville, TN 37235

Refer to MSC-24363-1, volume and number of this NASA Tech Briefs *issue, and the page number.*

HyDE Framework for Stochastic and Hybrid Model-Based Diagnosis

It uses hybrid models built by the users and sensor data from the system to deduce the state of the system over time.

Ames Research Center, Moffett Field, California

Hybrid Diagnosis Engine (HyDE) is a general framework for stochastic and hybrid model-bused diagnosis that offers flexibility to the diagnosis application designer. The HyDE architecture supports the use of multiple modeling paradigms at the component and system level. Several alternative algorithms are available for the various steps in diagnostic reasoning. This approach is extensible, with support for the addition of new modeling paradigms as well as diagnostic reasoning algorithms for existing or new modeling paradigms.

HyDE is a general framework for stochastic hybrid model-based diagnosis of discrete faults; that is, spontaneous changes in operating modes of components. HyDE combines ideas from consistency-based and stochastic approaches to model- based diagnosis using discrete and continuous models to create a flexible and extensible architecture for stochastic and hybrid diagnosis. HyDE supports the use of multiple paradigms and is extensible to support new paradigms.

HyDE generates candidate diagnoses and checks them for consistency with the observations. It uses hybrid models built by the users and sensor data from the system to deduce the state of the system over time, including changes in state indicative of faults.

At each time step when observations are available, HyDE checks each existing candidate for continued consistency with the new observations. If the candidate is consistent, it continues to remain in the candidate set. If it is not consistent, then the information about the inconsistency is used to generate successor candidates while discarding the candidate that was inconsistent.

The models used by HyDE are similar to simulation models. They describe the expected behavior of the system under nominal and fault conditions. The model can be constructed in modular and hierarchical fashion by building component/subsystem models (which may themselves contain component/subsystem models) and linking them through shared variables/parameters. The component model is expressed as operating modes of the component and conditions for transitions between these various modes. Faults are modeled as transitions whose conditions for transitions are unknown (and have to be inferred through the reasoning process).

Finally, the behavior of the components is expressed as a set of variables/parameters and relations governing the interaction between the variables. The hybrid nature of the systems being modeled is captured by a combination of the above transitional model and behavioral model. Stochasticity is captured as probabilities associated with transitions (indicating the likelihood of that transition being taken), as well as noise on the sensed variables.

This work was done by Sriram Narasimhan and Lee Brownston of Ames Research Center. Further information is contained in a TSP (see page 1). ARC-15570-1

IMAGESEER — IMAGEs for Education and Research

Web portal shares image data with research institutions.

Goddard Space Flight Center, Greenbelt, Maryland

IMAGESEER is a new Web portal that brings easy access to NASA image data for non-NASA researchers, educators, and students. The IMAGESEER Web site and database are specifically designed to be utilized by the university community, to enable teaching image processing (IP) techniques on NASA data, as well as to provide reference benchmark data to