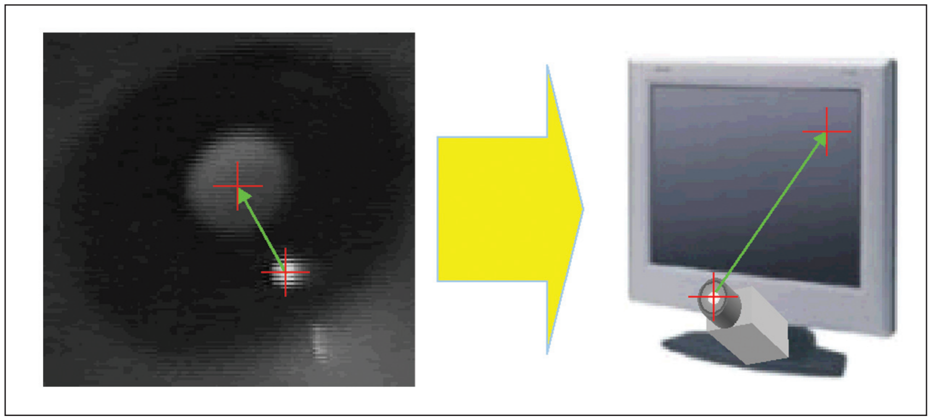
much as the image features of interest (the cornea and pupil) typically occupy a small part of the camera frame, this ROI capability can be exploited to determine the direction of gaze at a high frame rate by reading out from the ROI that contains the cornea and pupil (but not from the rest of the image) repeatedly.

One of the present algorithms exploits the ROI capability. The algorithm takes horizontal row slices and takes advantage of the symmetry of the pupil and cornea circles and of the gray-scale contrasts of the pupil and cornea with respect to other parts of the eye. The algorithm determines which horizontal image slices contain the pupil and cornea, and, on each valid slice, the end coordinates of the pupil and cornea. Information from multiple slices is then combined to robustly locate the centroids of the pupil and cornea images.



The **Vector Between the Centroids** of pupil and corneal reflections is computed and then used to compute the direction of gaze and the gaze point.

The other of the two present algorithms is a modified version of an older algorithm for estimating the direction of gaze from the centroids of the pupil and cornea. The modification lies in the use of the coordinates of the centroids, rather than differences between the coordinates of the centroids, in a gaze-mapping equation. The equation locates a gaze point, defined as the intersection of the gaze axis with a surface of interest, which is typically a computer display screen (see figure). The expected advantage of the modification is to make the gaze computation less dependent on some simplifying assumptions that are sometimes not accurate.

This work was done by Ashit Talukder, John-Michael Morookian, and James Lambert of Caltech for NASA's Jet Propulsion Laboratory. 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:

Innovative Technology Assets Management
JPL
Mail Stop 202-233
4800 Oak Grove Drive
Pasadena, CA 91109-8099
E-mail: iaoffice@jpl.nasa.gov

Refer to NPO-30699, volume and number of this NASA Tech Briefs *issue, and the page number.*

Adapting ASPEN for Orbital Express

Declarative modeling brings efficiency to encoded procedures and allows for guarantees on resource usage and time usage.

NASA's Jet Propulsion Laboratory, Pasadena, California

By studying the Orbital Express mission, modeling the spacecraft and scenarios, and testing the system, a technique has been developed that uses recursive decomposition to represent procedural actions declaratively, schema-level uncertainty reasoning to make uncertainty reasoning tractable, and lightweight, natural language processing to automatically parse procedures to produce declarative models.

Schema-level uncertainty reasoning has, at its core, the basic assumption that certain variables are uncertain, but not independent. Once any are known, then the others become known. This is important where a variable is uncertain for an action and many actions of the same type exist in the plan. For example, if the number of retries to purge pump lines was unknown (but bounded), and each attempt required a sub-plan, then, once the correct number of attempts required for a purge was known, it would likely be the same for all subsequent purges. This greatly reduces the space of plans that needs to be searched to ensure that all executions are feasible.

To accommodate changing scenario procedures, each is ingested into a tabular format in temporal order, and a simple natural-language parser is used to read each step and to derive the impact of that step on memory, power, and communications. Then an ASPEN (Activity Scheduling and Planning Environment) model is produced based on this analysis. The model is tested and further changed by hand, if necessary, to reflect the actual procedure. This results in a great savings of time used for modeling procedures.

Many processes that need to be modeled in ASPEN (a declarative system) are, in fact, procedural. ASPEN includes the ability to model activities in a hierarchical fashion, but this representation breaks down if there is a practically unbounded number of sub-activities and decomposition topologies. However, if recursive decomposition is allowed, HTN-like encodings are enabled to represent most procedural phenomena.

For example, if a switch requires a variable (but known at the time of the attempt) number of attempts to switch on, one can recurse on the number of remaining switch attempts and decompose into either the same switching activity with one less required attempt, or not decompose at all (or decompose into a dummy task), resulting in the end of the decomposition. In fact, any bounded procedural behavior can be modeled using recursive decompositions assuming that the variables impinging the disjunctive decomposition decision are computable at the time that the decision is made. This enables one to represent tasks that are controlled outside of the scheduler, but that the scheduler must accommodate, without requiring one to give a declarative model of the procedural behavior.

This work was done by Caroline Chouinard, Daniel Tran, Grailing Jones, Van Dang, and Russell Knight of Caltech for NASA's Jet Propulsion Laboratory.

The software used in this innovation is available for commercial licensing. Please contact Daniel Broderick of the California Institute of Technology at danielb@caltech.edu. Refer to NPO-45262.