The steps include setting up a region of interest, a start position, and a stop position, as well as initially random traversal waypoints. The optimization routine moves the way points around for each candidate solution and attempts to evolve the best path with regards to the reference cost function. The program calculates a path connecting all the waypoints from start to finish, and feeds this path to a cost function. The cost function determines various metrics such as length of path, collision with obstacles, work required to traverse the path, smoothness, weight on exploration of new territory vs. tracking the original outbound path, etc. The calculation of the optimal path is iterative; several rounds of feeding candidate solutions and using their associated costs to calculate new candidate solutions are required. The practical result of the pairing of this cost function strategy with PSO is that an optimal path is evolved much faster than random search, and completely forgiving of discontinuities.

The path planning prototype can be re-tasked on the fly and uses a unique "way point" optimization strategy. Unlike other optimization strategies, this one will work in a discontinuous environment with no modification necessary and is guaranteed to provide a continuous path from start to finish.

*This work was done by Brian Birge of L-3 Communications for Johnson Space Center. Further information is contained in a TSP (see page 1). MSC-24864-1*

# Smart-Divert Powered Descent Guidance to Avoid the Backshell Landing Dispersion Ellipse

**The software and methods are valid for planetary or lunar powered descent.**

*NASA's Jet Propulsion Laboratory, Pasadena, California*

A smart-divert capability has been added into the Powered Descent Guidance (PDG) software originally developed for Mars pinpoint and precision landing. The smart-divert algorithm accounts for the landing dispersions of the entry backshell, which separates from the lander vehicle at the end of the parachute descent phase and prior to powered descent. The smart-divert PDG algorithm utilizes the onboard fuel and vehicle thrust vectoring to mitigate landing error in an intelligent way: ensuring that the lander touches down with minimum-fuel usage at the minimum distance from the desired landing location that also avoids impact by the descending backshell.

The smart-divert PDG software implements a computationally efficient, convex formulation of the powered-descent guidance problem to provide pinpoint or precision-landing guidance solutions that are fuel-optimal and satisfy physical thrust bound and pointing constraints, as well as position and speed constraints. The initial smart-divert implementation enforced a lateral-divert corridor parallel to the ground velocity vector; this was based on guidance requirements for MSL (Mars Science Laboratory) landings. This initial method was overly conservative since the divert corridor was infinite in the down-range direction despite the backshell landing inside a calculable dispersion ellipse. Basing the divert constraint instead on a local tangent to the backshell dispersion ellipse in the direction of the desired landing site provides a far less conservative constraint. The resulting enhanced smart-divert PDG algorithm avoids impact with the descending backshell and has reduced conservatism.

*This work was done by John M. Carson and Behcet Acikmese of Caltech for NASA's Jet Propulsion Laboratory. For more information, contact iaoffice@jpl.nasa.gov.*

*The software used in this innovation is available for commercial licensing. Please contact Dan Broderick at Daniel.F.Broderick@jpl.nasa.gov. Refer to NPO-47884.*

# Estimating Foreign-Object-Debris Density From Photogrammetry Data

*John F. Kennedy Space Center, Florida*

Within the first few seconds after launch of STS-124, debris traveling vertically near the vehicle was captured on two 16-mm film cameras surrounding the launch pad. One particular piece of debris caught the attention of engineers investigating the release of the flame trench fire bricks. The question to be answered was if the debris was a fire brick, and if it represented the first bricks that were ejected from the flame trench wall, or was the object one of the pieces of debris normally ejected from the vehicle during launch. If it was typical launch debris, such as SRB throat plug foam, why was it traveling vertically and parallel to the vehicle during launch, instead of following its normal trajectory, flying horizontally toward the north perimeter fence?

By utilizing the Runge-Kutta integration method for velocity and the Verlet integration method for position, a method that suppresses trajectory computational instabilities due to noisy position data was obtained. This combination of integration methods provides a means to extract the best estimate of drag force and drag coefficient under the non-ideal conditions of limited position data. This integration strategy leads immediately to the best possible estimate of object density, within the constraints of unknown particle shape. These types of calculations do not exist in readily available off-the-shelf simulation software, especially where photogrammetry data is needed as an input.

A robust numerical method of iteratively solving for the drag force and coefficient of drag of an unknown object has been developed and implemented in Mathematica in a form readily convertible to other codes. This algorithm is based on an innovative combination of the Verlet and Runge-Kutta integration methods. The input data is object position data as a function of time, which might, for example, be based on a previous photogrammetry analysis. This new method is not limited to object location based on photogrammetry.

*This work was done by Jason Long and Philip Metzger of Kennedy Space Center, and John Lane of ASRC Aerospace Corporation. Further information is contained in a TSP (see page 1). KSC-13251*

# Adaptive Sampling of Spatiotemporal Phenomena With Optimization Criteria

*NASA's Jet Propulsion Laboratory, Pasadena, California*

This work was designed to find a way to optimally (or near optimally) sample spatiotemporal phenomena based on limited sensing capability, and to create a model that can be run to estimate uncertainties, as well as to estimate covariances. The goal was to maximize (or minimize) some function of the overall uncertainty.

The uncertainties and covariances were modeled presuming a parametric distribution, and then the model was used to approximate the overall information gain, and consequently, the objective function from each potential sense. These candidate sensings were then cross-checked against operation costs and feasibility. Consequently, an operations plan was derived that combined both operational constraints/costs and sensing gain.

Probabilistic modeling was used to perform an approximate inversion of the model, which enabled calculation of sensing gains, and subsequent combination with operational costs. This incorporation of operations models to assess cost and feasibility for specific classes of vehicles is unique.

*This work was done by Steve A. Chien and David R. Thompson of Caltech, and Kian Hsiang Low of the National University of Singapore for NASA's Jet Propulsion Laboratory. Further information is contained in a TSP (see page 1). NPO-48664*

# Building a 2.5D Digital Elevation Model From 2D Imagery

## High-quality DEMs are generated from a collection of 2D images.

*NASA's Jet Propulsion Laboratory, Pasadena, California*

When projecting imagery into a georeferenced coordinate frame, one needs to have some model of the geographical region that is being projected to. This model can sometimes be a simple geometrical curve, such as an ellipse or even a plane. However, to obtain accurate projections, one needs to have a more sophisticated model that encodes the undulations in the terrain including things like mountains, valleys, and even manmade structures. The product that is often used for this purpose is a Digital Elevation Model (DEM).

The technology presented here generates a high-quality DEM from a collection of 2D images taken from multiple viewpoints, plus pose data for each of the images and a camera model for the sensor. The technology assumes that the images are all of the same region of the environment.

The pose data for each image is used as an initial estimate of the geometric relationship between the images, but the pose data is often noisy and not of sufficient quality to build a high-quality DEM. Therefore, the source imagery is passed through a feature-tracking algorithm and multi-plane-homography algorithm, which refine the geometric transforms between images. The images and their refined poses are then passed to a stereo algorithm, which generates dense 3D data for each image in the sequence. The 3D data from each image is then placed into a consistent coordinate frame and passed to a routine that divides the coordinate frame into a number of cells. The 3D points that fall into each cell are collected, and basic statistics are applied to determine the elevation of that cell. The result of this step is a DEM that is in an arbitrary coordinate frame. This DEM is then filtered and smoothed in order to remove small artifacts.

The final step in the algorithm is to take the initial DEM and rotate and translate it to be in the world coordinate frame [such as UTM (Universal Transverse Mercator), MGRS (Military Grid Reference System), or geodetic] such that it can be saved in a standard DEM format and used for projection.

*This work was done by Curtis W. Padgett, Adnan I. Ansar, Shane Brennan, Yang Cheng, Daniel S. Clouse, and Eduardo Almeida of Caltech for NASA's Jet Propulsion Laboratory. For more information, contact iaoffice@jpl.nasa.gov.*

*The software used in this innovation is available for commercial licensing. Please contact Dan Broderick at Daniel.F.Broderick@jpl.nasa.gov. NPO-47571.*