Check for updates

Robust Model-Based Monocular Pose Initialization for Noncooperative Spacecraft Rendezvous

Sumant Sharma*
Stanford University, Stanford, California 94305
Jacopo Ventura†
Technische Universität München, 85748 Garching, Germany
and
Simone D'Amico‡
Stanford University, Stanford, California 94305

DOI: 10.2514/1.A34124

This work addresses the design and validation of a robust monocular vision-based pose initialization architecture for close-range onorbit-servicing and formation-flying applications. The aim is to rapidly determine the pose of a passive space resident object using its known three-dimensional wireframe model and a single low-resolution two-dimensional image collected on board the servicer spacecraft. In contrast to previous works, the proposed architecture is onboard executable and capable of estimating the pose of the client without the use of fiducial markers and without any a priori range measurements or state information. A novel feature detection method based on the filtering of the weak image gradients is used to identify the true edges of the client in the image, even in presence of the Earth in background. The detected features are synthesized using simple geometric constraints to dramatically reduce the search space of the feature correspondence problem, which is solved using the EP*n*P method. This approach is proven to be an order of magnitude faster than the state-of-the-art random sample consensus methods. A fast Newton–Raphson method that minimizes the fit error between corresponding image and model features is employed to refine the pose estimate and to resolve pose ambiguity. The proposed methodology is tested using actual space imagery collected during the PRISMA mission at about a 700 km altitude and a 10 m interspacecraft separation.

Downloaded by STANFORD UNIVERSITY on July 2, 2018 | http://arc.aiaa.org | DOI: 10.2514/1.A34124

I. Introduction

THE onboard determination of the pose (i.e., the relative position and attitude) of a noncooperative client spacecraft using a monocular camera is a key-enabling technology for future onorbiting servicing and debris removal missions such as PROBA-3 by ESA [1], autonomous nanosatellite guardian for evaluating local space (ANGELS) by the U.S. Air Force [2], and Restore-L by NASA [3]. The knowledge of the current pose of the client spacecraft during proximity operations allows real-time generation of the full-degree-of-freedom approach trajectory and control update [4]. This aspect is crucial in noncooperative maneuvers because the kinematic characteristics of the client are unknown before the launch of the mission; therefore, the rendezvous and docking trajectory must be generated on board using the current state estimates. In contrast to other state-of-the-art systems based on light detection and ranging (LIDAR) and stereo camera sensors, monocular navigation ensures rapid pose determination under low power and mass requirements [5]. Therefore, monocular pose determination is also the natural candidate as a navigation system in small spacecraft such as CubeSats for future formation flights missions [6,7].

Due to these advantages, navigation systems employing monocular cameras have been proposed in order to enable rapid pose estimation and tracking in close range: up to a few centimeters to the target using limited mass and power resources [8–15]. Typically, these systems employ an image processing subsystem that identifies the visible target's features in the monocular image followed by a dedicated pose solver and an extended Kalman filter. This routine is executed in a closed loop for pose tracking during the rendezvous maneuver. In general, the pose solver is an iterative algorithm that minimizes a certain fit error between the features detected in the image and the corresponding features of a known reference three-dimensional (3-D) model. All of the aforementioned navigation systems also require a priori information of the pose to kick start the pose estimation and tracking. In the presence of a poor initialization, these systems may converge to a local minimum, resulting in incorrect pose solutions. Existing work on pose initialization for spaceborne applications aims to directly apply terrestrial robotic navigation algorithms to the space imagery [16–18]. However, these techniques are 1) too computationally expensive to implement on board CubeSats, 2) require extensive offline training, or 3) require the presence of known fiducial markers on the client spacecraft.

The problem of determining the pose of an object using a monocular image and the object's 3-D model has been extensively addressed in the research field of computer vision, typically for terrestrial applications. One of the first solutions to this problem is due to Dhome et al. [19], who proposed a closed-form solution to solve for the pose, given the correspondences between the edges detected in the image and the line segments of the 3-D model. To solve for the correct correspondences, it follows an exhaustive predictive and verification step by matching all possible sets of three 3-D model line segments with three two-dimensional (2-D) image edges. To avoid an exhaustive search for the feature correspondences, authors have used soft assign [20,21]; however, its accuracy depends on manual tuning of parameters and it has been shown to be slower than random sample consensus (RANSAC) [22]. Following the work of Dhome et al. [19], numerous algorithms for finding the object's pose from the image and model feature correspondences have been proposed [23–27]. Although these algorithms do not require a priori pose information, they produce a correct pose solution only if a number of correct feature correspondences are provided. Notably, this aspect limits the real-time application of these algorithms for pose initialization.

In regard to solving the problem of pose initialization for spacecraft applications, Kanani et al. [18] and Petit et al. [28]

Received 23 October 2017; revision received 12 March 2018; accepted for publication 18 March 2018; published online 27 June 2018. Copyright © 2018 by Sumant Sharma, Jacopo Ventura, and Simone D'Amico. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission. All requests for copying and permission to reprint should be submitted to CCC at www.copyright.com; employ the ISSN 0022-4650 (print) or 1533-6794 (online) to initiate your request. See also AIAA Rights and Permissions www.aiaa.org/randp.

*Ph.D. Student, Department of Aeronautics & Astronautics, 496 Lomita Mall.

†Research Engineer, German Space Operations Center, Space Flight Technology, Münchner Strasse 20.

‡Assistant Professor, Department of Aeronautics & Astronautics, 496 Lomita Mall.

Downloaded by STANFORD UNIVERSITY on July 2, 2018 | http://arc.aiaa.org | DOI: 10.2514/1.A34124

presented a pose initialization architecture based on a dedicated offline learning approach where a hierarchical model view graph using prototype views of the 3-D model was built. The view graph was then explored during the online phase to find the prototype view for which the contour corresponds the most with the detected contour in the image. This approach is inspired from template matching approaches where the input monocular image is searched for specific features and/or image sections that can be matched to an assigned template [29]. However, in the absence of a large database of precomputed renderings of the target, these approaches tend to be very limiting because small changes in the target orientation and position may greatly affect its appearance. Grompone [10] proposed an initialization scheme that employed the Harris algorithm [30] for feature detection, together with a linear eight-point algorithm to solve for the pose. However, the scheme only provided relative distance information based on background subtraction and Gaussian blob detection algorithms. Tests on the actual images of the Soyuz and Orbital Express spacecraft showed that the methodology was capable of correctly determining the region of interest (ROI) in the image. Finally, D'Amico et al. [31] proposed to solve the pose initialization problem through perceptual organization of the client's true edges detected in the image using the Sobel and Hough algorithms [32]. Preliminary tests on actual images from the PRISMA mission [33] showed the capability to determine the precise relative position and attitude of the target vehicle without a priori state information. However, tests also demonstrated two key limitations of this technique: first, the initial pose is dependent on a computationally expensive iterative view-space discretization not suitable for real-time execution; second, the image processing lacks robustness to illumination and background conditions.

In summary, the existing state-of-the-art methods show promising progress toward solving the problem of pose initialization for spaceborne applications; however, they suffer from two main challenges. First, the image background, the high reflectivity of the target's surface materials, and the absence of sunlight during the orbit pose significant problems in detecting the key features of a noncooperative spacecraft. The second challenge is posed by the large search space for the correct feature correspondences between the 3-D model and the image. An exhaustive prediction and verification procedure is too computationally expensive for spaceborne hardware, especially on smaller platforms such as CubeSats.

To address these challenges, this paper presents the design and validation of the Sharma–Ventura–D'Amico (SVD) architecture for robust monocular vision-based pose initialization. With respect to the state-of-the-art techniques, the proposed architecture is 1) robust to the background in the images; 2) requires no a priori knowledge of the target spacecraft's attitude and position; and 3) is computationally efficient, i.e., an accurate pose can be initialized within a few seconds. To ensure robustness to background, this paper introduces the weak gradient elimination (WGE) technique. WGE detects and eliminates the regions of weak gradient in the image and highlights the regions of the strongest gradients. In this way, only the strongest edges in the image are used for feature extraction using the Hough transform [32]. The extracted features are then organized into geometrically complex groups to dramatically reduce the search space of the feature correspondence problem. Possible hypotheses for the feature correspondence are used to efficiently solve the perspective-n-point (PnP) problem. The architecture is validated through actual space imagery collected during the PRISMA mission [33] at about a 700 km altitude and a 10 m interspacecraft separation.

The paper is organized as follows: Sec. II introduces the pose initialization problem and an overview of the proposed architecture. Sections III and IV describe the technical design details of the image processing and pose determination subsystems, respectively. Section V presents four tests that compare the proposed architecture against the state-of-the-art techniques for monocular vision-based pose initialization. These tests use synthetic imagery as well as imagery collected during the PRISMA mission. Section VI elaborates on the conclusions with a discussion on the applicability of this architecture in a monocular vision-based navigation system for onorbit servicing and formation flying missions.

II. Problem Statement and Architecture Overview

The schematic representation of the noncooperative pose initialization problem from a 2-D image and a 3-D model is shown in Fig. 1. In particular, the pose initialization problem consists of determining the position of the target's center of mass $\boldsymbol{t}^C$ and the orientation of the vehicle's principal axes R^{BC} with respect to the camera frame C. It is assumed that the 3-D model of the target is defined in the body-fixed coordinate system B, and it is aligned with the target's principal axes with its origin at the center of mass. The orientation is defined by the direction cosine matrix R^{BC} from the coordinate system of B to C.

Let $\boldsymbol{q}^B$ be a point of the 3-D model expressed in coordinate system B. By employing the standard pinhole camera model, the corresponding point $\boldsymbol{p} = [u, v]^T$ in the rectified image can be obtained using the 3-D/2-D true perspective projection equation:

$$\boldsymbol{r}^C = [x^C \quad y^C \quad z^C]^T = R^{\mathrm{BC}}\boldsymbol{q}^B + \boldsymbol{t}^C \tag{1}$$

$$\boldsymbol{p} = \left[\frac{x^C}{z^C}f_x + C_x, \frac{y^C}{z^C}f_y + C_y\right] \tag{2}$$

where $\boldsymbol{r}^C$ represents the point of the target spacecraft expressed in camera frame C according to the current pose: $\boldsymbol{t}^C$ and R^{BC}. Although f_x and f_y denote the focal lengths of the camera, (C_x, C_y) denotes the principal point of the image. Without loss of generality, it is assumed that direction C_3 is pointed along the boresight of the camera and that directions C_1 and C_2 are aligned with directions P_1 and P_2 of the image frame P, respectively. The unknown coefficients in the 3-D/2-D true perspective equations are the three components of the client position $\boldsymbol{t}^C$ and the three parameters that define the rotation matrix R^{BC} of the target orientation. To solve this system of equations, at least three image points and corresponding model points are required. However, at least six correspondences between the image and model points are required to obtain a unique solution with a general configuration of points [34].

Figure 2 illustrates the proposed architecture to solve for the initial pose of a noncooperative client. The key features of this architecture are 1) the fusion of the WGE technique with the Sobel edge detector and the Hough algorithm for feature detection; 2) the use of feature synthesis to reduce the search space for feature correspondences between the image and the 3-D model; and 3) the combination of the EPnP solver and the

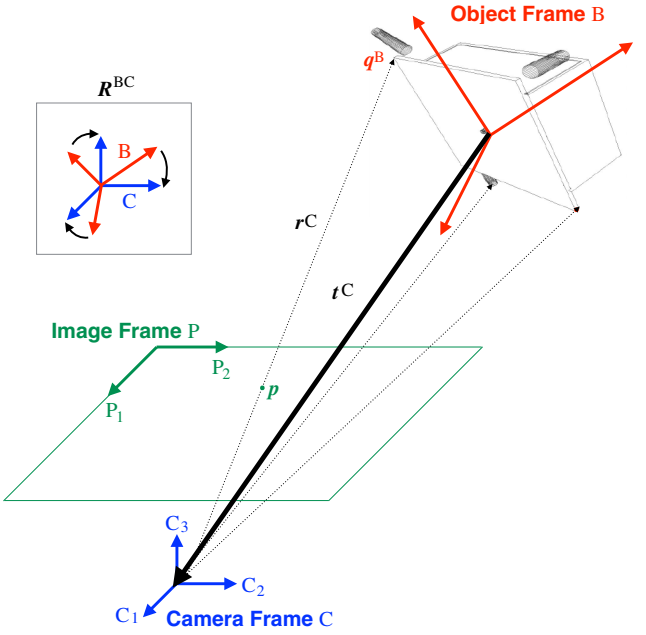


Fig. 1 Schematic representation of the pose estimation problem using a monocular image.

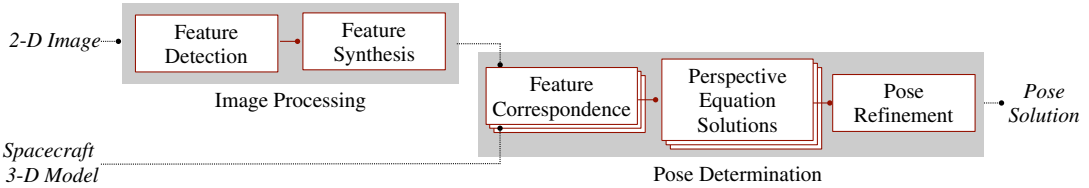


Fig. 2 Proposed pose initialization architecture with inputs of a single 2-D image, a 3-D model, and an output of the pose solution.

Newton–Raphson method for pose determination. The architecture consists of two subsystems:

1) The image processing subsystem accepts a single 2-D image as its input. It distinguishes the target spacecraft from the background, detects the target spacecraft's edge features, and then synthesizes them into geometric groups.

2) The pose determination subsystem accepts the feature groups detected in the image processing subsystem and the 3-D model as its input. It pairs the 2-D and 3-D geometric groups to create multiple correspondence hypotheses. For each hypothesis, the endpoints of the line segments forming the geometric groups are used in solving the 3-D/2-D perspective Eqs. (1) and (2). Of the multiple resulting pose solutions, the top five are iteratively refined and the best is selected using a reprojection error as the confidence metric.

With respect to the state-of-the-art image processing, the key innovation of this subsystem is the use of the WGE technique to efficiently distinguish the target spacecraft from the background. The same technique is used to boost the output of off-the-shelf edge detection techniques to provide a robust identification of the small as well as large edges of the spacecraft. Second, the key innovation in the pose determination subsystem is the use of geometrically complex feature groups detected in the image to solve the feature correspondence problem. This use of feature groups overcomes the challenge of pose ambiguity while still being more computationally efficient than the state-of-the-art methods such as the pose-space grid search or RANSAC. Lastly, the pose solution output is accompanied by its reprojection error, which denotes the quality of its fit with the detected image features. Therefore, if the reprojection error is calculated to be higher than a threshold, the solution can be classified as a "low-confidence solution." In case none of the features or a low number of features is detected in the image, a coarse relative position solution is output using the ROI detected by the WGE technique. In this manner, the SVD architecture can be used iteratively on a sequence of images until a "high-confidence solution" is available.

III. Image Processing

The goal of the image processing subsystem is to extract the most significant features of the target spacecraft in the input image. Subsequently, the extracted features are organized in geometrically complex groups in order to perform an efficient feature correspondences analysis for pose determination.

Pose initialization is assumed to be executed at the beginning of the close-range procedures, ideally before the pose tracking that leads to the capture of the client. Because the distance from the target is still considerable (30 m in the case of the PRISMA mission [33]), small details in the client's surface are not visible from the acquired image and only the most significant features such as the endpoints of the body edges, antennas, and solar panels can be distinguished. For this reason, the image processing subsystem solely focuses on extracting edges or straight line segments. Furthermore, as opposed to features based on color gradients, textures, and optical flow, edges are less sensitive to illumination changes [35].

Robust image processing is one of the main challenges in vision-based pose estimation. In fact, illumination conditions in space may even vary during a single orbit, and therefore may cause inaccurate and unreliable features detection. This is mainly due to the high-contrast illumination and low signal-to-noise ratio. Moreover, the potential presence of the Earth in the image background affects the image processing due to the presence of additional features of the planet's surface and the high reflectivity of oceans and clouds. State-of-the-art techniques for straight line segment detection such as the Canny edge detector [36] followed by the Hough transform [32] may be biased if applied directly to the image because these algorithms are gradient based and do not distinguish the foreground from the background. These methods also require the definition of numerous hyperparameters, which are difficult to tune for broad applicability because the imaged scene and the illumination conditions are constantly changing throughout the orbit. Finally, current methods for image segmentation [29] are computationally expensive, and therefore not suitable for onboard spacecraft applications.

To effectively and rapidly detect the client's edges, even in the presence of the Earth in the background, a hybrid image processing subsystem is proposed in Fig. 3. With respect to the off-the-shelf feature detection techniques, the key innovation of this subsystem is the introduction of the WGE technique and its fusion with the state-of-the-art edge detection techniques to provide an efficient and robust identification of the true edges of the spacecraft. As demonstrated in Sec. V, the WGE technique identifies a more accurate and robust ROI in the image as compared to the state-of-the-art techniques, such as maximally stable extremal regions (MSERs) [37]. The ROI detection makes the subsystem robust to the background, as well as allows for an automated selection of hyperparameters required for the Hough transform. Hence, the subsystem not only finds straight line segments corresponding to the large features of the spacecraft but it also detects straight line segments corresponding to smaller features such as antennas. Even though the current image processing subsystem relies on detecting straight line segments, it can be easily extended to include cylindrical, spherical, and circular features through the formulation of separate Hough transforms. The edges corresponding to these features can then be classified as additional feature groups within the proposed architecture. In the following sections, the proposed feature detection and feature synthesis procedures are described in detail.

A. Feature Detection

The feature detection procedure aims at identifying a robust set of line segments in the rectified input image corresponding to the most significant true edges of the client spacecraft. Some line segments correspond to the spacecraft's large features (such as the contour of bus and solar panels), whereas others correspond to small features such as antennas. In particular, small features are important for resolving geometry-related ambiguity in pose determination. This aspect will be discussed in more detail in Sec. IV.

Figure 3 shows the intermediate steps of the feature detection procedure. The input raw image is assumed to be rectified, i.e., corrected for lens distortion [38]. The Gaussian filter is applied to the input image in order to attenuate the image noise. The filtered image is then subjected to two parallel streams. In the first stream, the WGE technique is applied in order to distinguish the spacecraft from the background. A Hough transform is then applied to find smaller edges that can correspond to features such as the antennas. In the second stream, the Sobel algorithm is used for edge detection, followed by another Hough transform to extract long edges that can correspond to features such as the solar panels. The line segments obtained from these two streams are finally merged, whereas duplicates are discarded.

1. *Weak Gradient Elimination*

The weak gradient elimination technique was conceived to distinguish the target spacecraft in the foreground from the background. The size of the resulting ROI enabled the automated selection of hyperparameters required for the detection of both small

Downloaded by STANFORD UNIVERSITY on July 2, 2018 | http://arc.aiaa.org | DOI: 10.2514/1.A34124

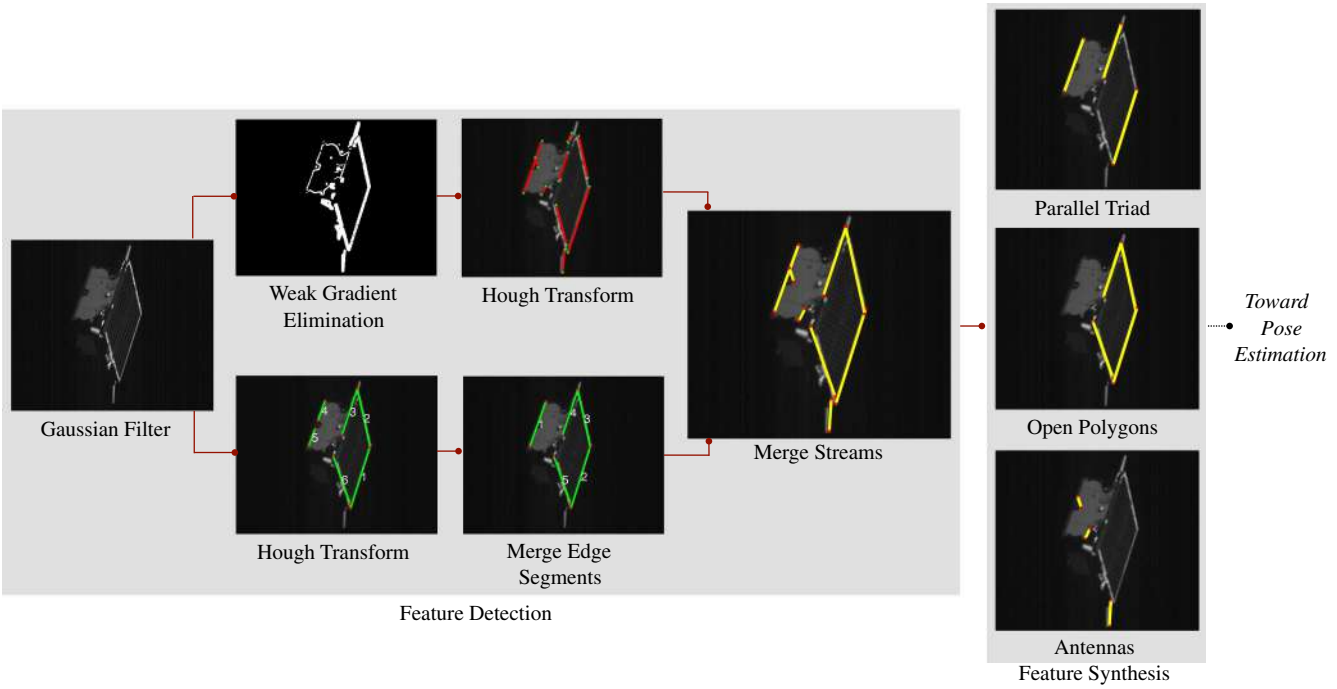


Fig. 3 Main steps of the image processing subsystem with a single 2-D image as the input and feature groups as the output.

and large features of the spacecraft. State-of-the-art algorithms based on the Hough transform rely on a single set of manually tuned hyperparameters, and therefore tend to either detect long line segments only (as these get more votes) or fuse the short line segments with one of the proximal longer line segments. Moreover, if the image has the planet in the background, line segments belonging to clouds or coastlines bias the output. The WGE technique solves both of these problems.

The first step of this technique is the calculation of the image gradient $\boldsymbol{G}(\boldsymbol{u}, \boldsymbol{v})$ at all pixel locations by using the Prewitt operator [39]. In particular, two 3×3 kernels are convolved with the original image to calculate approximations of the horizontal and vertical derivatives. If we define the input image as $\boldsymbol{A}$, the approximation of the horizontal derivative at each point in the image as $\boldsymbol{G}_x$, and the approximation of the vertical derivative at each point in the image as $\boldsymbol{G}_y$, then

$$\boldsymbol{G}_x = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} * \boldsymbol{A} \qquad \boldsymbol{G}_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{bmatrix} * \boldsymbol{A} \tag{3}$$

where $*$ denotes the two-dimensional convolution operation. At each point in the image $\boldsymbol{A}(u, v)$, the resulting gradient approximations can be combined to give the gradient magnitude using

$$\boldsymbol{G}(u, v) = \sqrt{\boldsymbol{G}_x^2(u, v) + \boldsymbol{G}_y^2(u, v)} \tag{4}$$

Figure 4a shows the gradient of a test image that has the Earth as the background. Due to the presence of the planet, it is difficult to distinguish the spacecraft's true edges from the background clearly. To detect the spacecraft, the histogram of the gradient is obtained by sorting $\boldsymbol{G}(\boldsymbol{u}, \boldsymbol{v})$ in 100 uniformly spaced bins. As can be observed in Fig. 4b, most of the gradient intensities are weak and correspond to the features in the background or on the spacecraft's surface. The obtained histogram can be approximated by an exponential probability distribution function (PDF), as in Fig. 4b. The weak gradient pixel locations are then classified by thresholding the PDF fit to the gradient histogram. More precisely, the bin corresponding to the cumulative distribution of 0.99 is found by calculating the area under the curve of the PDF. All pixel locations corresponding to bins below this are classified as "weak" and their gradient value is set to zero. Figure 4c shows the result of this technique; the pixel locations corresponding to the background and reflective surfaces have been eliminated, leaving behind pixel locations corresponding to the most prominent features of the spacecraft.

Algorithms relying on the Hough transform require the tuning of hyperparameters such as the expected minimum length of line segments and the expected maximum gap between two points to be considered in the same line segment. This makes these algorithms unsuitable for broad applicability because these parameters need to

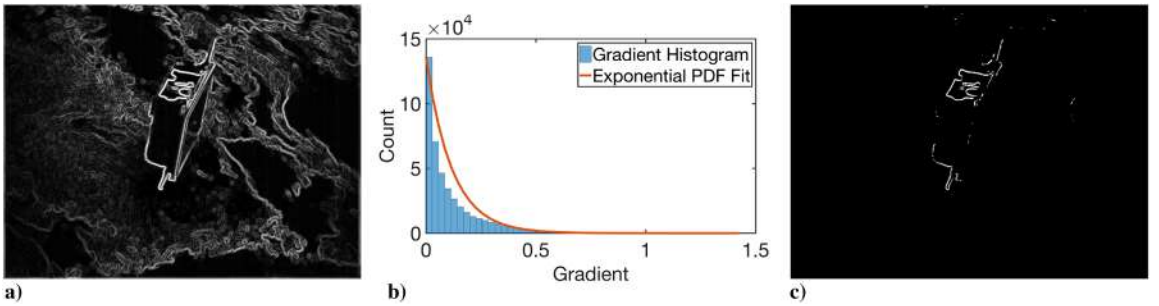
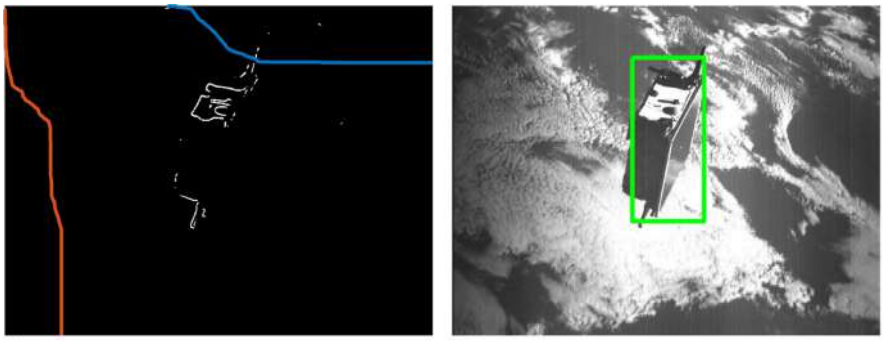


Fig. 4 Steps of weak gradient elimination: a) gradient detection of original image, b) histogram of normalized gradient values and exponential PDF approximation, and c) output filtered image gradients.

Downloaded by STANFORD UNIVERSITY on July 2, 2018 | http://arc.aiaa.org | DOI: 10.2514/1.A34124



a) b)

Fig. 5 ROI detection process: a) cumulative population of strong gradients along the two image axes and b) output ROI.

Downloaded by STANFORD UNIVERSITY on July 2, 2018 | http://arc.aiaa.org | DOI: 10.2514/1.A34124

be varied for different imaging scenarios [40]. To overcome this challenge, the WGE technique allows for an adaptive computation of these hyperparameters through the detection of a rectangular ROI around the target spacecraft in the image. The limits of the rectangular ROI are determined independently in each of the two axes of the image. To build the ROI, the cumulative distribution function (CDF) of the filtered image gradient obtained from the WGE is determined along the two image axes, as seen in Fig. 5. Assuming the filtered image gradient is normally distributed, the coordinates of the ROI are determined by axes positions corresponding to CDFs of 0.025 and 0.975 (therefore, only the central 95% of the normal distribution is considered).

To extract line segments corresponding to small spacecraft features, the Hough transform is applied to the binary image of the filtered gradient. The required hyperparameters of the Hough transform (namely, the expected minimum length of the line segments $l_{\text{min,Hough}}$ and the maximum gap between two points to be considered in the same line segment λ_{Hough}) can be calculated as scalar multiples of the diagonal length of the ROI l_{ROI}:

$$l_{\text{min,Hough}} = \kappa_1 * l_{\text{ROI}} \qquad \lambda_{\text{Hough}} = \kappa_2 * l_{\text{ROI}} \tag{5}$$

The outputs of short line segments are stored and later merged with the line segments belonging to large features. The scalars κ_1 and κ_2 can be empirically estimated in a simulation on ground before the mission or can be estimated on board using prior knowledge of the interspacecraft range based on an angles-only navigation phase [41].

An innovative feature of the WGE technique is its ability to provide a coarse relative position solution even before the pose determination subsystem. Figure 6 shows that the knowledge of the diagonal characteristic length L_C of the spacecraft 3-D model and the diagonal length of the detected ROI l_{ROI} can be used to obtain the position of the target spacecraft relative to the camera frame $\boldsymbol{t}^C$. In particular, the range to the target spacecraft from the origin of the camera frame is

$$\|\boldsymbol{t}^C\|_2 = \frac{((f_x + f_y)/2)L_C}{l_{\text{ROI}}} \tag{6}$$

where f_x and f_y denote the focal lengths of the camera. Azimuth and elevation angles (α, β) from the origin of the camera frame C to the origin of the body-fixed coordinate system B can be derived using the principal point of the image (C_x, C_y) and the center of the ROI (B_x, B_y):

$$\alpha = \tan^{-1}\left(\frac{B_x - C_x}{f_x}\right) \tag{7}$$

$$\beta = \tan^{-1}\left(\frac{B_y - C_y}{f_y}\right) \tag{8}$$

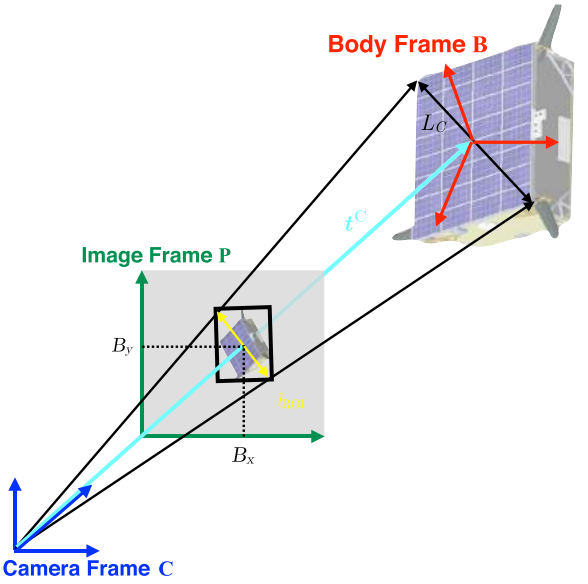


Fig. 6 Calculation of a coarse relative position solution using the WGE technique.

Finally, the coarse relative position solution is given by

$$\boldsymbol{t}^C = \begin{bmatrix} C_\alpha & 0 & -S_\alpha \\ 0 & 1 & 0 \\ S_\alpha & 0 & C_\alpha \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & C_\beta & S_\beta \\ 0 & -S_\beta & C_\beta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \|\boldsymbol{t}^C\|_2 \end{bmatrix} \tag{9}$$

2. *Sobel and Hough*

The second stream of the feature detection procedure consists of the application of the Sobel operator and the Hough transform (S&H) technique [32] to the rectified image. The objective is to easily extract line segments corresponding to the silhouette of the large components of the spacecraft. Any line segment for which the midpoint lies outside the ROI detected from the WGE technique is rejected. The Hough transform hyperparameters are calculated as

$$l_{\text{min,Hough}} = \kappa_3 * l_{\text{ROI}} \qquad \lambda_{\text{Hough}} = \kappa_4 * l_{\text{ROI}} \tag{10}$$

Note that, instead of manually tuning the Hough transform hyperparameters for each image separately, the hyperparameters are adaptively computed based on the scalar multiples κ_3 and κ_4. Due to this formulation of the Hough hyperparameters, the proposed straight line segment extraction is largely robust to the interspacecraft separation. The hypothesis is that, as the target spacecraft gets closer to the camera, the expected lengths of its edges in the image plane are expected to grow proportionally to the size of the detected bounding box. Hence,

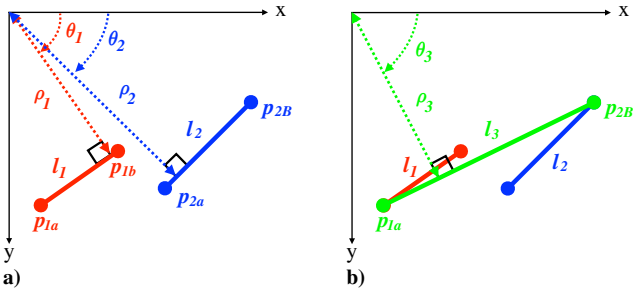


Fig. 7 Merging of two truncated edges: a) original edges, and b) output merged edge.

the determination of parameters κ_3 and κ_4 needs to occur just once: for example, in an offline phase before the mission.

3. Merging Edges

The output line segments from the Hough transform often correspond to multiple and truncated edges. In Fig. 3, edge 4 and edge 5 obtained from the Hough transform correspond to the same true line segment. To resolve this issue, similar line segments are merged into a single line segment. For example, consider two line segments, l_1 and l_2, expressed in the polar form as shown in Fig. 7a:

$$l_1{:}\rho_1 = x\cos\theta_1 + y\sin\theta_1 \tag{11}$$

$$l_2{:}\rho_2 = x\cos\theta_2 + y\sin\theta_2 \tag{12}$$

The condition of similarity is that $|\theta_1 - \theta_2| < \theta_{\text{thresh}}$ and $|\rho_1 - \rho_2| < \rho_{\text{thresh}}$. Furthermore, the Euclidean distance between the farthest pair of endpoints of the two line segments must be less than d_{thresh}. The parameter d_{thresh} is adaptively computed for each image as half of the mean length of the detected edge segments, whereas the parameters θ_{thresh} and ρ_{thresh} are set equal to the resolution of θ and ρ in the Hough space. If the similarity condition is verified, the two line segments are replaced with l_3, which is the line segment defined by the farthest pair of endpoints of the line segments, as shown in Fig. 7b:

$$l_3{:}\rho_3 = x\cos\theta_3 + y\sin\theta_3 \tag{13}$$

4. Merging Streams

The final step of the feature detection procedure is to merge the line segments detected with the WGE and S&H techniques. Only unique line segments are output from each of the two streams before merging. This uniqueness check resolves the issue of detecting repeated edges as encountered in previous work [31]. Pairs of close and similar line segments are detected in the output of both the streams separately, and only the longer line segment from the pair is preserved. Using the example of the pair of line segments l_1 and l_2 from Eqs. (11) and (12), the longer line segment is retained if $|\theta_1 - \theta_2| < \tilde{\theta}_{\text{thresh}}$, $|\rho_1 - \rho_2| < \tilde{\rho}_{\text{thresh}}$, and the Euclidean distance between the midpoints is less than half of the length of the longer line segment. The parameters $\tilde{\theta}_{\text{thresh}}$ and $\tilde{\rho}_{\text{thresh}}$ can be tuned to a certain threshold or expressed as functions of the size of the ROI.

The results of the two streams are combined, taking into account the cases where short line segments detected from the WGE technique overlap or intersect with the large line segments output from the S&H technique. Pairs of line segments from the two streams are compared, and we check whether they intersect. If they do intersect, then assuming the shorter line segment is divided into two portions with lengths L_1 and L_2 (where $L_1 < L_2$), only the longer line segment is preserved if $L_1/L_2 > 0.25$.

B. Feature Synthesis

The objective of feature synthesis is to organize the extracted line segments into higher-level features to reduce the search space of the correspondence problem, i.e., matching features of the model to the features in the image. Given n points in the image and m points in the 3-D model, the number of possible correspondences and, equivalently, the number of hypothetical pose solutions are given by

$$\binom{m}{6}\binom{n}{6}6!$$

Note that we need at least six corresponding model and image points to guarantee a unique solution of the PnP problem [34]. However, more constraints can be generated using the knowledge of the 3-D model and by organizing the points into higher-level groups. For example, if four of these image points belong to a polygonal feature such as a solar panel, then a unique solution can be found using just four correspondences. This drastically reduces the number of possible correspondences to $8m_{\text{planar}}n_{\text{planar}}$, where m_{planar} and n_{planar} are the numbers of four-sided polygonal features of the 3-D model and the image, respectively. Therefore, the key idea is to solve the correspondence problem using just a small number of higher-level feature groups instead of using a large number of feature points.

In this implementation of feature synthesis, the detected line segments are organized into five groups, namely, parallel pair, proximal pair, open polygonal triad, parallel triad, and closed polygonal tetrad. These groups are built using relations that are preserved over a wide range of camera viewpoints [42], thereby adding robustness to the proposed architecture. Moreover, these groups can be easily and quickly found from the detected line segments by examining just a few geometric constraints, as observed in Fig. 8. For example, two segments compose a proximity pair if they satisfy the condition

$$d_{12} \leq d_{\max} \tag{14}$$

where d_{12} is the shortest distance between endpoints, and $d_{\max}$ is a threshold value in pixels. Similarly, two segments compose a parallel pair if they satisfy the condition

$$\theta_{12} = \|\theta_1 - \theta_2\| \leq \theta_{\max} \tag{15}$$

where θ_1 and θ_2 are the line segment slopes, and $\theta_{\max}$ is a threshold value in degrees. If two groups of proximity pairs share a line segment, they are categorized as an open polygonal triad if they satisfy the condition

$$(P_{1a} - P_{3a}) \cdot (P_{1b} - P_{3b}) > 0 \tag{16}$$

This ensures that the endpoints of the noncommon line segments lie on the same side with respect to the shared line segment. If two open polygonal triads are found with two shared line segments, then they are categorized as a closed polygonal tetrad. Similarly, if two parallel pairs are found with a shared line segment, they are categorized as a parallel triad. Finally, any line segments that were detected through the WGE technique feature detection stream but were not detected from the S&H feature detection stream are classified as antennas if the length of the segment is less than one-third of l_{ROI}. This particular threshold can be easily tuned before the mission as the 3-D model of the target spacecraft is assumed to be available. As for the 3-D model of the target spacecraft, the same groups can be precomputed using the same conditions before the mission. Figure 9 shows the results from the application of feature synthesis routine on some actual space imagery from the PRISMA mission.

IV. Pose Determination

The position and attitude of the target spacecraft are determined by solving the 3-D/2-D perspective projection equation for R^{BC} and $\boldsymbol{t}^C$. Because this requires corresponding image and 3-D model points, the detected feature groups must be first matched with the corresponding feature groups of the model. Notably, several possible combinations between the image and correlated model features must be considered because the exact correspondences are unknown. The 3-D/2-D true perspective projection equation is then solved for each combination of feature correspondences using the EPnP method [26]. The best pose candidates are refined through a Newton–Raphson algorithm in

Downloaded by STANFORD UNIVERSITY on July 2, 2018 | http://arc.aiaa.org | DOI: 10.2514/1.A34124

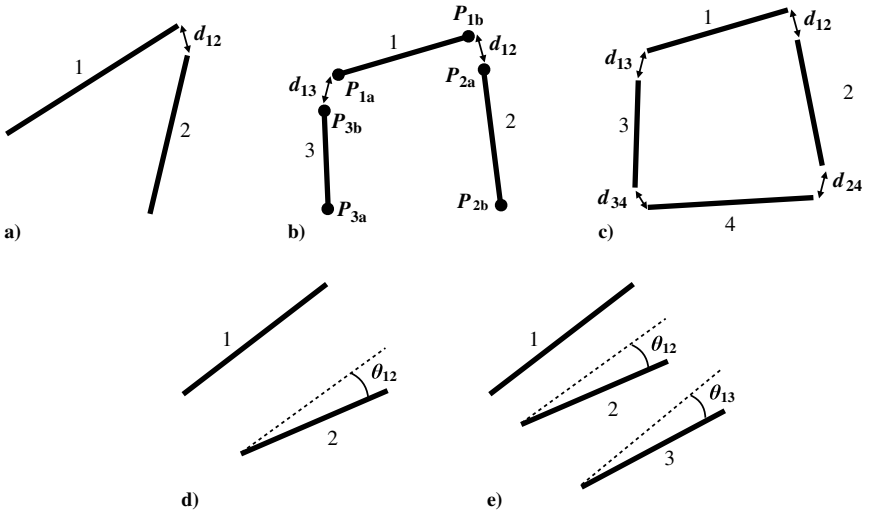


Fig. 8 Synthesis of detected line segments into higher-level features: a) proximal pair, b) open polygonal triad, c) closed polygonal tetrad, d) parallel pair, and e) parallel triad.

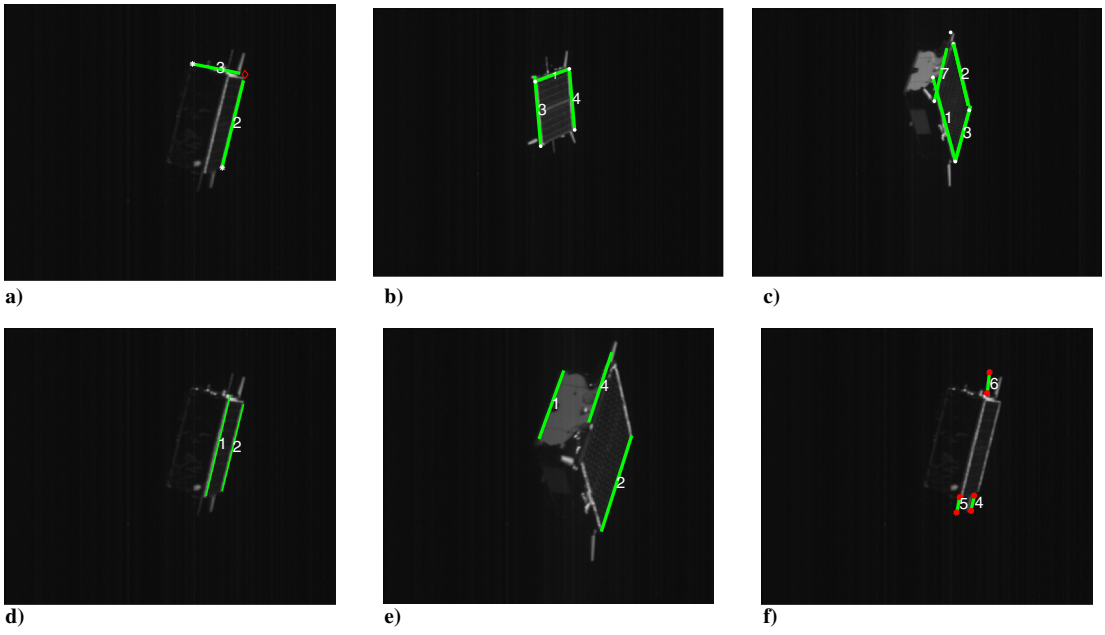


Fig. 9 Intermediate results from synthesis of detected line segments into higher-level features: a) proximal pair, b) open polygonal triad, c) closed polygonal tetrad, d) parallel pair, e) parallel triad, and f) antennas.

order to output a single pose solution. Figure 10 presents the pose determination subsystem, and the following subsections discuss its main steps in detail.

A. Feature Correspondence

The correspondences between the image points and the 3-D model points are obtained by pairing each feature group detected in the image with each analogous group of the 3-D model of the target spacecraft. For each matching feature group pair, the endpoints of the line segments in the image are hypothesized to correspond with the endpoints of the 3-D model's lines through simple combinations. For instance, a closed polygon identified in the image is coupled with every closed polygon of the model. This provides eight different combinations of the point correspondences between the closed polygon detected in the image and the closed polygon of the 3-D model. This approach is applied to all the feature groups considered in the feature synthesis in Sec. IV.

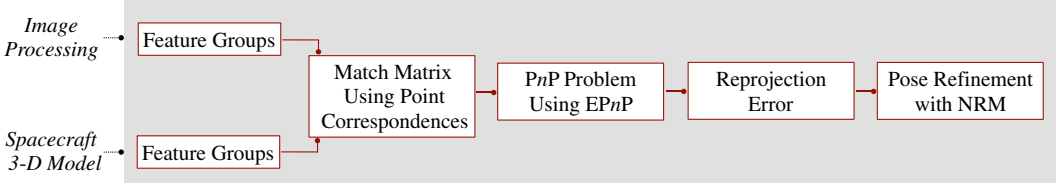


Fig. 10 Main steps of the pose determination subsystem with feature groups from the image and the 3-D model as input and a single pose solution as the output.

Downloaded by STANFORD UNIVERSITY on July 2, 2018 | http://arc.aiaa.org | DOI: 10.2514/1.A34124

Table 1 Expected number of rows in the match matrix (column 5) based on the most geometrically complex feature group detected in the image (column 1)

Feature group	Number of points per feature group	Number of feature groups in image	Number of feature groups in 3-D model	Number of rows in match matrix
Closed polygonal tetrad	4	ϕ_a	ϕ'_a	$8\phi_a\phi'_a\phi_f\phi'_f$
Open polygonal tetrad	4	ϕ_b	ϕ'_b	$8\phi_b\phi'_b\phi_f\phi'_f$
Parallel triad	6	ϕ_c	ϕ'_c	$24\phi_c\phi'_c$
Parallel pair	4	ϕ_d	ϕ'_d	$8\phi_d\phi'_d\phi_f\phi'_f$
Proximal pair	3	ϕ_e	ϕ'_e	$2\phi_e\phi'_e\phi_f\phi'_f$
Antenna	1	ϕ_f	ϕ'_f	— —

The point correspondences between the image and the 3-D model are stored in the so-called match matrix, which is then input to the EPnP method [26]. The rows of the matrix represent the different hypotheses for feature correspondence, whereas the columns store the corresponding 3-D model and 2-D image points. EPnP requires at least six point correspondences to guarantee a unique pose solution from Eqs. (1) and (2) [43], whereas a single feature group typically provides less than six (see Table 1). Therefore, the point correspondences from the feature groups are combined with the point correspondences provided by the antennas feature group to form at least six point correspondences. Note that the decision to combine point correspondences from complex feature groups with point correspondences from the antennas feature group is a design choice because the Tango spacecraft has five prominent antennas visible from most viewing angles. For other spacecraft where antennas are either not present or prominent, point correspondences from other complex feature groups may be used to form at least six point correspondences. The feature groups are ranked according to their geometric complexity as follows (in descending order): closed polygonal tetrad, open polygonal tetrad, parallel triad, proximal pair, and parallel pair. To build the match matrix, only the most geometrically complex feature group detected in that image is considered. Table 1 shows the number of rows in the match matrix for a typical scenario where at least three antennas are detected. Building the match matrix in this manner has two main advantages: first, the probability of an accidental detection of the higher geometric complexity group is less than that of the lower geometric complexity group [42]; and second, the higher the geometric complexity of the feature group, the lower the number of possible correspondences between the image and the 3-D model features.

Unlike most previous algorithms to hypothesize feature correspondence, not all possible feature matches are treated identically. Instead, a small set of matches is hypothesized and then verified. This is in stark contrast to view-based approaches where image features were compared with precomputed 2-D views of the object to determine the object pose [44–47]. These approaches tried to deal with the full geometric search space by clustering the views. None of these are practical for the constrained space-hardened hardware due to their extremely large geometric search space.

B. EPnP

Each combination of feature correspondences stored in the match matrix is employed to solve the 3-D/2-D perspective equation using the EPnP method [26]. Notably, EPnP is chosen among state-of-the-art algorithms because it ensures rapid convergence and robustness to image noise and feature outliers [43].

The key concept of the EPnP method is its use of four noncoplanar unknown control points: $\boldsymbol{c}_1^C$, $\boldsymbol{c}_2^C$, $\boldsymbol{c}_3^C$, and $\boldsymbol{c}_4^C$. Let n be the number of correspondences between the image and model features. Each model point can be expressed in the camera coordinate system C as a linear combination of the control points according to

$$\boldsymbol{r}_i^C = \sum_{j=1}^{4} \gamma_{ij}\boldsymbol{c}_j^C \qquad i = 1, 2, \ldots, n \tag{17}$$

In the ideal case, each feature point $\boldsymbol{q}_i = [u_i, v_i]$ detected in the image coincides to the corresponding model point $\boldsymbol{p}_i^B$ projected onto the image. Notably, this yields to

$$u_i = \frac{x_i^C}{z_i^C} f_x + C_x \qquad i = 1, 2, \ldots, n \tag{18}$$

$$v_i = \frac{y_i^C}{z_i^C} f_y + C_y \qquad i = 1, 2, \ldots, n \tag{19}$$

where $\boldsymbol{r}_i^C = [x_i^C, y_i^C, z_i^C]$ is the model point written in the camera coordinate system and obtained from $\boldsymbol{p}_i^B$ using Eq. (1). By substituting Eq. (17) into Eqs. (18) and (19), we obtain the following linear equations for each pair of features:

$$\sum_{j=1}^{4} \gamma_{ij} f_x c_{x_j}^C + \sum_{j=1}^{4} (C_x - u_i)\gamma_{ij} c_{z_j}^C = 0 \qquad i = 1, 2, \ldots, n \tag{20}$$

$$\sum_{j=1}^{4} \gamma_{ij} f_y c_{y_j}^C + \sum_{j=1}^{4} (C_y - v_i)\gamma_{ij} c_{z_j}^C = 0 \qquad i = 1, 2, \ldots, n \tag{21}$$

This linear system of $2 \times n$ equations is solved for the 12 unknown parameters that are the components of the control points $\boldsymbol{c}_1^C$, $\boldsymbol{c}_2^C$, $\boldsymbol{c}_3^C$, and $\boldsymbol{c}_4^C$. Therefore, at least $n = 6$ correspondences between the image and model features are required to obtain the solution. The pose of the target is then retrieved using Eqs. (1) and (17).

The EPnP algorithm is applied to each combination of feature correspondences defined in the match matrix. The reprojection error is the Euclidean distance between image features and corresponding model points projected onto the image, which is mathematically expressed as

$$E_{2D} = \frac{1}{n}\sum_{i=1}^{n} \sqrt{\left[u_i - \left(\frac{x_i^C}{z_i^C} f_x + C_x\right)\right]^2 + \left[v_i - \left(\frac{y_i^C}{z_i^C} f_y + C_y\right)\right]^2} \tag{22}$$

Notably, $\boldsymbol{r}_i^C = [x_i^C, y_i^C, z_i^C]$ is computed from the model feature point $\boldsymbol{p}_i^B$ using Eq. (1).

C. Pose Refinement

The best five pose solutions in terms of their solution error [Eq. (22)] are used for pose refinement. The first step of this procedure is to solve the 3-D/2-D perspective projection equation using the Newton–Raphson method (NRM) [31] using each pose solution as the first guess. For each feature correspondence, the following fit error between the detected image feature and the projected model point is defined:

$$\boldsymbol{E}_i = \left[u_i - \left(\frac{x_i^C}{z_i^C} f_x + C_x\right), v_i - \left(\frac{y_i^C}{z_i^C} f_y + C_y\right)\right] \qquad i = 1, 2, \ldots, n \tag{23}$$

where $\boldsymbol{r}_i^C = [x_i^C, y_i^C, z_i^C]$ is obtained from $\boldsymbol{p}_i^B$ using Eq. (1). The fit error in Eq. (23) has six unknown parameters of $\boldsymbol{x} = [\boldsymbol{t}^C, \boldsymbol{\theta}_{BC}]$, where $\boldsymbol{\theta}_{BC}$ is the Euler angles sequence that defines the rotation matrix R^{BC}.

Downloaded by STANFORD UNIVERSITY on July 2, 2018 | http://arc.aiaa.org | DOI: 10.2514/1.A34124

Downloaded by STANFORD UNIVERSITY on July 2, 2018 | http://arc.aiaa.org | DOI: 10.2514/1.A34124

Because each feature correspondence provides two conditions, at least three matches between detected corners and projected features are required to solve the system of equations defined by the fit errors. Let $n \geq 3$ be the number of correspondences between the image and model features. The system of $2 \times n$ equations that must be solved for $\boldsymbol{x}$ is given by

$$\boldsymbol{E}_s = \begin{bmatrix} \boldsymbol{E}_1 \\ \cdots \\ \boldsymbol{E}_n \end{bmatrix} \tag{24}$$

The NRM solves this system of equations by iteratively updating the solution as

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - (\boldsymbol{J}^T\boldsymbol{J})^{-1}\boldsymbol{J}^T\boldsymbol{E}_s(\boldsymbol{x}_k) \tag{25}$$

where $\boldsymbol{E}_s(\boldsymbol{x}_k)$ is evaluated using Eq. (24) at $\boldsymbol{x}_k$, and $\boldsymbol{J}$ is the Jacobian of the system:

$$\boldsymbol{J} = \frac{\partial \boldsymbol{E}_s}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial \boldsymbol{q}_1}{\partial \boldsymbol{r}_1^C}\frac{\partial \boldsymbol{r}_1^C}{\partial \boldsymbol{t}^C} & \frac{\partial \boldsymbol{q}_1}{\partial \boldsymbol{r}_1^C}\frac{\partial \boldsymbol{r}_1^C}{\partial \boldsymbol{\theta}_{\mathrm{BC}}} \\ \cdots & \cdots \\ \frac{\partial \boldsymbol{q}_n}{\partial \boldsymbol{r}_n^C}\frac{\partial \boldsymbol{r}_n^C}{\partial \boldsymbol{t}^C} & \frac{\partial \boldsymbol{q}_n}{\partial \boldsymbol{r}_n^C}\frac{\partial \boldsymbol{r}_n^C}{\partial \boldsymbol{\theta}_{\mathrm{BC}}} \end{bmatrix} \tag{26}$$

The partial derivatives in Eq. (26) are obtained from Eqs. (1) and (2) and are given by the following:

$$\frac{\partial \boldsymbol{q}}{\partial \boldsymbol{r}^C} = \begin{bmatrix} \frac{f_x}{z^C} & 0 & -\frac{f_x x^C}{(z^C)^2} \\ 0 & \frac{f_y y^C}{z^C} & -\frac{f_y}{(z^C)^2} \end{bmatrix} \tag{27}$$

$$\frac{\partial \boldsymbol{r}^C}{\partial \boldsymbol{t}^C} = R^{\mathrm{BC}}\boldsymbol{p}^B \tag{28}$$

$$\frac{\partial \boldsymbol{r}^C}{\partial \boldsymbol{t}^C} = \frac{\partial R^{\mathrm{BC}}}{\partial \boldsymbol{\theta}_{\mathrm{BC}}}\boldsymbol{p}^B \tag{29}$$

The iterative routine stops when either the improvement of the solution achieves the tolerance or the number of iterations reaches the maximum number. By applying the NRM to each of the selected best solutions generated by EPnP, new pose solutions are obtained. This set of solution candidates is used to project the 3-D model using the painter's algorithm. A nearest-neighbor search is employed to match the endpoints of the line segments detected in the image with the endpoints of the projected model's line segments. The output pose solution is the one that minimizes the reprojection error [see Eq. (22)]. Because the reprojection error is a measure of how well the pose solution "fits" the detected image features, it can be used as a confidence metric. A "high-confidence pose" will generally have a very low reprojection error as compared to a "low-confidence pose." This allows the proposed architecture to be run on successive images until a threshold reprojection error is met.

V. Validation

The SVD architecture and its constituent subsystems are independently tested on the imagery collected during the PRISMA mission [33] to evaluate their performance and quantitatively compare their strengths and weaknesses against the state-of-the-art methods. The 3-D wireframe model used in this validation effort is illustrated in Fig. 11. This model is derived by reducing a high fidelity computer-aided design model of the Tango spacecraft [31] to only contain a low number of features. The set of features present in this model is selected to reduce possible pose ambiguities (by maintaining geometric asymmetry) and to reduce the number of feature correspondence

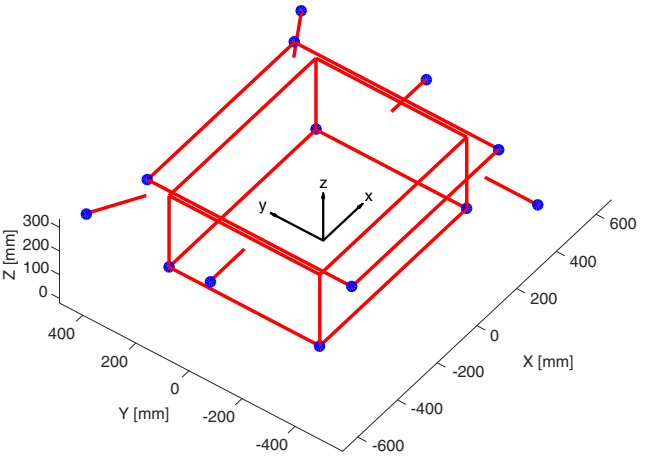


Fig. 11 3-D wireframe model of the Tango satellite.

hypotheses during pose determination. It consists of a polygon representing the solar panel (560×750 mm) and a convex polyhedron representing the spacecraft body ($560 \times 550 \times 300$ mm). Five additional segments (204 mm) represent the radio-frequency antennas. The origin of the body frame is located at the center of the bottom face of the spacecraft body. The model is input in MATLAB as a stereolithographic file from which information about surfaces and edges is generated. Notably, the same feature synthesis groups introduced in Sec. IV can be extracted from the 3-D model using the same condition checks and functions. We detected 16 proximal pairs, 18 parallel pairs, 12 parallel triads, 6 closed polygonal tetrads, 6 open polygonal tetrads, and 5 antennas from the 3-D model.

In the tests that used datasets containing PRISMA imagery, flight dynamics products from the PRISMA mission [48] have been used for performance evaluation. Specifically, onground precise relative orbit determination based on the Global Positioning System (accurate to about 2 cm 3-D rms) [48] is used as the "true" relative position, and an onboard coarse attitude estimate from the sun sensors and magnetometers (accurate to about 3° 3-D rms) [31] is used to calculate the true relative attitude. The accuracy in the estimated relative position is evaluated by the following translation error:

$$E_T = |\boldsymbol{t}^C_{\mathrm{true}} - \boldsymbol{t}^C_{\mathrm{est}}| \tag{30}$$

which represents the elementwise absolute difference between the position $\boldsymbol{t}^C_{\mathrm{true}}$ of the client obtained from the flight dynamics products and the position $\boldsymbol{t}^C_{\mathrm{est}}$ provided by the pose solution. Similarly, the accuracy of the attitude solution of the target vehicle is evaluated through the Euler angle representation of the rotational error

$$E_R(R_{\mathrm{diff}}) = (\phi, \theta, \psi) \tag{31}$$

where R_{diff} is a direction cosine matrix representing the relative rotation between the true value and the estimate value of R^{BC}:

$$R_{\mathrm{diff}} = R^{\mathrm{BC}}_{\mathrm{est}}(R^{\mathrm{BC}}_{\mathrm{true}})^T \tag{32}$$

Geometrically, ϕ, θ, and ψ represent the errors in the estimated attitude about the directions C_1, C_2, and C_3, respectively. As shown in Fig. 1, the direction C_3 is pointed along the boresight, whereas C_1 and C_2 are aligned with the image frame P. Lastly, the reprojection error $E_{2\mathrm{D}}$, of the final pose estimate is calculated using Eq. (22) and is reported in pixels. The reprojection error quantifies how closely the final pose estimate recreates the features detected in the image. Table 2 presents the purpose, the image datasets, and the methods used during the four tests conducted as part of the validation effort. The following subsections present a detailed account of these four tests.

A. Test 1

This test compared the performance of state-of-the-art feature extractors with that of WGE using a dataset of 142 images from the PRISMA mission (referred to in Table 2 as PRISMA-142).

Table 2 Description of the tests conducted for the validation and comparison of the SVD architecture against the state-of-the-art methods

Test name	Test purpose	Image datasets	Methods
Test 1	Extraction of line segment endpoints	PRISMA-142	FAST [49], BRISK [50], Sobel + Hough [32], Canny+Hough [36], Prewitt+Hough [39], WGE
Test 2	Detection of region of interest	PRISMA-142	MSER [37], WGE
Test 3	Solution of perspective equations	PRISMA-5, Synthetic-5	EPnP [26], ASPnL, LPnL-DLT, LPnL-LS, LPnL-ENull [24], Ansar and Daniilidis [25], Mirzaei and Roumeliotis [23]
Test 4	Pose initialization with unknown feature correspondences	PRISMA-25	WGE + RANSAC [34], WGE + SVD

The image processing subsystem is required to output pixel locations of edge endpoints that correspond to line segment endpoints of the 3-D model. This can be achieved either through 1) coupling edge detectors such as Sobel [32], Canny [36], and Prewitt [39] with the Hough transform; 2) using corner detectors such as features from accelerated segment test (FAST) [49]; or 3) using key-point detectors such as binary robust invariant scalable keypoints (BRISK) [50]. The true positive rate (TPR) and positive predictive value (PPV) were calculated for each image in PRISMA-142 as

$$\text{TPR} = \frac{\text{Number of true positives}}{\text{Number of true positives} + \text{Number of false positives}} \cdot 100 \tag{33}$$

$$\text{PPV} = \frac{\text{Number of true positives}}{\text{Number of true positives} + \text{Number of true negatives}} \cdot 100 \tag{34}$$

The ground truth for the edge endpoints were manually annotated on the images. A pixel location output was classified as a true positive if it was within a Euclidean distance of 5 pixels from a true edge endpoint. All output pixel locations outside this range were classified as false positives. Any true edge endpoint that remained undetected (i.e., none of the pixel locations output by the feature extractor were within a Euclidean distance of 5 pixels) was counted as a false negative.

Figure 12 shows that the WGE achieved a true positive rate of 32.8%, which is the highest among all feature detectors tested, followed closely by Canny (31.2%) and Sobel (30.4%). However, WGE has a precision of 21.6% as compared to 1.6% for Canny and 4.6% for Sobel. Moreover, this performance comes at an order of magnitude less computational time as other edge detectors because the weak gradient elimination process reduces the number of pixel locations considered by the Hough transform during feature extraction. The performance of WGE is, therefore, far superior to the feature extraction methods based on edge detection. The mean PPV value of the WGE in this test was lower than BRISK and FAST; however, there were zero images where WGE failed to produce a single true positive as compared to 30 for FAST and 11 for BRISK. Figure 13 shows that both FAST and BRISK were a magnitude faster in comparison to WGE. However, that was expected because WGE not only detected pixel locations of endpoints but also provided line segment information (i.e., which endpoints fall on the same edge).

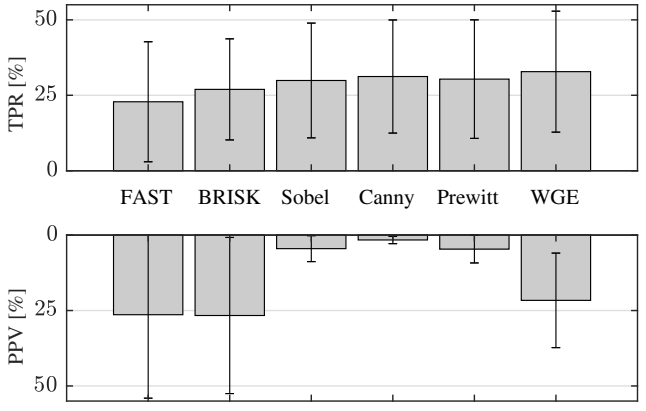


Fig. 12 Mean and standard deviation of the true positive rate and positive predictive values of the feature extraction algorithms on the PRISMA-142 dataset.

B. Test 2

This test compared the ROI output of the WGE on the PRISMA-142 dataset with that of the maximally stable extremal regions [37]. The MSER is a blob detector that outputs a list of pixel locations, whereas the WGE provides a bounding box in the image plane. Therefore, the pixel location list output by the MSER is first converted to a list of bounding boxes. Following that, a single (or multiple in some images) ROI is obtained by applying nonmaximum suppression (NMS) [51] to the list of bounding boxes. The output of MSER+NMS and WGE is compared against the ROI ground truth obtained through manual annotation of the images. The PPV and TPR for each ROI are calculated by classifying their bounded area and using Eqs. (33) and (34). Figure 14 shows the definitions of the true positive, false positive, and true negative regions in the image.

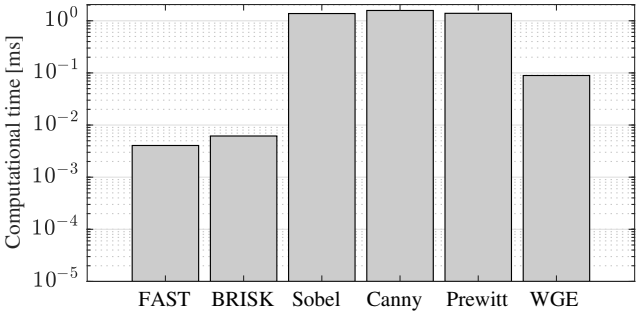


Fig. 13 Mean and standard deviation of the computational time required by the feature extraction algorithms on the PRISMA-142 dataset.

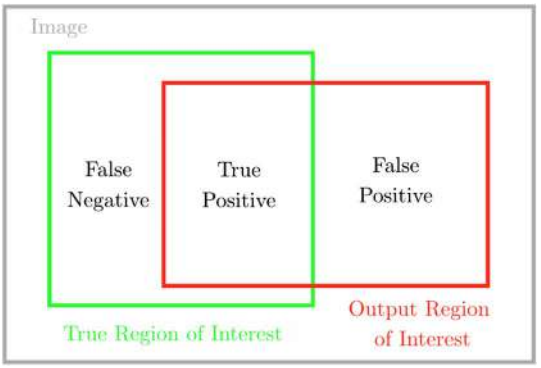


Fig. 14 Class definitions of the area bounded by the region-of-interest output by WGE and MSER + NMS.

Downloaded by STANFORD UNIVERSITY on July 2, 2018 | http://arc.aiaa.org | DOI: 10.2514/1.A34124

Downloaded by STANFORD UNIVERSITY on July 2, 2018 | http://arc.aiaa.org | DOI: 10.2514/1.A34124

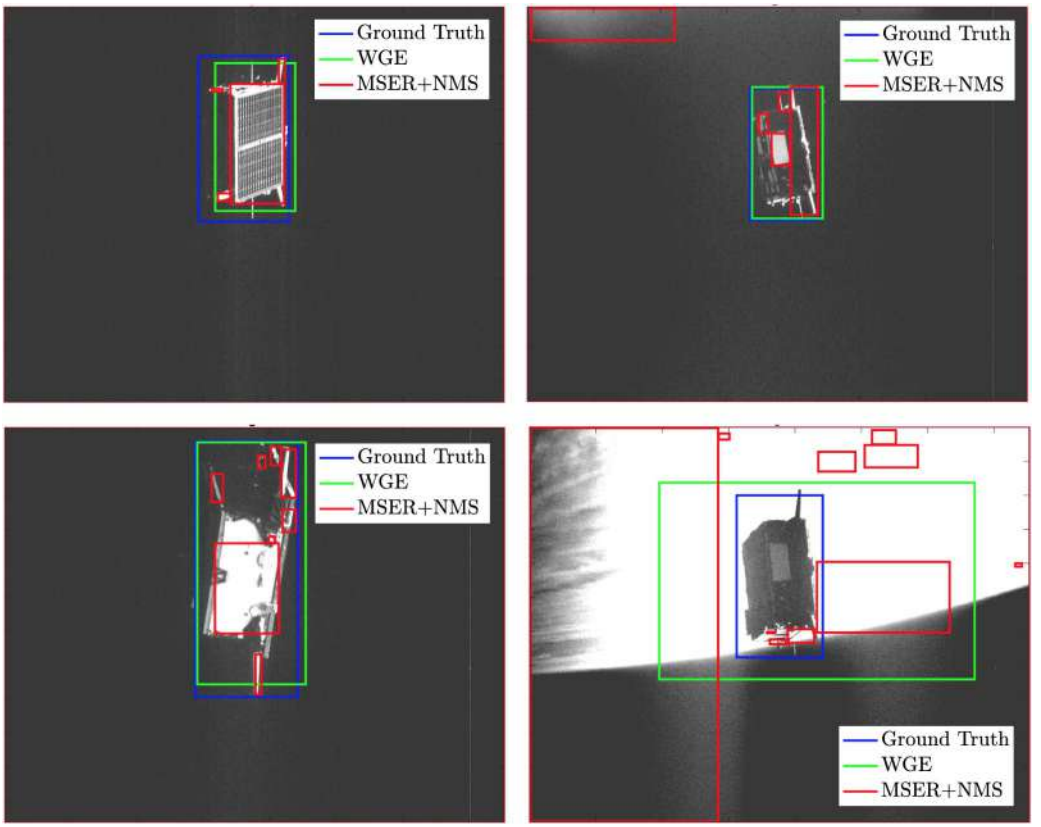


Fig. 15 Region-of-interest output by WGE and MSER + NMS on a set of four images from the PRISMA-142 dataset.

Figure 15 shows some of the results obtained using WGE and MSER + NMS on the PRISMA-142 dataset. The mean values of the TPR and PPV across all 142 images for WGE were measured as 90.59 and 85.98%, respectively. These values are superior to those measured for MSER + NMS (mean TPR = 89.45% and mean PPV = 81.77%). The mean computational time for MSER + NMS, as measured using `tic-toc` on MATLAB running on a 2.4 GHz Intel Core i5 machine, was 0.4867 s; whereas the mean computational time for WGE was 0.0878 s. Moreover, as seen in Fig. 15, MSER + NMS had a tendency to produce multiple ROIs per image; conventionally, a machine learning

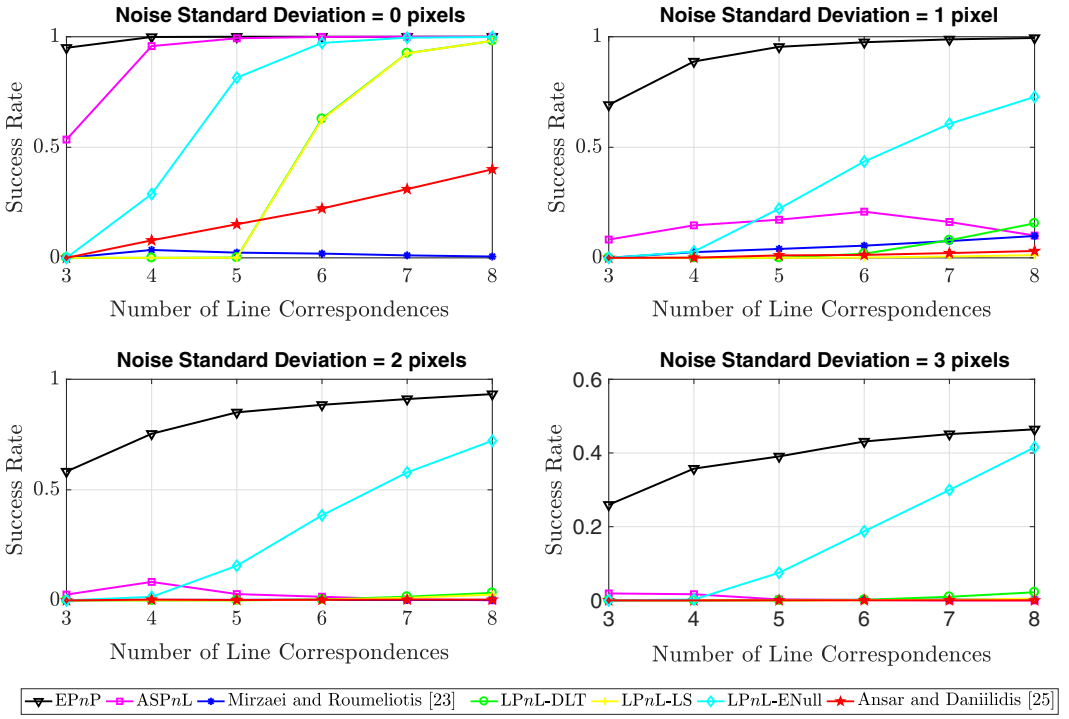


Fig. 16 Average success rate of pose solvers as a function of the number of line correspondences. Plots generated using five random poses and different levels of noise in the projected image.

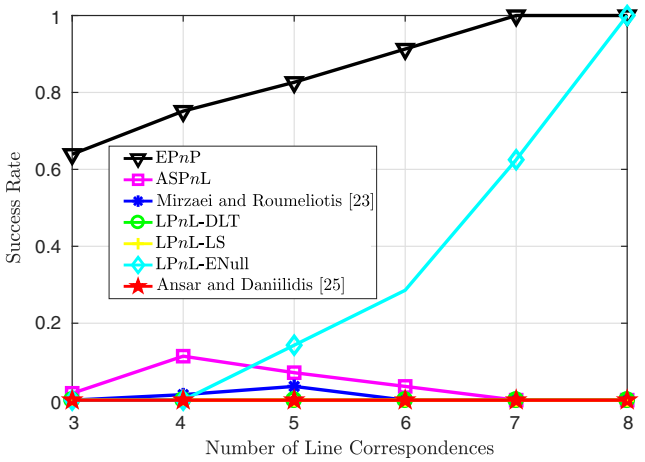


Fig. 17 Average success rate of the pose solvers as a function of the number of line correspondences. Plots generated using five images from the PRISMA mission. DLT: direct linear transform.

Table 3 Accuracy and computation runtime of the pose outputs from EP*n*P and LP*n*L-ENull

Algorithm	$\|E_T\|_2$, m	$\|E_R\|_2$, deg	Runtime, ms
EP*n*P	0.23	2.7	3.0
LP*n*L-ENull	0.30	8.1	3.5

algorithm is required to classify whether the ROI contains the object of interest. However, multiple ROIs per image could be a desirable property if more than one satellite were present in the image. Figure 15 also exhibits a failure case for WGE, where the curvature of the Earth is present in the image. The output of WGE is biased because the horizon represents a sharp change in image intensity.

C. Test 3

This test compares the output of several perspective equation solvers to select one that is both accurate and computationally fast. In particular, the attitude and translation output of several P*n*L algorithms (ASP*n*L [24], LP*n*L [24], Ansar and Daniilidis [25], and Mirzaei and Roumeliotis [23]) is compared against EP*n*P [26], which is a P*n*P solver. The test only considers a single P*n*P solver because EP*n*P has been shown to have superior performance in comparison to other P*n*P solvers [43]. This test employs the use of two datasets. The first dataset contains five synthetic images that are generated by projecting the 3-D model of the spacecraft (see Fig. 11) onto the image plane using random poses. For each image, the P*n*L solvers are tested on different combinations of line correspondences. To create the combinations, several sizes of combinations are considered and each solver is tested on all possible combinations of line correspondences of that size. Unique line segment endpoints are selected from the line correspondences as input for the EP*n*P. The second dataset contains five actual images of the Tango spacecraft acquired during the PRISMA mission [33]. Line correspondences between the image and the 3-D model are manually selected, and each algorithm is tested on several combinations of correspondences. To compare the output of the pose

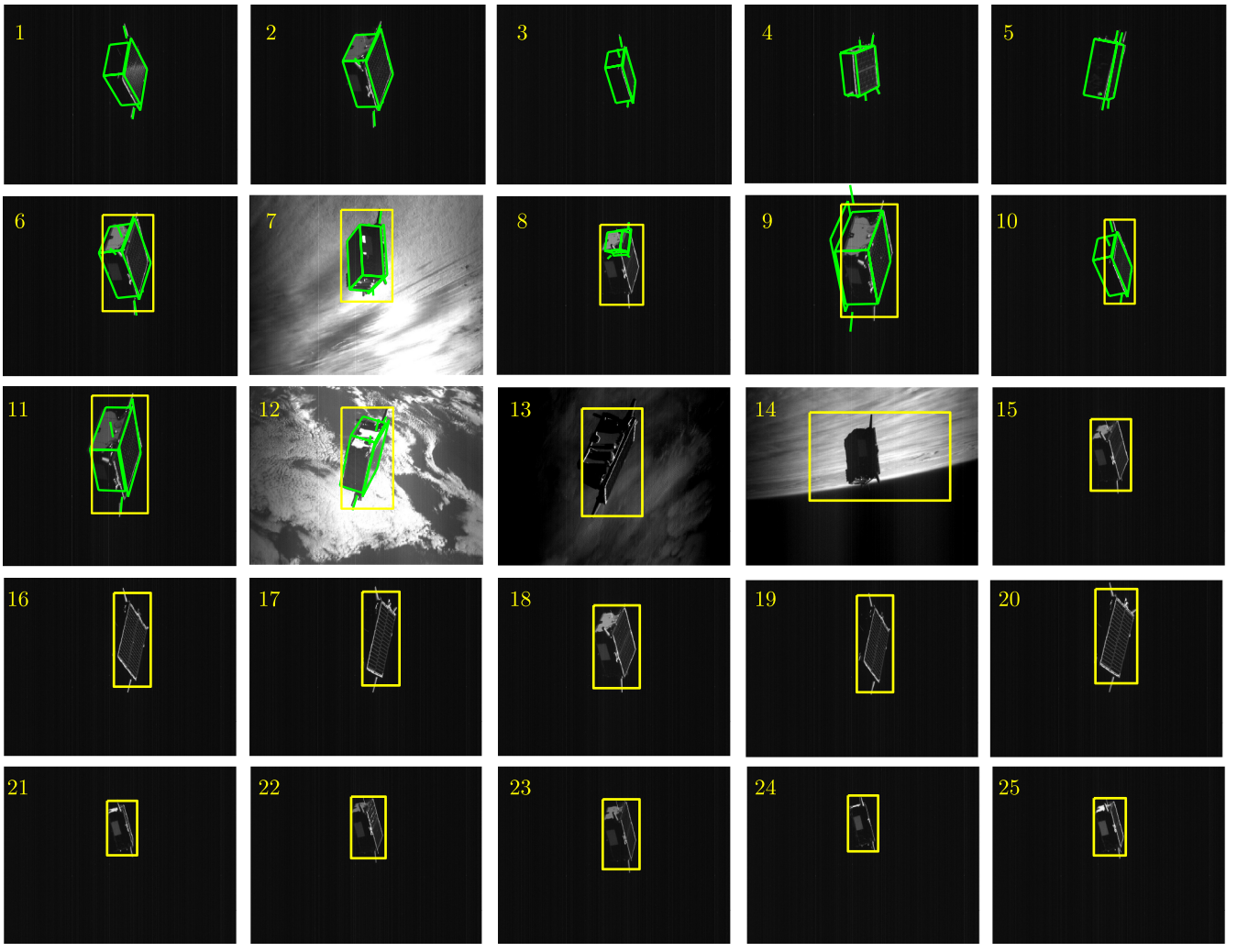


Fig. 18 Pose initialization results using the SVD architecture.

Downloaded by STANFORD UNIVERSITY on July 2, 2018 | http://arc.aiaa.org | DOI: 10.2514/1.A34124

Downloaded by STANFORD UNIVERSITY on July 2, 2018 | http://arc.aiaa.org | DOI: 10.2514/1.A34124

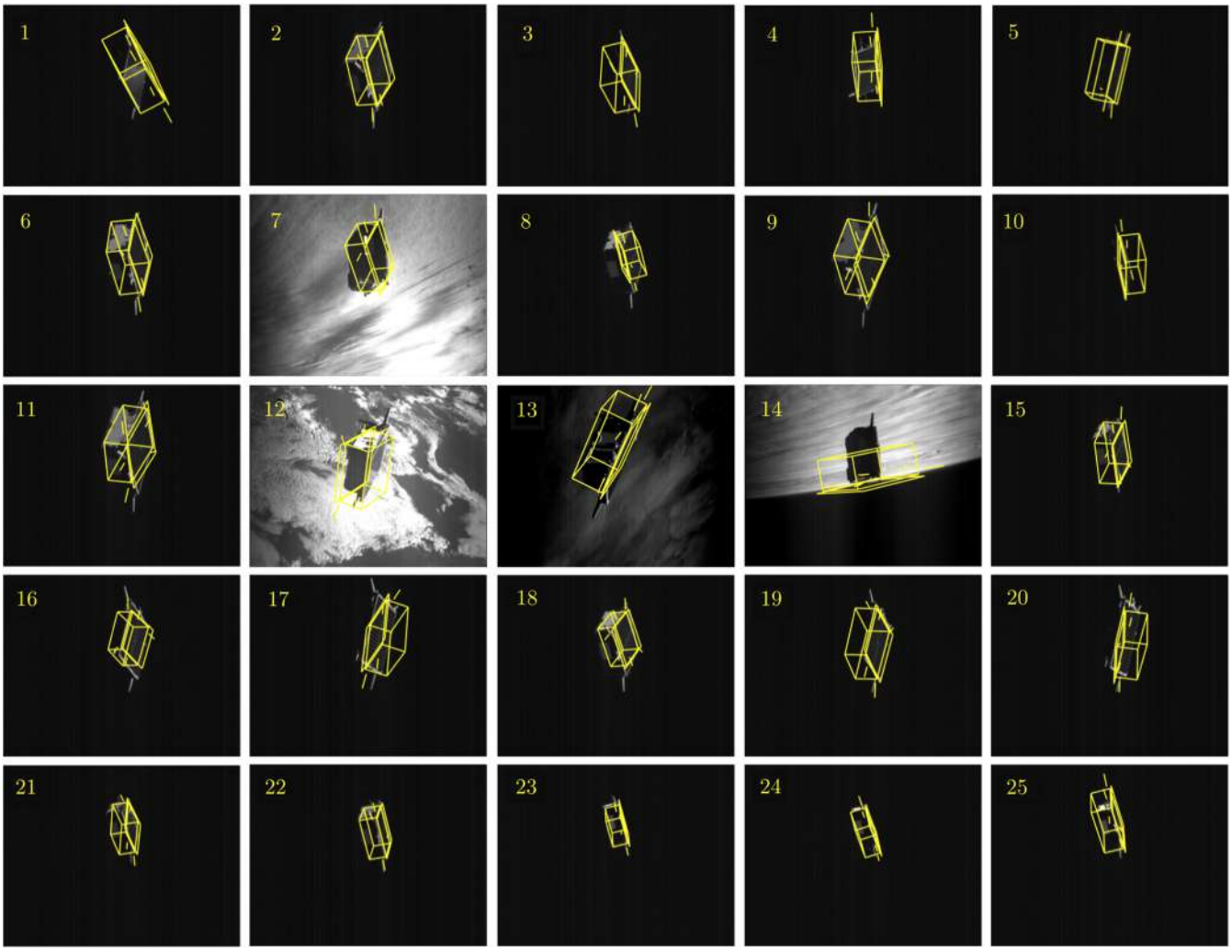


Fig. 19 Pose initialization results using the RANSAC architecture (showing the 3-D model projected on the image plane using the pose solution).

solvers, a success rate based on the number of "correct" pose outputs is calculated. In particular, the success rate is the fraction of pose outputs that have $\|E_T\|_2 < 30$ cm and $\|E_R\|_2 < 10$ deg.

Figure 16 shows the success rate of the pose solvers tested on the dataset of synthetic images. Measurement noise due to the image sensor characteristics is simulated by adding Gaussian noise with zero mean and varying levels of standard deviation to the input of the pose solvers. EP*n*P was found to have the highest success rate for every size of input correspondence and noise level. The reason of the poor performance of perspective-*n*-line algorithms (P*n*L) solvers is that these solvers require a large number of line correspondences to solve the P*n*L problem and they are very sensitive to noise, especially when few correspondences are provided. This result is not surprising because other authors have only demonstrated satisfactory results of P*n*L solvers with at least 10 line correspondences [24]. However, the performance of EP*n*P also degenerates with increasing levels of measurement noise.

Figure 17 shows the performance of the pose solvers tested on the dataset of five real images from the PRISMA mission. Similar to the test cases of the synthetic dataset, EP*n*P was again found to have the highest average success rate for all sets of feature correspondences. Note that the failure cases of EP*n*P with three- and four-feature correspondences are due to the fact that coplanar point correspondences may lead to multiple pose solutions [52]. In cases where both EP*n*P and LP*n*L-ENull produced a correct pose solution, their accuracy (using $\|E_T\|_2$ and $\|E_R\|_2$) and computational runtime (using MATLAB commands *tic* and *toc*) were measured for comparison. As summarized in Table 3, EP*n*P offered a superior performance in terms of both pose accuracy and runtime. For these reasons, EP*n*P was chosen as the pose solver in the proposed pose initialization architecture.

D. Test 4

This test compares the pose estimate of the proposed pose initialization architecture based on the use of feature groups, with one

Table 4 Accuracy of the pose solutions provided by SVD and RANSAC on the PRISMA-25 dataset[a]

SVD solution type	Number of images	Method	E_R, deg	E_T, m
High-confidence pose	5	SVD	(−0.57, 0.59, −1.37)	(0.14, 0.06, 0.51)
		RANSAC	(29.84, 7.52, −17.87)	(0.46, 0.52, 1.94)
Low-confidence pose	7	SVD	(−23.56, −0.67, 16.78)	(0.18, 0.005, 0.75)
		RANSAC	(75.66, 2.18, −22.16)	(0.28, 0.35, 1.13)
Relative position only	13	SVD	— —	(0.07, 0.03, 0.51)
		RANSAC	(−10.04, −2.11, 23.41)	(0.24, 0.48, 1.38)

[a]Values of E_R and E_T are mean values computed across the different images belonging to the particular solution type.

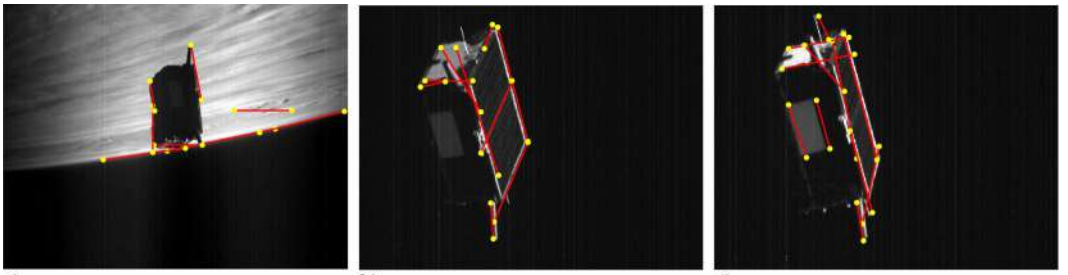


Fig. 20 Example images where the image processing subsystem output contained spurious edges.

based on the use of RANSAC [34]. In particular, the test highlights the advantages and disadvantages of handling unknown feature correspondences through feature synthesis as compared to random hypotheses. RANSAC randomly samples three edges from the image and three line segments from the 3-D model, and their provides their endpoints to EPnP for a pose solution. This pose is then verified by projecting the 3-D model on the image plane and counting the number of "inlier" edges that agree with it, i.e., the number of detected edges that are closer than a predefined threshold to the corresponding projected edges. The pose solution with the highest number of inlier edges is output after k random hypotheses. The value of k is governed by

$$k = \frac{\log(1-p)}{\log(1-(w/mn)^6)} \tag{35}$$

where p is the probability of randomly drawing a correct sample of corresponding 3-D model and image feature points, w is the number of detected image feature points that correspond to at least one 3-D model point, and m and n are the total number of 3-D model and image feature points, respectively. In case two pose solutions have the same number of inlier edges, the solution that has a lower median Euclidean distance between the inlier edge endpoints and their corresponding projected edge endpoints is chosen. For a consistent comparison of RANSAC with SVD, both architectures used point features from the same image processing subsystem. A dataset of 25 images from the PRISMA mission were used in this test, and the pose estimates from the two architectures were compared against the PRISMA flight dynamic products to calculate E_R and E_T.

Figures 18 and 19 visualize the pose solutions computed by SVD and RANSAC, respectively; whereas Table 4 presents the attitude and translation accuracy of the solutions. In particular, Table 4 presents the mean and standard deviation of E_R and E_T for the three classes of pose solutions provided by SVD. As seen in Fig. 18, SVD produced high-confidence pose solutions for images 1–5, low confidence solutions for images 6–12, and relative position solutions only for images 13–25. For images 6–25, Fig. 18 also shows the ROI detected by WGE. The architecture using RANSAC produced a pose solution for all 25 images; however, the accuracy for these solutions was lower as compared to SVD. As seen in Fig. 19, the pose solutions by RANSAC were successful in aligning the detected edges with the line segments in the 3-D model, producing meter-level relative position accuracy. However, in general, this alignment of edges does not guarantee a correct relative attitude solution, as is visible in Fig. 19 and Table 4. In contrast, the SVD architecture provided a high-confidence pose solution for five images, a low-confidence pose solution in seven images, and only a relative position solution for 13 images. The SVD high-confidence pose solutions had decimeter-level relative position accuracy and degree-level relative attitude accuracy. In comparison, the relative attitude accuracy for the SVD low-confidence pose solutions was poor due to the geometric ambiguity resulting from the low number of attitude distinguishing features detected. For example, due to the geometry of the Tango spacecraft, it was impossible to distinguish between two attitude solutions that were mirrors of each other if only a single polygonal tetrad and an antenna were detected. In these cases, the correct attitude solution was found to always be part of the set of top five solutions input to the NRM during the pose determination subsystem. However, after the NRM, the reprojection error of the correct solution was almost equal but slightly lower than the output solution. Lastly, SVD did not produce an attitude solution for 13 images because these images resulted in a pose solution with a high reprojection error. A common characteristic of these images was the detection of partial edges, duplicate edges, and the detection of edges that were absent in the 3-D model. Figure 20a shows spurious edges that were detected due to the presence of the horizon, which did not get eliminated by the WGE technique due to its sharp intensity gradient; Figs. 20b and 20c show the detection of duplicate edges as well as edges detected from parts of the spacecraft absent in its 3-D model. For both RANSAC and SVD architectures, the highest uncertainty in relative position and attitude solutions was in the C_3 direction, which was aligned with the camera boresight. The test was run on a 2.4 GHz Intel Core i5-4570T processor and made use of vectorized implementation of the SVD pose initialization architecture in MATLAB. The MATLAB command *tic* was used to start a stopwatch timer when the image processing began, whereas the command *toc* was used to stop the timer when the pose solution was output. For the 12 images in which SVD produced a pose solution, it required 8.2163 s on average. A majority of the runtime was spent in solving the feature correspondence problem, which was to be expected because the architecture did not rely on an a priori guess of the pose. In comparison, RANSAC required an average of 13.464 s for the same set of 12 images where SVD produced a pose solution.

VI. Conclusions

This paper described an architecture for robust model-based pose initialization of noncooperative spacecraft to enable autonomous proximity operations. The detailed description of the proposed techniques was accompanied by a thorough validation effort using synthetic images as well as actual space imagery from the PRISMA mission to show the functional applicability and the accuracy potential. The proposed architecture improved upon the state of the art by introducing a hybrid approach to image processing by fusing the WGE technique with the Sobel operator followed by Hough transform to detect both small and large features of the target spacecraft. The hyperparameters of the Hough transform were expressed as scalar multiples of the size of the ROI determined from the image processing subsystem, thereby alleviating the problem of manually tuning them for each image manually. The scalar multiples could easily determined using onground simulations or on board using a prior knowledge of the interspacecraft range from angles-only measurements. Comparisons with independent flight dynamics operational products have shown pose accuracy at the level of 1.5968 deg (3-D rms error) and 0.5322 m (3-D rms error). Notably, the errors in the translation and attitude in and along the boresight direction were the largest. On average, the SVD architecture required 8.2163 s to produce a pose solution when implemented in MATLAB on a 2.4 GHz Intel Core i5-4570T processor. Notably, up to 92% of the runtime was contributed by EPnP calls to determine the correct feature correspondences between the image and the 3-D model. Therefore, there would be merit in further developing techniques to detect complex feature

Downloaded by STANFORD UNIVERSITY on July 2, 2018 | http://arc.aiaa.org | DOI: 10.2514/1.A34124

groups; for example, separate Hough transforms could be employed for each complex geometric shape.

Future work will exploit features from subsequent images to estimate the pose if a complex feature group is not detected in the first image. Additionally, the determination of the initial pose at close range can be aided by the estimation of the relative orbit through azimuth and elevation angle measurements made at far range. Moreover, once the pose is determined for the first image, it will be used as an initial guess for the pose in subsequent images. This will lead to a dramatically lower computational runtime for these images because the expensive step of determining the feature correspondence between the image and the 3-D model does not have to be repeated. Output of the feature detection is processed to merge any partially detected line segments, and this has vastly improved the quality of edge detection as compared to previous work; however, further improvements must be made because it is still susceptible to producing spurious edges. This leads to the architecture producing no relative attitude solution for 13 out of 25 PRISMA images. Hence, methods for image processing using alternative features types must be explored to make it more robust. Finally, future work will incorporate the use of simulated test imagery and hardware-in-the-loop experiments to train and evaluate the SVD pose initialization architecture.

Downloaded by STANFORD UNIVERSITY on July 2, 2018 | http://arc.aiaa.org | DOI: 10.2514/1.A34124

Acknowledgments

The authors would like to thank the King Abdulaziz City for Science and Technology Center of Excellence for Research in Aeronautics and Astronautics at Stanford University for sponsoring this work. The authors would also like to thank OHB Sweden; DLR, German Aerospace Center; and the Technical University of Denmark for the PRISMA images used in this work. The authors would like to thank Joshua Sullivan of the Space Rendezvous Laboratory for his intellectual discussions regarding the weak gradient elimination technique.

References

[1] Castellani, L. T., Llorente, J. S., Ibarz, J. M. F., and Ruiz, M., "PROBA-3 Mission," *International Journal of Space Science and Engineering*, Vol. 1, No. 4, 2013, pp. 349–366.

[2] Laboratory, A. F. R., "Fact Sheet: Automated Navigation and Guidance Experiment for Local Space (ANGELS)," 2013, https://www.smallsat.org/conference/angels-fact-sheet.pdf.

[3] Reed, B. B., Smith, R. C., Naasz, B. J., Pellegrino, J. F., and Bacon, C. E., "The Restore-L Servicing Mission," *AIAA Space Forum*, AIAA Paper 2016-5478, 2016, pp. 1–8. doi:10.2514/6.2016-5478

[4] Ventura, J., Ciarcià, M., Romano, M., and Walter, U., "Fast and Near-Optimal Guidance for Docking to Uncontrolled Spacecraft," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 12, 2017, pp. 3138–3154. doi:10.2514/1.G001843

[5] Ventura, J., "Autonomous Proximity Operations for Noncooperative Space Target," Ph.D. Thesis, Technical Univ. of Munich, Munich, 2016.

[6] Spencer, D. A., Chait, S. B., Schulte, P. Z., Okseniuk, K. J., and Veto, M., "Prox-1 University-Class Mission to Demonstrate Automated Proximity Operations," *Journal of Spacecraft and Rockets*, Vol. 53, No. 5, 2016, pp. 847–863. doi:10.2514/1.A33526

[7] Bowen, J., Villa, M., and Williams, A., "CubeSat Based Rendezvous, Proximity Operations, and Docking in the CPOD Mission," *29th AIAA/USU Conference on Small Satellites*, Paper SSC15-III-5, Logan, UT, 2015.

[8] Sharma, S., and D'Amico, S., "Reduced-Dynamics Pose Estimation for Non-Cooperative Spacecraft Rendezvous Using Monocular Vision," *40th Annual AAS Guidance and Control Conference*, AAS Paper 17-073, Breckenridge, CO, 2017.

[9] Benninghoff, H., Boge, T., and Tzschichholz, T., "Hardware-in-the-Loop Rendezvous Simulation Involving an Autonomous Guidance, Navigation and Control System," *Proceedings of the 1st IAA Conference on Dynamics and Control of Space Systems*, Porto, Portugal, March 2012.

[10] Grompone, A. A., "Vision-Based 3-D Motion Estimation for On-Orbit Proximity Satellite Tracking and Navigation," Ph.D. Thesis, Naval Postgraduate School, Monterey, CA, 2015.

[11] Naasz, B. J., Van Eepoel, J., Queen, S. Z., Southward, C. M., and Hannah, J., "Flight Results from the HST SM4 Relative Navigation Sensor System," *33rd Annual AAS Guidance and Control Conference*, American Astronautical Soc. Paper 10-086, Breckenridge, CO, 2010.

[12] Chien, C.-H., and Baker, K., "Pose Estimation for Servicing of Orbital Replacement Units in a Cluttered Environment," *Proceedings of the 2004 IEEE International Conference on Robotics & Automation*, IEEE Publ., Piscataway, NJ, April 2004, pp. 5141–5146.

[13] Kim, S.-G., Crassidis, J. L., Cheng, Y., Fosbury, A. M., and Junkins, J. L., "Kalman Filtering for Relative Spacecraft Attitude and Position Estimation," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 5, 2014, pp. 1706–1711. doi:10.2514/1.G000204

[14] Valasek, J., Gunnam, K., Kimmett, J., Tandale, M. D., Junkins, J. L., and Hughes, D., "Vision-Based Sensor and Navigation System for Autonomous Air Refueling," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 5, 2005, pp. 979–989. doi:10.2514/1.11934

[15] Fosbury, A., and Crassidis, J., "Relative Navigation of Air Vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 4, 2008, pp. 824–834. doi:10.2514/1.33698

[16] Cropp, A., "Pose Estimation and Relative Orbit Determination of a Nearby Target Microsatellite Using Passive Imagery," Ph.D. Thesis, Electronic Engineering, Information Technology and Mathematics, Surrey Space Centre, Univ. of Surrey, Guildford, England, U.K., April 2001.

[17] Benn, M., "Vision Based Navigation Sensors for Spacecraft Rendezvous and Docking," Ph.D. Thesis, Danish Technical Univ., Lyngby, Denmark, 2010.

[18] Kanani, K., Petit, A., Marchand, E., Chabot, T., and Gerber, B., "Vision Based Navigation for Debris Removal Missions," *63rd International Astronautical Congress*, Paper IAC-12, A6, 5, 9, x14900, Naples, Italy, 2012.

[19] Dhome, M., Richetin, M., Lapresteé, J. T., and Rives, G., "Determination of the Attitude of 3-D Objects from a Single Perspective View," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11, No. 12, 1989, pp. 1265–1278. doi:10.1109/34.41365

[20] Gold, S., Lui, C. P., and Rangarajan, A., "New Algorithms for 2-D and 3-D Point Matching: Pose Estimation and Correspondence," *Pattern Recognition*, Vol. 31, No. 8, Aug. 1998, pp. 1019–1031. doi:10.1016/S0031-3203(98)80010-1

[21] David, P., Dementhon, D., Duraiswami, R., and Samet, H., "SoftPOSIT: Simultaneous Pose and Correspondence Determination," *International Journal of Computer Vision*, Vol. 59, No. 3, 2004, pp. 259–284. doi:10.1023/B:VISI.0000025800.10423.1f

[22] Attia, M., Slama, Y., and Kamoun, M. A., "On Performance Evaluation of Registration Algorithms for 3-D Point Clouds," *Proceedings of the 13th International Conference on Computer Graphics, Imaging and Visualization (CGiV)*, Beni Mellal, 2016, pp. 45–50. doi:10.1109/CGiV.2016.18

[23] Mirzaei, F. M., and Roumeliotis, S. I., "Globally Optimal Pose Estimation from Line-Plane Correspondences," *2011 IEEE International Conference on Robotics and Automation*, Shanghai, 2011, pp. 5581–5588. doi:10.1109/ICRA.2011.5980272

[24] Xu, C., Zhang, L., Cheng, L., and Koch, R., "Pose Estimation from Line Correspondences: A Complete Analysis and a Series of Solutions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 39, No. 6, 2017, pp. 1209–1222. doi:10.1109/TPAMI.2016.2582162

[25] Ansar, A., and Daniilidis, K., "Linear Pose Estimation from Points or Lines," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 5, 2003, pp. 578–589. doi:10.1109/TPAMI.2003.1195992

[26] Lepetit, V., Moreno-Noguer, F., and Fua, P., "EPnP: An Accurate O(n) Solution to the PnP Problem," *International Journal of Computer Vision*, Vol. 81, No. 2, 2009, pp. 155–166. doi:10.1007/s11263-008-0152-6

[27] Dementhon, D. F., and Davis, L. S., "Model-Based Object Pose in 25 Lines of Code," *International Journal of Computer Vision*, Vol. 15, Nos. 1–2, 1995, pp. 123–141. doi:10.1007/BF01450852

[28] Petit, A., Kanani, K., and Marchand, E., "Vision-Based Detection and Tracking for Space Navigation in a Rendezvous Context," *International Symposium on Artificial Intelligence, Robotics and Automation in Space, i-SAIRAS*, 2012.

[29] Gonzalez, R. C., and Woods, R. E., *Digital Image Processing*, Prentice–Hall, Upper Saddle River, NJ, 2002, pp. 701–704.

[30] Harris, C., and Stephens, M., "A Combined Corner and Edge Detector," *Proceedings of the Alvey Vision Conference*, 1988, pp. 147–151.
doi:10.5244/C.2.23

[31] D'Amico, S., Benn, M., and Jørgensen, J. L., "Pose Estimation of an Uncooperative Spacecraft from Actual Space Imagery," *International Journal of Space Science and Engineering*, Vol. 2, No. 2, 2014, pp. 171–189.
doi:10.1504/IJSPACESE.2014.060600

[32] Duda, R. O., and Hart, P. E., "Use of the Hough Transform to Detect Lines and Cures in Pictures," *Communications of the Association Computing Machinery*, Vol. 15, No. 1, 1972, pp. 11–15.

[33] Bodin, P., Noteborn, R., Larsson, R., Karlsson, T., D'Amico, S., Ardaens, J.-S., Delpech, M., and Berges, J. C., "PRISMA Formation Flying Demonstrator: Overview and Conclusions from the Nominal Mission," *35th Annual AAS Guidance and Control Conference*, AAS Paper 12-072, 2012.

[34] Fischler, M. A., and Bolles, R. C., "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Communications of the ACM*, Vol. 24, No. 6, 1981, pp. 381–395.
doi:10.1145/358669.358692

[35] Yilmaz, A., Javed, O., and Shah, M., "Object Tracking," *ACM Computing Survey*, Vol. 38, No. 4, 2006, Paper 13-es.
doi:10.1145/1177352

[36] Canny, J., "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, No. 6, 1986, pp. 679–698.
doi:10.1109/TPAMI.1986.4767851

[37] Matas, J., Chum, O., Urban, M., and Pajdla, T., "Robust Wide Baseline Stereo from Maximally Stable Extremal Regions," *Image and Vision Computing*, Vol. 22, No. 10, 2004, pp. 761–767.
doi:10.1016/j.imavis.2004.02.006

[38] Heikkilä, J., and Silvén, O., "A Four-Step Camera Calibration Procedure with Implicit Image Correction," *Computer Vision and Pattern Recognition*, IEEE Publ., Piscataway, NJ, pp. 1106–1112.
doi:10.1109/CVPR.1997.609468

[39] Prewitt, J., "Object Enhancement and Extraction," *Picture Processing and Psychopictorics*, Vol. 10, No. 1, 1970, pp. 15–19.

[40] Barinova, O., Lempitsky, V., and Kohli, P., "On Detection of Multiple Object Instances Using Hough Transform," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 34, No. 9, 2012, pp. 1773–1784.
doi:10.1109/TPAMI.2012.79

[41] Sullivan, J., Koenig, A., and D'Amico, S., "Improved Maneuver-Free Approach to Angles-Only Navigation for Space Rendezvous," *26th AAS/AIAA Space Flight Mechanics Conference*, AAS Paper 16-530, Napa, CA, 2016.

[42] Lowe, D. G., "Three-Dimensional Object Recognition from Single Two-Dimensional Images," *Artificial Intelligence*, Vol. 31, No. 3, 1987, pp. 355–395.
doi:10.1016/0004-3702(87)90070-1

[43] Sharma, S., and D'Amico, S., "Comparative Assessment of Techniques for Initial Pose Estimation Using Monocular Vision," *Acta Astronautica*, Vol. 123, June 2016, pp. 435–445.
doi:10.1016/j.actaastro.2015.12.032

[44] Borotschnig, H., Paletta, L., Prantl, M., and Pinz, A., "Appearance-Based Active Object Recognition," *Image and Vision Computing*, Vol. 18, No. 9, 2000, pp. 715–727.
doi:10.1016/S0262-8856(99)00075-X

[45] Byne, J., and Anderson, J., "A CAD-Based Computer Vision System," *Image and Vision Computing*, Vol. 16, No. 8, 1998, pp. 533–539.
doi:10.1016/S0262-8856(98)00100-0

[46] Cyr, C. M., and Kimia, B. B., "3-D Object Recognition Using Shape Similarity-Based Aspect Graph," *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, Vol. 1, Vancouver, BC, 2001, pp. 254–261.
doi:10.1109/ICCV.2001.937526

[47] Grimson, W. E. L., and Huttenlocher, D. P., "On the Verification of Hypothesized Matches in Model-Based Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 12, 1991, pp. 1201–1213.
doi:10.1109/34.106994

[48] Ardaens, J. S., D'Amico, S., and Montenbruck, O., "Final Commissioning of the Prisma GPS Navigation System," *Journal of Aerospace Engineering, Sciences and Applications*, Vol. 4, No. 3, 2012, pp. 104–118.
doi:10.7446/jaesa

[49] Rosten, E., and Drummond, T., "Machine Learning for High-Speed Corner Detection," *Computer Vision–ECCV 2006*, Vol. 3951, Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Springer, New York, 2006, pp. 430–443.
doi:10.1007/11744023_34

[50] Leutenegger, S., Chli, M., and Siegwart, R. Y., "BRISK: Binary Robust Invariant Scalable Keypoints," *Proceedings of the IEEE International Conference on Computer Vision*, IEEE Publ., Piscataway, NJ, pp. 2548–2555.
doi:10.1109/ICCV.2011.6126542

[51] Dalal, N., and Triggs, B., "Histograms of Oriented Gradients for Human Detection," *Proceedings—2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, Vol. I, IEEE Publ., Piscataway, NJ, 2005, pp. 886–893.
doi:10.1109/CVPR.2005.177

[52] Gao, X. S., Hou, X. R., Tang, J., and Cheng, H. F., "Complete Solution Classification for the Perspective-Three-Point Problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 8, 2003, pp. 930–943.
doi:10.1109/TPAMI.2003.1217599

J. A. Christian
Associate Editor

Downloaded by STANFORD UNIVERSITY on July 2, 2018 | http://arc.aiaa.org | DOI: 10.2514/1.A34124