JET PROPULSION LABORATORY **INTERNAL DOCUMENT**
JPL D-32943
September 23, 2005

To: David S. Bayard (Fiscal Year 2005 R&TD on Small Body Guidance, Navigation, and Control)
From: Adnan Ansar
Subject: Small Body GN&C Research Report: Feature Recognition Algorithms

ABSTRACT

This report consists of two parts. In the first, we present a number of enhancements to the feature recognition work presented in last year's GN&C Research Report on feature recognition algorithms [1]. These include improvements to the scale invariance properties of detected features, greater robustness in feature recognition, and implementation changes intended to decrease processing time. The second part focuses on the *Twice Around Study* in which synthetic imagery of a virtual camera orbiting a small body is used for catalog generation, feature detection and state estimation. We will present in this report details of the computer vision aspects of this study. This includes incorporation of previous algorithmic work, additional enhancements and details of the implementation.

1 Introduction

In last year's report [1], we presented a basic infrastructure for scale invariant feature detection adapted from David Lowe's SIFT algorithm [2]. The goal of this work was to establish a generic class of visual **General Landmarks** which could be generated and cataloged automatically during a small body mission and then subsequently recognized for precise localization of the spacecraft, whether for navigation purposes, sample return or reacquisition of a scientifically interesting site.

The focus of feature detection work during this fiscal year has been on testing these techniques in simulation and adapting them as needed based on performance. Thus, we have developed an Estimation, Sensing and Perception (ESP) testbed in which the vision and estimation aspects of the Small Body R&TD task have interfaced. This simulation environment has been the driver for work in feature detection, both in terms of algorithmic modifications and implementation.

The focus of effort in the ESP testbed has thus far been a simple scenario in which a spacecraft orbits a small body with ~ 500 m radius in an approximately 2 km orbit. We refer to this as the *Twice Around Study*. The title is derived from a two stage process in which two or more orbits of a candidate small body are performed. The first orbit is used for catalog generation and subsequent orbits to evaluate automatic localization and trajectory determination. In the current version of the simulation, we use the ground truth spacecraft trajectory in combination with stereo vision techniques to generate a catalog of landmarks, called the Feature Catalog (FCAT). This catalog contains the 3D positions of landmarks with associated covariances as well as the descriptors used for later identification of the landmarks. Later, modified orbits are used to test the ability to recognize previously seen landmarks. During the second pass, landmarks are detected, matched to the catalog and stored in a Landmark Table (LMT) on a frame-by-frame basis. The LMT contains for each landmark, the bearing angles to its 3D position on the target body, associated covariance estimates, and the landmark descriptors. The LMT is then supplied to the state estimator, which filters this data to estimate a trajectory for comparison to ground truth. We also produce a vision-based pose (position and attitude) estimate for the camera from single frame measurements as both a sanity check for the estimator and as an outlier detection mechanism for the feature detection algorithm. Finally, we produce an interest operator based frame-to-frame set of feature correspondences. This uses normalized cross correlation to identify features across adjacent frames. The information is recorded in the Paired Feature Table (PFT) for delivery to the estimator. While the PFT datatype lacks an absolute reference, it provides enough information for a velocity-like estimate similar to the Descent Image Motion Estimation System (DIMES) on the Mars Exploration Rover (MER) landers [3].

A parallel effort [4] is underway to use bundle adjustment techniques for catalog determination. This will eliminate the current dependence on ground truth trajectory for the catalog generation step. However, since the majority of work this year has used the earlier version of the catalog generation scheme, we will describe it in detail below.

We begin with an overview of modifications made to the work presented in the last fiscal year in response to challenges posed by integration of vision algorithms into the ESP testbed.

2 Algorithmic Enhancements

2.1 Robustness / Outlier Rejection

Our earliest attempts at integrating vision based landmark detection and identification into the ESP testbed met with only marginal success. This was due in large part to false matches in the LMT being reported to the estimator. Thus, outlier rejection and increased robustness of the matching scheme have been a major focus of effort this year.

2.1.1 Pose estimation and RANSAC

Our current outlier rejection scheme used in constructing the LMT depends heavily on vision based pose estimation. Pose estimation refers to recovery of the 6 DOF position and attitude of a camera based on image data, specifically known correspondences between 3D points and their 2D image projections. This is exactly the setting in which the LMT operates, since we match 3D landmarks in the target coordinate frame with their projections in 2D imagery acquired during orbit. We presented pose estimation as a possible side benefit of the computer vision work in last year's report [1], primarily for use as a sanity check on the estimator. Now, in addition to this, we are using it successfully for outlier rejection.

The principal is simple. Say $(R, T) \in SE(3)$ is the Euclidean transformation between the target and camera coordinate frames represented as a rotation matrix and translation vector. Then in homogeneous pixel coordinates the camera projection p of a 3D point P expressed in the body frame of the target is given by

$$\begin{pmatrix} \hat{p}_x \\ \hat{p}_y \\ \hat{p}_z \end{pmatrix} = A\,[R \mid T] = A \begin{pmatrix} r_{11} & r_{12} & r_{13} & T_x \\ r_{21} & r_{22} & r_{23} & T_y \\ r_{31} & r_{32} & r_{33} & T_z \end{pmatrix} \begin{pmatrix} P_x \\ P_y \\ P_z \\ 1 \end{pmatrix} \tag{1}$$

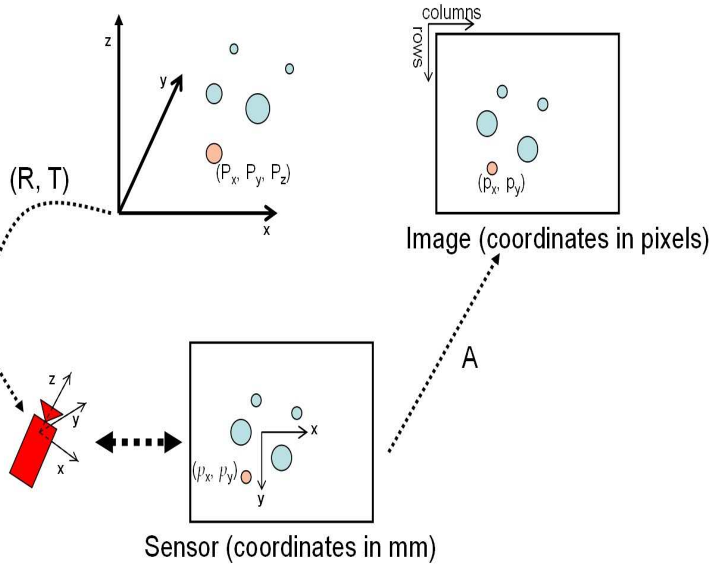

Figure 1: A point (P_x, P_y, P_z) is first transformed into the camera coordinate frame by the Euclidean motion (R, T). The resulting image on the sensor plane is transformed into pixel coordinates by A.

where the pixel coordinates of p are $(p_x, p_y) = (\frac{\hat{p}_x}{\hat{p}_z}, \frac{\hat{p}_y}{\hat{p}_z})$ and

$$A = \begin{pmatrix} f_x & \kappa & u_x \\ 0 & f_y & u_y \\ 0 & 0 & 1 \end{pmatrix}$$

with f_x, f_y representing the horizontal and vertical focal lengths of the virtual camera, κ the skewness of the sensor plane, and (u_x, u_y) the pixel coordinates of the image center. In our simulations, these parameters are computed from an explicit set of rendered images using standard calibration techniques. In a real sensor, we must also compensate for non-linearity in the camera projection model. However, this is a well understood process and we ignore it here. The steps in Eq. 1 are shown in graphical form in Fig. 1.

If (R, T) is known accurately, we can determine whether a given match in the LMT is false by measuring its deviation in pixel coordinates from Eq. 1. More precisely, if the point P projects to $(x_{proj}, y_{proj}) = (\frac{px}{pz}, \frac{py}{pz})$ following Eq. 1 and the location of the matched image point extracted by the feature detection algorithm is (x_{meas}, y_{meas}), then outlier amounts to a threshold on

$$\varepsilon_{reproj} = \sqrt{(x_{proj} - x_{meas})^2 + (y_{proj} - y_{meas})^2} \tag{2}$$

Any point for which the reprojection error ε_{reproj} in Eq. 2 is too large is rejected.

However, our only mechanism for determining (R, T) in the first place is the LMT. Provided there are enough matches and under the assumption that most are correct, we solve the pose problem using the RANSAC framework [5]. This involves recovering the camera pose by standard methods [6, 7] for several minimal subsets of the LMT. We then accept as valid the camera pose which produces the smallest median error in Eq. 1 over all points in the LMT. The resulting (R, T) is used for outlier detection. We outline the procedure.

- For m iterations:
    - Randomly select n entries from the LMT
    - If $n = 4$ use [6] to solve pose
    - If $n > 4$ initialize with [6] and refine with [7] to solve pose
    - Compute the median error from Eq. 2 over all points in LMT
        - If median error is smaller than previous smallest median error, save (R, T) as best model
- Compute error for all points in LMT using best (R, T)
- Reject outliers using test based on fourth spread of errors (discussed below)
- Construct LMT using landmark matches not rejected
- Recompute (R, T) using all inliers

In the above discussion, m is chosen in a statistically meaningful way following [5] and based on estimates for likelihood of any given point in the LMT being valid. We typically choose $n = 5$ unless there are too few points. Note that the algorithm described in [7] is an iterative technique that requires a minimum of 5 points. By its nature, it is subject to convergence and local minima issues. On the other hand, the algorithm described in [6] is algebraic, does not become trapped in local minima, and works with as few as 4 points. However, it is somewhat less accurate. The combination of the two has proved quite useful in our work.

The use of the median error for model estimation in the inner loop above is less prone to bias due to single extreme outliers. Finally, the fourth spread is a well established, simple procedure for outlier elimination for many symmetric 1-dimensional distributions. Given a distribution d, let med(d) represent the median. Then the fourth spread, $\mathcal{F}$ is defines by

$$\begin{aligned} d_{lo} &= \{p \in d : p < \text{med}(d)\} \qquad m_{lo} = \text{med}(d_{lo}) \\ d_{hi} &= \{p \in d : p > \text{med}(d)\} \qquad m_{hi} = \text{med}(d_{hi}) \\ \mathcal{F} &= m_{hi} - m_{lo} \end{aligned}$$

Outliers are marked as points in the set

$$\{p \in d : p < m_{lo} - 1.5 * \mathcal{F}\} \cup \{p \in d : p > m_{hi} + 1.5 * \mathcal{F}\}$$

In the case of a zero-mean, Gaussian distribution, the cutoff for the fourth spread, i.e. $m_{hi} + 1.5\mathcal{F}$, corresponds to approximately 2.7 standard deviations.

Observe in the above discussion that if the size of the LMT is < 5, pose estimation does not help. Also, while we can characterize the probability of finding a poor model given the likelihood of any given match being bad, this probability is always greater than zero. Hence, pose estimation even in the case of sufficiently many points will minimize but not eliminate the possibility of bad matches in the LMT.

We are still investigating techniques for systematic outlier rejection in these and other difficult cases. However, we currently have a few heuristics in place which have resulted in noticeable improvements.

2.1.2 Heuristics

In the current version of the ESP testbed, the vision component does no intelligent pruning of FCAT based on prior knowledge of spacecraft position. We have greatly increased the likelihood of correct matches in the LMT by introducing a simple pruning technique that processes the LMT twice. In the first pass, we compute matches and then average the 3D vectors describing these matched landmarks. This defines a principal direction. We then prune the full catalog so that the inner product of any vector with this principal direction is positive. Let

$$\begin{aligned} \bar{v} &= \text{mean}(\{v \in \text{FCAT} : v \in \text{LMT}\}) \\ \text{FCAT}_2 &= \{v \in \text{FCAT} : v \cdot \bar{v} > 0\} \end{aligned}$$

In the shorthand notation above, v refers only to the 3D vector associated with landmarks as expressed in the target coordinate frame, not the full contents of the FCAT or LMT associated with a given landmark such as covariance data and descriptor information. A second pass computes a new LMT from FCAT_2 only. This effectively restricts the catalog search to the portion of the body facing the spacecraft. Even this minimal constraint leads to significant improvements. First, spurious matches on the other side of the body are ignored. Second, those matches which fail to pass the uniqueness criterion for entry in the LMT (see [1]) because of similar features on the far side of the body are now more likely to match correctly.

Another simple heuristic, which is useful in the case of only 1 or 2 matches, is to admit any landmark with a very strong match. In other words, if the descriptor extracted from an orbital

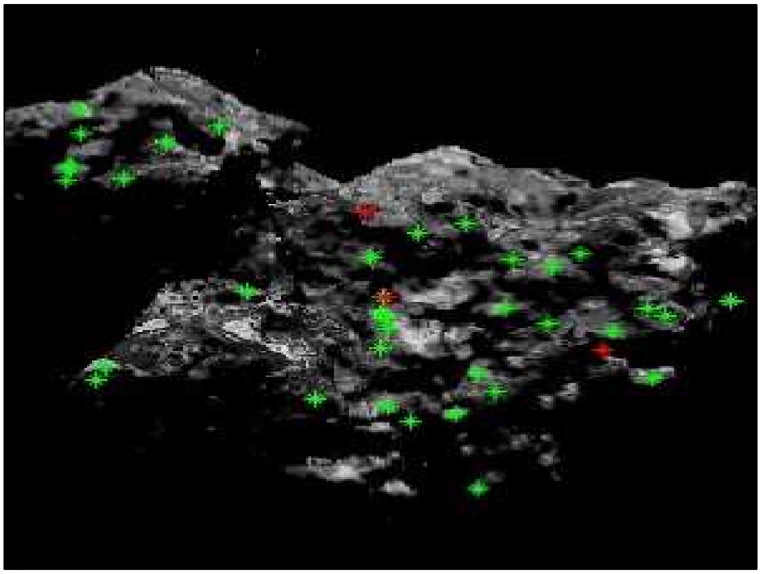

Figure 2: Example of outlier rejection on one frame, using synthetic imagery in ESP testbed. Valid matches to the catalog are shown in green, rejected outliers in red.

image matches a catalog descriptor to within some tolerance (variable, depending on the body), the match is always accepted.

In Fig. 2 we show the effect of applying the techniques outlined above. Those landmarks which are rejected as outliers are shown in red, while those which are accepted as valid are shown in green.

2.1.3 Homography and RANSAC

For direct frame-to-frame comparisons, we have also implemented a simple homography-based outlier rejection scheme similar to the pose estimation scheme outlined in §2.1.1. This assumes that two images are related by a plane homography so that if q_1 is any point in the first image expressed in homogeneous coordinates, there exists a 3×3 matrix H such that the corresponding point q_2 in the second image satisfies

$$q_2 = H \cdot q_1 \tag{3}$$

We explain Eq. 3 in a little more detail. Let P be a plane in $\mathbf{R}^3$ spanned by vectors $\{\mathbf{v}, \mathbf{w}\}$ expressed in some global coordinate frame. $\{\mathbf{v}, \mathbf{w}\}$ represents an explicit choice of basis for P, but the following discussion is independent of the particular choice. Suppose a camera images the plane from two different positions given by $\{(R_i, T_i) \mid i = 1, 2\}$. Any point on $p \in P$ can be expressed as

$$p = \alpha\mathbf{v} + \beta\mathbf{w} \tag{4}$$

in the global frame. The homogeneous image coordinates q_i of p in each of the two camera frames is given by

$$q_i = A[R_i \cdot p + T_i] \tag{5}$$

following Eq. 1. If we now substitute Eq. 4 into Eq. 5 and simplify, we obtain

$$\begin{aligned} q_i &= A[R_i \cdot (\alpha\mathbf{v} + \beta\mathbf{w}) + T_i] \\ &= A[R_i \cdot \mathbf{v} \quad R_i \cdot \mathbf{w} \quad T_i] \cdot [\alpha \quad \beta \quad 1]^T \\ &= \hat{H}_i \cdot [\alpha \quad \beta \quad 1]^T \end{aligned} \tag{6}$$

One can show that if T_i is spanned by $\{R_i \cdot \mathbf{v}, R_i \cdot \mathbf{w}\}$ the plane P contains the image center of the camera in position i. If we ignore this singular case, $\hat{H}_i$ has full rank and is invertible. It then follows from Eq. 6 that in homogeneous coordinates

$$\begin{aligned} q_2 &= \hat{H}_2 \cdot [\alpha \quad \beta \quad 1]^T \\ &= \hat{H}_2 \hat{H}_1^{-1} \cdot q_1 \\ &= H \cdot q_1 \end{aligned}$$

Note that the above discussion holds strictly only if all points lie on a plane in space. However, in many situations a near-planar assumption is adequate. For outlier rejection we proceed as with the pose estimation framework replacing Eq. 1 with Eq. 3.

2.2 Enhancement to Scale Invariance

We now describe a change to the core implementation of the feature detector. This has given us much better invariance to changes in scale than the previous version of the algorithm. We briefly summarize the relevant portion of the previous approach.

Given an image, the algorithm extracts salient features at different scales. In this context, “scale” refers to the portion of the frequency spectrum occupied by the feature. In other words, a filter tuned to the appropriate frequency will have a high response at the given feature. This is accomplished by constructing a stack of bandpass filtered copies of the image and finding extrema in the stack. See [1] for details. In the past, we organized this stack into a collection of octaves, each containing multiple scales. Each octave contained data with frequency content starting at roughly half that of data at the same relative scale in the previous octave. This organizational scheme followed the description in a pre-print version of [2]. It is intended to optimize the scale

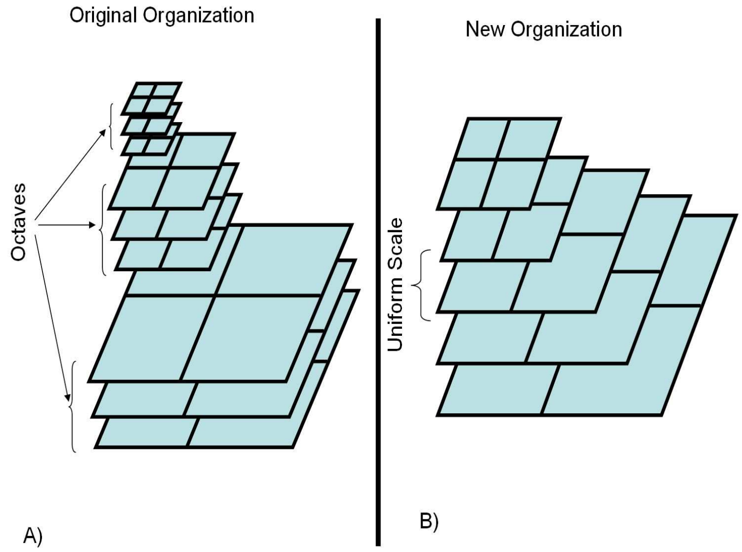

Figure 3: A) The scale/octave organization used in earlier versions of the feature detection algorithm. B) The uniform scale spacing used in current version. The new organization sacrifices some efficiency for greater versatility.

calculation by requiring explicit computation of only one octave in the scale-space followed by successive resampling to obtain other octaves.

We have found that at little expense to computational cost (see §2.3), we are able to produce much greater invariance to scale by modifying this scheme. Instead of organizing in octaves and scales, we create a simple image pyramid with uniform distribution in scale. We illustrate the difference in Fig. 3.

We now present the new scheme in greater detail. Let $I(x, y)$ represent the original intensity image, where the function I returns the gray value of the image at pixel coordinates (x, y). Let f_{scale} be a scale factor controlling the image resampling between successive layers of the pyramid, which we will call I_n. In other words, if dim$(I_0 = I) =$ rows × cols is the original image size, then

$$\text{dim}(I_n) = \frac{\text{rows}}{f_{scale}^n} \times \frac{\text{cols}}{f_{scale}^n}$$

Let $G(x, y, \sigma)$ be the 2D Gaussian kernel defined by

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp(-\frac{x^2 + y^2}{2\sigma^2})$$

The equivalent to the Difference of Gaussians (DoG) space as defined in [1], i.e. the bandpass filtered stack resulting from the original image, is given by

$$D_n(x,y) = I_n(x,y) * \{G(x,y,\sigma \cdot f_{scale}) - G(x,y,\sigma)\} \tag{7}$$

where the $*$ operator represents convolution. The $\{D_n\}$ are pictured in part B. of Fig. 3. In parallel, we also compute and save $\{D_n^-\}$ and $\{D_n^+\}$ defined as

$$\begin{aligned} D_{n+1}^-(x,y) &= I_n(x,y) * [G(x,y,\sigma \cdot f_{scale}^2) - G(x,y,\sigma \cdot f_{scale})] \\ D_{n-1}^+(x,y) &= I_n(x,y) * [G(x,y,\sigma) - G(x,y,\frac{\sigma}{f_{scale}})] \end{aligned}$$

Note that because of the combination of image rescaling and the change in the width of the Gaussian kernels, D_{n-1}^+ has the same frequency content as D_{n-1} but the same image dimensions as D_n. Similarly, D_{n+1}^- has the frequency content of D_{n+1} but also has the dimensions of D_n. This allows us to easily identify salient points in the DoG stack by directly comparing images of identical size but separated uniformly in frequency. In other words, a point $(x_o, y_o) \in D_n$ is a candidate for a feature provided

$$(x_o, y_o) = \underset{||x-x_o||\leq 1, ||y-yo||\leq 1}{\text{argmax}} (\max(D_n(x,y), D_{n-1}^+(x,y), D_{n+1}^-(x,y)))$$

or

$$(x_o, y_o) = \underset{||x-x_o||\leq 1, ||y-yo||\leq 1}{\text{argmin}} (\min(D_n(x,y), D_{n-1}^+(x,y), D_{n+1}^-(x,y)))$$

We demonstrate the effectiveness of this new scheme using imagery from the Deep Impact mission. In Fig. 4 can be seen four images of the Tempel 1 comet taken by the Impactor imager at widely varying distances. Using the previous version of the feature algorithm, we were unable to find enough valid matches between frames to compute image transforms and register one frame to another. With the new version of the algorithm, we can match over scale changes exceeding a factor of 2. Note that the registration shown in E), F) and G) of Fig. 4 uses matched landmarks directly and is completely automatic; there is no human interaction or guidance involved. The match between successive frames is illustrated in red. Suppose for any image pair, the first frame is called I_a and the second is called I_b. The feature matching algorithm produces discrete sets of matched features $\{q_i^a \in I_a : i = 1..n\}$ and $\{q_i^b \in I_b : i = 1..n\}$ expressed in homogeneous coordinates. Image warp is then accomplished by computing a plane homography H_a^b such that

$$\varepsilon = \sum_{i=1}^{n} ||H_a^b \cdot q_i^a - q_i^b||$$

is minimized. Then for all points $\{p \in I_b\}$, we compute the coordinates $\{H^{-1}p\}$ and superimpose the result on I_a in red. The computed homography directly indicates the relative scale change

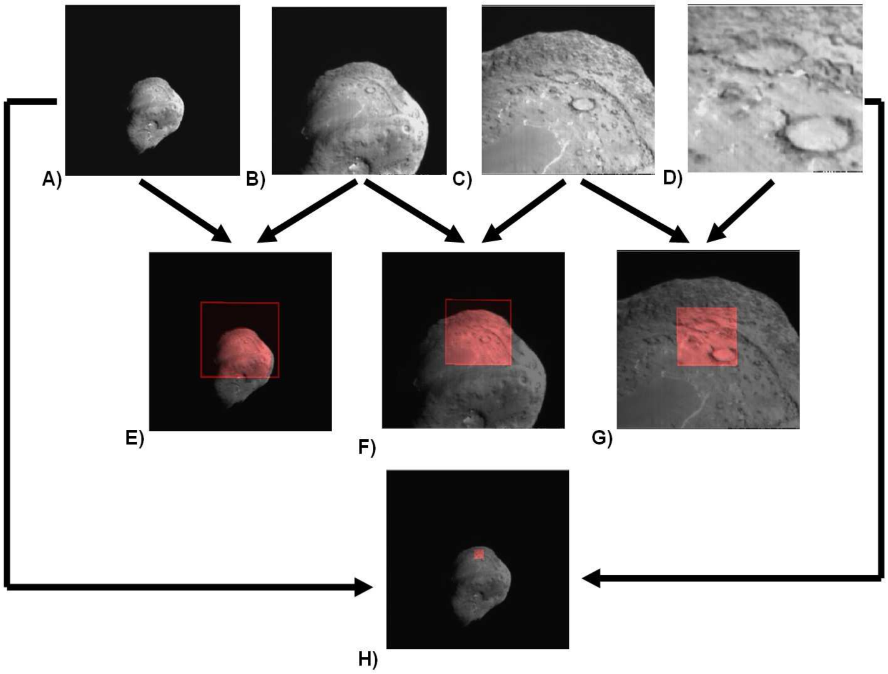

Figure 4: A)-D) Four images of Tempel 1 acquired by the Deep Impact impactor. E) Automatic registration of first image pair using feature algorithm. F) Automatic registration of second image pair. G) Automatic registration of third image pair. H) Registration of 1st and 4th images by concatenation of pairwise transforms.

between frames. This is a factor of ~ 2 for the first to second frame and ~ 3 for the second to third and third to fourth frames. Suppose that the three pairwise homographies are labeled H_a^b, H_b^c and H_c^d. Then we compute a joint homography across all three image pairs by

$$H_a^d = H_a^b \cdot H_b^c \cdot H_c^d$$

In H) of Fig. 4, we show the results of using H_a^d to register the first and last of the four frames. This shows a correct registration over a scale factor approaching 20.

This has important implications for FCAT. It indicates that we can construct a chain of matches to identify landmarks from very close imagery (e.g. during descent) to counterparts on a distant scale, allowing us to maintain global context even during close approach.

2.2.1 Illumination Invariance

In the previous report, we showed a number of simple techniques to enhance illumination invariance. The basic idea was to perform an approximation of a highpass filter by doing background subtraction prior to computing features. Thus, if I is the original intensity image, we compute $J = I - (I * B)$ for B_n an averaging box filter of size $n \times n$. We have since concluded that even this simple scheme introduces problems in scale invariance. Assuming the inherent scale of objects in the scene has changed, using B_n of the same size in both images has the effect of shifting the DoG pyramids in Fig. 3 in a way that fails to reflect scene content. The ideal scheme is as follows: If objects in I_b are scaled by a factor of f with respect to their counterparts in I_a. We should use

$$J_a = I_a - (I_a * B_n)$$
$$J_b = I_b - (I_b * B_{n \cdot f})$$

This implies some knowledge of the factor f. While we cannot assume that f is known with accuracy, even a rough approximation from altimetry or state information is adequate. As an example, consider frames A) and B) of Fig. 4. In the earlier fixed box filter approach we found a total of 8 matches after filtering, one of which was false. In the new scheme, the filter size used for the first image is chosen to be approximately one third the size used for the second. The actual scale change as determined from the homography transform computed between the two frames is 2.28. With this modification, we obtain 26 matches, all of which are correct.

2.3 Speedup

The feature detection algorithm can be divided into four stages:

- Stage 1: Scale-space generation
- Stage 2: Interest point detection and filtering
- Stage 3: Feature vector generation
- Stage 4: Feature matching

Although our current ESP testbed is primarily for proof of concept and is written mostly in Matlab, we have, nevertheless, made a number of modifications to decrease computation time. The most significant impact has been through information reuse. Earlier versions of the algorithm performed a number of redundant gradient computations during both the interest point detection and feature vector generation stages. Some additional allocation of processing time and memory during the early scale-space generation stage has greatly reduced computational time for these later, more expensive stages. The final feature matching stage has been sped up by a factor of 10 by rewriting the most time consuming part of the code in C with a MEX interface to Matlab. Components of the gradient computation and image rescaling also now use MEX. Since the two versions of the

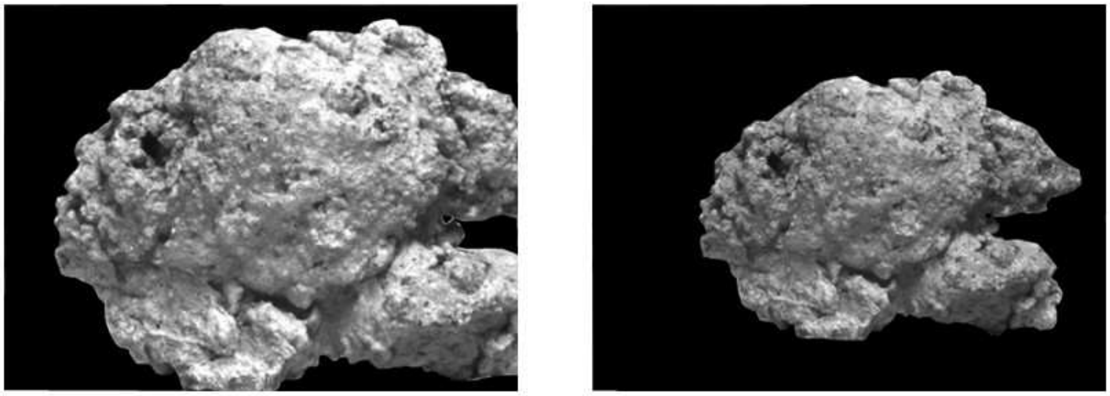

Figure 5: Imagery used for test results recorded in Table 1. The right image is approximately 30% smaller than the left.

algorithm produce different numbers of features and matches, runtime comparison must be done with caution. We show results for representative run of the two algorithms on the 512×384 resolution imagery shown in Fig. 5. Note that there is a moderate 30% scale change between frames. We compare the scale-space generation times directly, but for both stages 2 and 3 above, we divide total time by the number of detected features. For stage 4, we divide total time by the product of the number of features detected in each of the two images to give a time per comparison for match evaluation. Finally, we record the total number of matches and the total runtime per match. These results are recorded in Table 1. The dramatic increase in the total number of matched features is entirely due to the algorithmic changes in §2.2. The more relevant numbers for evaluating speedup are the total time/match and time/feature vector. These processes have been sped up significantly.

	Old Implementation	New Implementation
Stage 1 time	3.15 s	5.60 s
Stage 2 time/feature	3.16×10^{-3} s	1.99×10^{-3} s
Stage 3 time/feature	9.06×10^{-3} s	5.84×10^{-4} s
Stage 4 time/comparison	9.37×10^{-6} s	9.32×10^{-7} s
Features matched	36	341
Total time/match	0.95 s	0.07 s

Table 1: Runtimes for old and new algorithm implementations for a sample feature pair shown in Fig. 5 with moderate scale change.

3 Twice Around Study

The computer vision component of the Twice Around Study consists of 3 primary contributions. These are (1) catalog generation, (2) landmark identification and (3) frame-to-frame tracking. These three components results in the FCAT, LMT and PFT, respectively. We have reported in [4] recent advances in generating the feature catalog from Bundle Adjustment, in the absence of trajectory information. However, for the purposes of this report, we present a trajectory based solution used this year for the majority of our tests.

3.1 Catalog Generation

The current version of the twice around study begins with an initial orbit with known time history and target to camera transformations. Rendering software generates the camera view during orbit. The current version of the rendering drapes a simple image over a 3D model of the target. Work is currently underway to produce more photo-realistic model based renderings [8].

We populate FCAT by computing feature matches across successive image pairs or triplets. Spurious matches are eliminated using the epipolar constraint (See [1]) in the case of pairwise matching and the trifocal constraint in the case of image triplets. Like the epipolar constraint, the trifocal constraint is a linear set of relationships between 3 images of a static scene encoded in the so-called trifocal tensor. Details can be found in [9]. This formulation, like the fundamental matrix in the epipolar case, does not require explicit computation of camera pose or 3D structure. It provides a direct pixel level constraint on point correspondences across 3 frames.

Our early implementation of FCAT used the epipolar constraint. The latest version uses the trifocal constraint for greater stability of landmarks across viewpoints. In other words, a landmark which is seen in at least 3 frames and satisfies the trifocal constraint is (1) less like to be a false match and (2) more likely to remain stable over multiple viewpoints. The end result is fewer but better landmarks in FCAT. An example run using ~ 200 images in a single orbit around a model of the asteroid Itokawa produced a catalog with 3050 entries with the epipolar constraint and 1367 entries with the trifocal constraint.

Once a match is made across two or three images, we use stereo triangulation techniques to localize the point in space. The 3D data is recorded along with the feature descriptors in FCAT. A second pass through the just generated FCAT prunes any duplicate entries. Duplicates are evaluated by their proximity in 3D. Currently, any two landmarks within 3 meters of one another are considered redundant. This threshold is a function of target size and image resolution. Imagery of several frames of an orbit of the Itokawa model and the associated pruned FCAT are shown in Fig. 6.

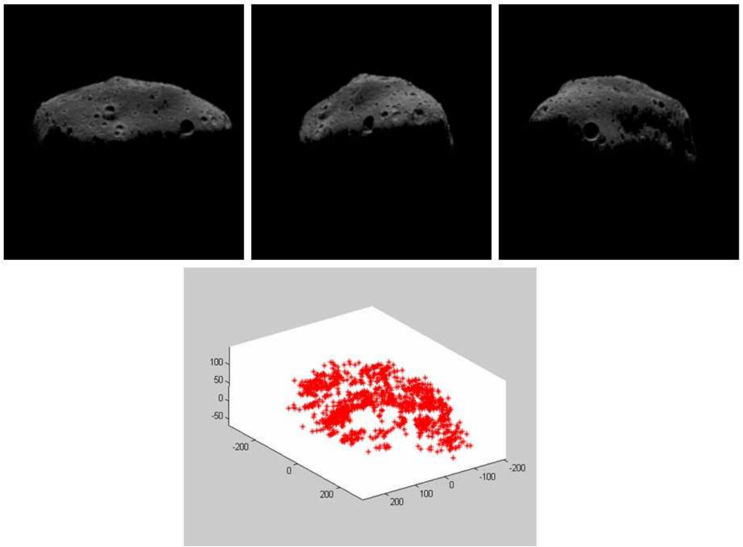

Figure 6: Three rendered frames from orbit of Itokawa model shown above. Reconstructed points in FCAT shown below.

Naturally, the quality of the 3D estimate of any point depends on the feature localization in the image as well as on the relative camera and scene geometry. This information is essential to the state estimation portion of the ESP testbed. In order to capture this, we compute an explicit covariance.

3.1.1 Covariance estimates in FCAT

The 3D triangulation error in generation of the catalog is a function of the camera motion, the intrinsic camera parameters, and the image plane match error. Since the trajectory is known and the camera is assumed calibrated, the first two error sources do not contribute. We consider the match error.

Suppose a coordinate frame is attached to the camera with $\hat{Z}$ along the optical axis and $\hat{X}$ and $\hat{Y}$ on the image plane. Suppose further that we have match errors between the first and second frame of magnitude Δx and Δy pixels in the image plane with $\Delta p = \sqrt{(\Delta x)^2 + (\Delta y)^2}$.

The dependence of stereo triangulation on these errors is well understood. See [10] for an overview. Assuming the camera has approximately the same viewing direction $\hat{Z}$ in consecutive images, a generally valid assumption over two or three frames, the stereo error in this direction is related to the image plane match error Δp by

$$\Delta Z = \frac{Z^2}{f \cdot B} \Delta p \tag{8}$$

where the camera focal length in pixels is f, the baseline of the motion is B in meters and the distance to the 3D target point is Z. If Δp is interpreted as a standard deviation, then ΔZ is the standard deviation in recovered distance along the viewing direction. The lateral errors ΔX and ΔY in triangulation assuming match errors of Δx and Δy in the image plane are given by

$$\begin{aligned} \Delta X &= \frac{X}{f} \Delta x \\ \Delta Y &= \frac{Y}{f} \Delta y \end{aligned} \tag{9}$$

In the trifocal case, we average the quantities in Eqs. 8 and 9 over all three stereo pairs arising from the 3 camera positions. In the frame of the camera, the error covariance can be approximated by

$$C_{camera} = \begin{pmatrix} (\Delta X)^2 & 0 & 0 \\ 0 & (\Delta Y)^2 & 0 \\ 0 & 0 & (\Delta Z)^2 \end{pmatrix}$$

Let R be the rotation matrix relating the average over all camera frames in the pair or triplet used for triangulation to the body frame of the target. Then

$$C_{target} = R^T C_{camera} R$$

What remains to be determined are the quantities Δx, Δy and Δp. In standard stereo methods, these quantities can be determined from the subpixel approximation in the correlator. However, in our case, we match only points in descriptor space and rely wholly on the localization accuracy of the feature detection algorithm. There is no direct image based match. Thus, the match error is not easily quantified. However, we have determined empirically that $\Delta x = \Delta y = 0.5$ pixel is a reasonable figure, and our estimates are currently based on these numbers. More systematic techniques for evaluating this match error are under consideration. One candidate method is to use a directly measured triangulation error on a known 3D structure to infer the associated match error.

3.2 Landmark Detection

Once the catalog has been produced, we generate modified orbits with known trajectories around the same body and render new images. For each frame in the new orbit sequence, we find features and match to FCAT. This data is recorded in the LMT and passed to the state estimator. For each

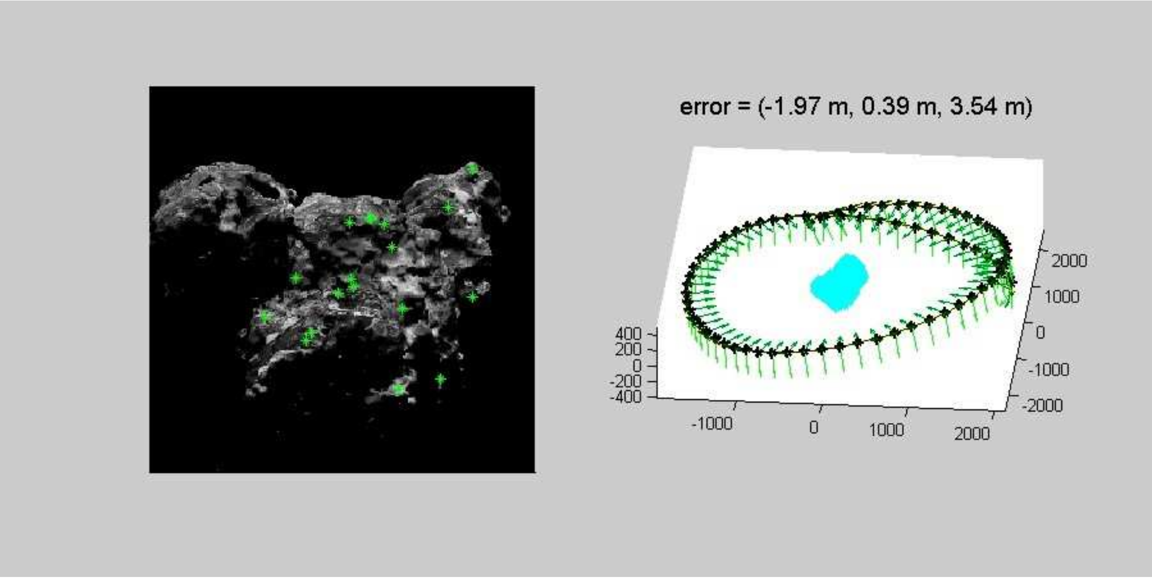

Figure 7: On the left is the final frame of the second pass in an example run of the twice around study showing correctly matched features. On the right is shown the ground truth trajectory as a moving frame in green, and the vision based camera localization from the LMT as black dots. The error is uniformly bounded by ~ 10 m for this 2 km orbit.

matched landmark, the LMT records the 3D position of the landmark from FCAT, bearing angles in the image frame to the landmark along with covariance estimates, and the full landmark descriptor.

The details of the match mechanism are covered in [1] and modifications to the basic method to enhance robustness were described above in §2.1. We show in Fig. 7 the results from a representative run of the twice around study using the body also pictured in Fig. 2. The original catalog was produced from a sequence of 200 images in one complete near-circular orbit of the ~ 500 m body at a distance of ~ 2 km from the center of mass. In the second pass, we took a similar but modified orbit and matched landmarks against 70 frames. Fig. 7 shows the ground truth orbit as a moving coordinate frame in green and the recovered camera position as a series of black dots. Note that these are single-frame position estimates using vision based methods only. We do not record the integrated result of the estimator. Our only purpose is to verify the quality of matches in the LMT. We computed the error covariance to find that the 1σ error in the direction of largest uncertainty was 5.7 m. The absolute position error (norm of 3D error) was uniformly bounded by 23 m and had an RMS value of 8 m.

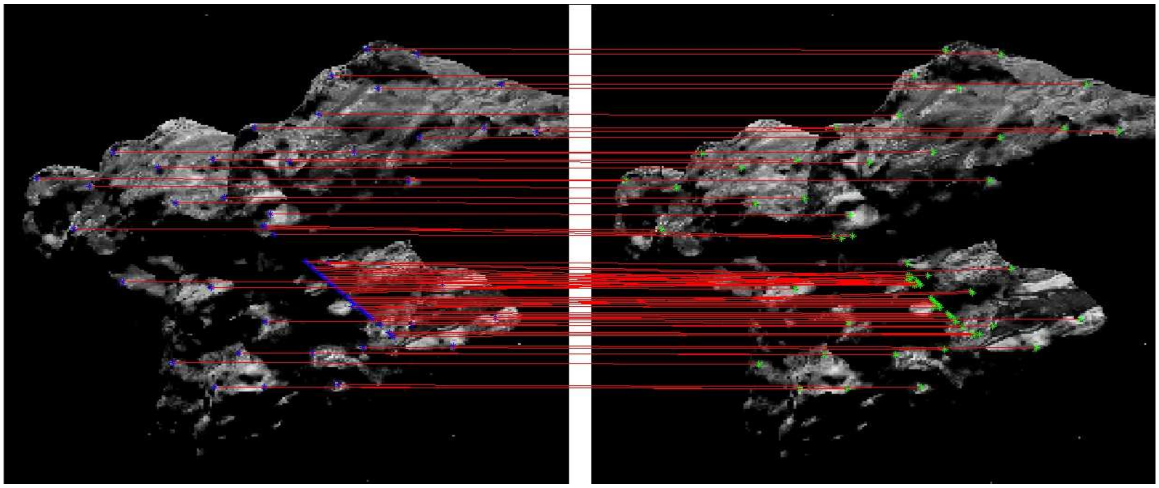

Figure 8: Frame-to-frame matches recorded in the PFT. Detected interest points on in the first image are shown in blue. Matched points from correlation are shown in green on the second image. The correspondence is indicated by red lines connecting matching points.

3.3 Frame-to-Frame Matching: PFT

We now describe our implementation of the PFT or Paired Feature Table, which records frame to frame matches. These are not necessarily landmarks, but salient image points which are expected to remain stable only over a short duration (e.g. between successive frames). The approach taken is to identify interest points using the Harris operator [11], a standard corner detector well established in computer vision. Details were also reviewed in a slightly different context in [1].

Once a set of interest points are detected, we attempt to match these in the second frame using the normalized cross-correlation borrowed from standard stereo vision techniques. After integer correlation, we perform a subpixel refinement using correlation scores from the 3×3 neighborhood of the best match. Suppose interest point p in the first frame matches $q = (x_o, y_o)$ in the second after correlation. Let $C_p(x, y)$ be the correlation score between p in the first frame and (x, y) in the second. We find the quadratic S which best approximates C_p in a 3×3 neighborhood of q by minimizing

$$\varepsilon = \sum_{x=x_o-1}^{x_o+1} \sum_{y=y_o-1}^{y_o+1} (S(x,y) - C_p(x,y))^2$$

Then the subpixel approximation is the point $\tilde{q} = (\tilde{x}_o, \tilde{y}_o)$ such that $S(\tilde{x}_o, \tilde{y}_o)$ is a local minimum. An example of the frame to frame match recorded in the PFT is shown in Fig. 8.

4 Conclusion and Future Work

We have presented in this report a number of enhancements and additions to the feature detection work reported last year. Much of this effort has been driven by the needs of the Twice Around Study. For example, robustness became a prime issue which we have attempted to address throughout this year. Various other implementation details required change or update to earlier methods. We have also greatly enhanced the scale invariance exhibited by the early version of the feature algorithm.

From the vision perspective, the most significant challenges for next year will involve extending the regime of applicability of the feature algorithms. We intended to test more widely varying orbits and changes in illumination. The latter has not yet been adequately addressed, but we are investigating work in the literature on simple parametrized lighting models. We are also continuing to explore techniques for outlier rejection to accommodate the more challenging tests we anticipate in the next year, should the task be funded.

5 Acknowledgments

This research was performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration and funded through the internal Research and Technology Development program.

References

[1] A. Ansar, “2004 small body GN&C research report: Feature recognition algorithms,” Tech. Rep. D-30714, Jet Propulsion Laboratory Internal Document, 2004.

[2] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[3] Y. Cheng, A. Johnson, and L. Matthies, “MER-DIMES: A planetary landing application of computer vision,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (San Diego, California), Jun 2005.

[4] Y. Cheng and A. Ansar, “2005 small body GN&C research report: Techniques for bision-based generation of landmark catalogs for navigation,” Tech. Rep. D-32944, Jet Propulsion Laboratory Internal Document, 2005.

[5] M. Fischler and R. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, pp. 381–395, 1981.

[6] A. Ansar and K. Daniilidis, “Linear pose estimation from points or lines.,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 578–589, 2003.

[7] C.-P. Lu, G. D. Hager, and E. Mjolsness, “Fast and globally convergent pose estimation from video images.,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 6, pp. 610–622, 2000.

[8] L. Phan, “2005 small body GN&C research report: Small body gn&c testbed (sgt),” Tech. Rep. D-32945, Jet Propulsion Laboratory Internal Document, 2005.

[9] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[10] W. Kim, A. Ansar, and R. Steele, “Stereo vision performance analysis and an application to multi-view target registration,” in *IEEE Conf. on Systems, Man and Cybernetics*, (Hawaii), 2005.

[11] C. Harris and M. Stephens, “A combined corner and edge detector,” in *Fourth Alvey Vision Conference*, (Manchester, UK), pp. 147–151, 1988.