Acta Astronautica 123 (2016) 435–445

Contents lists available at ScienceDirect

Acta Astronautica

journal homepage: www.elsevier.com/locate/actaastro

Comparative assessment of techniques for initial pose estimation using monocular vision

Sumant Sharma *, Simone D'Amico

Department of Aeronautics & Astronautics, Stanford University, Stanford, California, USA

ARTICLE INFO

Article history:
Received 22 July 2015
Received in revised form
12 December 2015
Accepted 23 December 2015
Available online 20 January 2016

Keywords:
Computer vision
Pose estimation
Monocular vision
Uncooperative spacecraft

ABSTRACT

This work addresses the comparative assessment of initial pose estimation techniques for monocular navigation to enable formation-flying and on-orbit servicing missions. Monocular navigation relies on finding an initial pose, i.e., a coarse estimate of the attitude and position of the space resident object with respect to the camera, based on a minimum number of features from a three dimensional computer model and a single two dimensional image. The initial pose is estimated without the use of fiducial markers, without any range measurements or any apriori relative motion information. Prior work has been done to compare different pose estimators for terrestrial applications, but there is a lack of functional and performance characterization of such algorithms in the context of missions involving rendezvous operations in the space environment. Use of state-of-the-art pose estimation algorithms designed for terrestrial applications is challenging in space due to factors such as limited on-board processing power, low carrier to noise ratio, and high image contrasts. This paper focuses on performance characterization of three initial pose estimation algorithms in the context of such missions and suggests improvements.

1. Introduction

Recent advancements have been made to utilize monocular vision navigation as an enabling technology for formation-flying and on-orbit servicing missions (e.g., PROBA-3 by ESA [1] , ANGELS by US Air Force [2], PRISMA by OHB Sweden [3]). These missions require approaching a passive space resident object from large distances (e.g., $>$ 30 km) in a fuel efficient, safe, and accurate manner. Simple modification of low cost instruments (e.g., star trackers) for high dynamic range can enable accurate navigation relative to the space resident object. Monocular navigation on such missions relies on finding an estimate of the initial pose, i.e., the attitude and position of the space resident object with respect to the camera, based on a minimum number of features from a three dimensional computer model and a single two dimensional image. For on-orbit servicing missions, this represents the scenario where the servicing spacecraft is "lost-in-space". Estimating the initial pose is especially critical as well as challenging in the design of a pose estimation system as there is no a priori information about the attitude and position of the target. Aside from a 3D wire-frame model of the space resident object, no assumption on the relative translational or rotational information is made.

Use of state-of-the-art computer vision techniques designed for terrestrial applications is challenging in space. For example, use of feature descriptors such as the Scale Invariant Feature Transform (SIFT) [4] in pose estimation for space imagery is too computationally expensive and yields poor results (see Fig. 1). We plot SIFT feature matches (in green) between two images taken during the

* Corresponding author. Tel.: +1 404 952 8102.
E-mail address: sharmas@stanford.edu (S. Sharma).

http://dx.doi.org/10.1016/j.actaastro.2015.12.032
0094-5765/

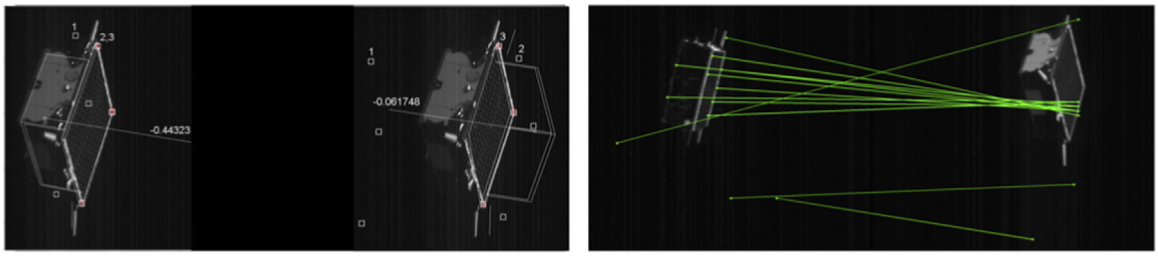

Fig. 1. Challenges in initial pose estimation using state-of-the-art techniques: pose ambiguity due to geometry (left) and poor feature matching (right).

PRISMA mission. To obtain correct feature matches, SIFT relies on high image acquisition rates and low image noise, both of which are unavailable in these images. Additionally, spacecraft geometry is often highly symmetrical resulting in ambiguity in attitude determination (see Fig. 1). Prior work has been done to compare different initial pose estimators [5,6], but there is a lack of functional and performance characterization of such algorithms in the context of missions involving rendezvous operations in the space environment.

For the purpose of this comparative assessment, we focus on a pose estimation architecture (see Fig. 2) based on points extracted from edge features due to their simplicity and accuracy [7]. As opposed to features based on color gradients, textures, and optical flow, edges are less sensitive to illumination changes and can easily distinguish boundaries of a spacecraft geometry from the background in an image.

Source of the 3D model features is a simplified wireframe model of the space resident object, assumed to be either stored on-board the servicing spacecraft or formed as a structure-from-motion problem which is solved alongside pose estimation [8]. Source of the 2D features is an object detection subsystem which processes, extracts, and describes features from a single image captured by the on-board navigation camera. A typical model reduction procedure and object detection subsystem are described and illustrated later in the paper.

The 3D model features and 2D features are passed into the initial pose estimation subsystem which generates a pose estimate without knowing the correspondence of these features. This is a challenging task without apriori estimates of the relative motion due to a large search space for the correct feature correspondence. However, the search space can be reduced using a method such as perceptual organization [9,10] which detects viewpoint-invariant feature groupings from the 2D image. These are then matched to corresponding structures of the 3D model in a probabilistic manner to create multiple correspondence hypotheses. These correspondence hypotheses need to be validated to find the correct one. Hence, for each correspondence hypothesis, n number of 2D and 3D features are used to calculate a relative pose by solving the Perspective-n-Point Problem (PnP) [11]. The resulting pose estimate is used to create virtual 2D image features by reprojecting the input 3D model features using true perspective projection. A measure of the reprojection error between the virtual 2D image features and the input 2D image features is used in validation. This process is repeated for all hypotheses in order to identify the correct feature correspondence and subsequently, a correct pose estimate. Hence, our interest is not in the PnP solvers' statistical use of a large number of measurements to solve an overdetermined systems. Rather, it is in their ingenuity in using a minimal number of points to estimate a coarse initial pose estimate with a minimal computational effort.

The performance of initial pose estimation hinges on the solution of the PnP problem. In the above architecture, a PnP solver could be called multiple times and will be subject to a wide variety of input from the object detection subsystem and 3D model reduction. Hence, a PnP solver should not only be fast and efficient but also more importantly be reliable and robust to overcome challenges unique to monocular vision-based navigation in space. In the remainder of the paper, we first present a formal problem statement of the PnP problem and then review state-of-the-art PnP solvers. We then introduce our framework of assessment of these solvers where a discussion of simulation input generation, performance criteria, and test cases is presented. Finally, we present relevant results from our assessment and conclude with a discussion on applicability of these solvers in a monocular vision-based navigation system for on-orbit servicing and formation flying missions.

2. Review of solution methods

Let $q_i = [x_i\ y_i\ z_i]^T$, where $i = 1, 2, \ldots, n$, be n 3D model points in the object reference framework B. Let $p_i = [u_i\ v_i\ 1]^T$, where $i = 1, 2, \ldots, n$, be the corresponding n image points in the image reference framework P. For a known camera focal length, f, the PnP problem aims to retrieve the rotation matrix from frame B to the camera reference framework C, R_{BC}, and the translation vector from the origin of frame B to the origin of frame C, T_{BC}:

$$p_i = [u_i\ v_i\ 1]^T = \left[\frac{\alpha_i}{\gamma_i} f\ \frac{\beta_i}{\gamma_i} f\ 1\right]^T$$

$$r_i = [\alpha_i\ \beta_i\ \gamma_i]^T = R_{BC}(q_i + T_{BC}) \tag{1}$$

Re-writing Eq. (1) using homogeneous coordinates and representing camera parameters as a matrix K, we would like to estimate the 3×4 pose matrix, P, whose first three

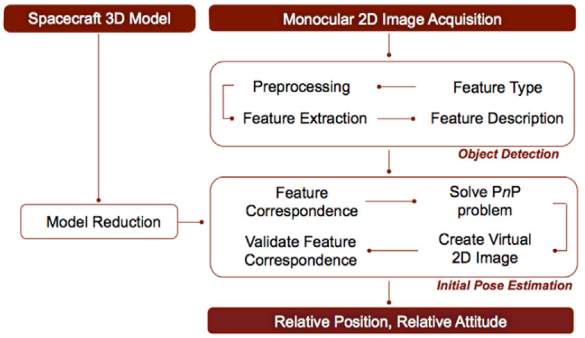

Fig. 2. Typical pose estimation architecture with inputs of a spacecraft 3D model and a 2D image and an output of a pose estimate.

columns represent R_{BC} and the fourth column represents T_{BC}.

$$\begin{bmatrix} w_i u_i \\ w_i v_i \\ w_i \end{bmatrix} = \begin{bmatrix} \\ K \\ \\ \end{bmatrix} \begin{bmatrix} \\ P \\ \\ \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} \tag{2}$$

Without loss of generality, we can expand Eq. (2) to obtain Eq. (3), assuming square pixels in the image sensor, zero skewness, and zero distortion:

$$\begin{bmatrix} w_i u_i \\ w_i v_i \\ w_i \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_{BC}(1,1) & R_{BC}(1,2) & R_{BC}(1,3) & T_{BC}(1) \\ R_{BC}(2,1) & R_{BC}(2,2) & R_{BC}(2,3) & T_{BC}(2) \\ R_{BC}(3,1) & R_{BC}(3,2) & R_{BC}(3,3) & T_{BC}(3) \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} \tag{3}$$

Eq. (3) has six unknown coefficients as there are six degrees of freedom in P. Three degrees are required to describe the relative attitude and three are required to describe the relative position. Hence, three image points, i.e., $n=3$ will provide six measurements which can be used to solve six equations in six unknowns. However, there will be four possible solutions as the conversion to non-homogeneous coordinates is non-linear. In fact, it has been shown that a unique solution with a general configuration of points is only possible with $n=6$ [11].

The P*n*P problem has been systematically investigated in literature and some of the most commonly used P*n*P solvers are PosIt and Coplanr PosIt [12], EP*n*P [13], OP*n*P[14], and Lu-Hager-Mjolsness method [15]. These solvers assume that the correspondence between model points and observed image points has already been established. This is not true in our case which makes the problem of initial pose estimation especially challenging. Note that even when a unique solution is theoretically available for the P*n*P problem, there is no guarantee that a unique solution actually exists in the case of symmetrical spacecraft geometries. For example, a P*n*P solver can provide six different solutions for a cube with six identical sides. Hence, we need to carry along multiple solutions of P*n*P even when we have more than six measurements due to the symmetry. State-of-the-art P*n*P solvers usually take either an iterative minimization approach or a multi-stage analytical approach. The multi-stage analytical approaches estimate coordinates of some or all points in the camera frame and then solve the problem in 3D space using linear forms of the perspective Eq. (1). This is done to make computation fast for large number of measurements at the expense of accuracy for small number of measurements. Other solvers that utilize non-linear constraints posed by Eq. (1) tend to be computationally expensive. The iterative minimization approaches minimize an error function defined in the image or object space. They typically take non-linear constraints of Eq. (1) into account but tend to get trapped in local minima. Four of these state-of-the-art solvers, namely PosIt, Coplanar PosIt, EP*n*P, and the Newton–Raphson Method [16,17] are discussed in more detail in the following sections to provide analytical reasoning behind their performance in initial pose estimation. PosIt, Coplanar PosIt, and the Newton–Raphson Method take an iterative minimization approach while EP*n*P takes a multi-stage analytical approach. Note that we exclude our assessment of the Lu-Hager-Mjolsness method [15] as it is also an iterative minimization approach which is shown to be approximately four times slower than the Newton–Raphson Method [18].

2.1. PosIt and Coplanar PosIt

PosIt [12] requires a minimum of four non-coplanar 3D model points and corresponding 2D image points for pose calculation. It approximates true perspective projection with a Scaled Orthographic Projection (SOP) to form a coarse estimate of the pose which is then iteratively improved until convergence to a solution. The algorithm for PosIt scales both sides of Eq. (3) by $1/T_{BC}(3)$. Then, expanding the first two rows yields the following relationship, where $s=f/T_{BC}(3)$:

$$x_i\, sR_{BC}(1,1)+y_i\, sR_{BC}(1,2)+z_i\, sR_{BC}(1,3)+sT_{BC}(1)=u_i\,\frac{w_i}{T_{BC}(3)} \tag{4}$$

$$x_i\, sR_{BC}(2,1)+y_i\, sR_{BC}(2,2)+z_i\, sR_{BC}(2,3)+sT_{BC}(2)=v_i\,\frac{w_i}{T_{BC}(3)} \tag{5}$$

PosIt initializes with the SOP assumption, i.e., $\frac{w_i}{T_{BC}(3)}=1$, so that (4) and (5) can be simplified to linear equations with only eight unknowns, i.e., $sR_{BC}(1,1)$, $sR_{BC}(1,2)$, $sR_{BC}(1,3)$, $sT_{BC}(1)$, $sR_{BC}(2,1)$, $sR_{BC}(2,2)$, $sR_{BC}(2,3)$, and $sT_{BC}(2)$. With $n=4$, these eight unknowns can be solved using the eight linear equations formed by writing (Eqs. (4) and 5) for $i=1,2,3$, and 4. Then using the resulting estimate of the pose matrix, the value of $\frac{w_i}{T_{BC}(3)}$ can be updated at the end of each iteration. For each iteration, the reprojected model points are calculated by using the current estimate of the pose matrix in Eq. (3). At the end of each iteration, an improvement score, ΔE, is calculated by measuring the Euclidean distance between the reprojected model points of the current and previous iterations.

$$\Delta E=\sum_{i=1}^{n} f\sqrt{\Delta u_i^2+\Delta v_i^2} \tag{6}$$

Iterations are stopped if either ΔE falls below 0.1 or 1000 iterations are reached. At low focal lengths or when the space resident object is close to the camera, SOP is not

a valid approximation of true perspective projection and leads to inaccurate solutions. However, we can expect to see a trend of improvement in PosIt's "performance" (relative to other solvers) as SOP begins to closely approximate true perspective projection when the space resident object is far from the camera.

Coplanar PosIt [19] is a variation of PosIt which exclusively addresses the case of coplanar 3D model points and corresponding 2D image points. It acknowledges that an SOP approximation leads to two possible solutions of R_{BC} due to an additional degree of freedom. This is addressed by keeping track of the solution with the lower reprojection error for a given iteration. The reprojection error is the average Euclidean distance between the measured image points, p_i, and the reprojected model points obtained by using the estimate pose matrix in Eq. (2). Similar to PosIt, the solutions are iteratively refined until ΔE of one of the branches falls below 0.1. A branch of the solutions is discarded if it estimates any of the 3D model points to be behind the camera. For a given iteration, if the two possible solutions lie close to each other in the solution space, only refining the solution with the lower reprojection error does not guarantee convergence to a solution with the eventual lower reprojection error. This is a major shortcoming of Coplanar PosIt and can only be remedied by sacrificing computational efficiency and tracking all possible solution branches.

2.2. EPnP

Applicable to both coplanar and non-coplanar 3D model points, EPnP [20] attempts to find a closed-form pose solution and requires a minimum of four corresponding model and image points. The main idea is to express the 3D model points as a weighted sum of four non-coplanar virtual control points, c_j, where $j=1,2,3$, and 4, in the object frame B (see Fig. 3). Using a matrix inversion, the homogeneous barycentric coordinates, α_{ij}, can be computed from the n 3D model points, q_i, assuming arbitrary coordinates of the control points.

$$q_i = \sum_{j=1}^{4} \alpha_{ij} c_j \tag{7}$$

Using Eq. (7), Eq. (3) can be re-written in terms of the 12 unknown control point coordinates in the camera frame, $[\hat{\alpha}_j,\ \hat{\beta}_j,\ \hat{\gamma}_j]^T$, where $j=1,2,3$, and 4:

$$\begin{bmatrix} w_i u_i \\ w_i v_i \\ w_i \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \sum_{j=1}^{4} \alpha_{ij} \begin{bmatrix} \hat{\alpha}_j \\ \hat{\beta}_j \\ \hat{\gamma}_j \end{bmatrix} \tag{8}$$

By substituting the values of w_i from the third row into the first two rows of Eq. (8), two linear equations are formed for each corresponding pair of a 3D model point and an image point:

$$\sum_{j=1}^{4} \alpha_{ij} f \hat{\alpha}_j - \alpha_{ij} u_i \hat{\gamma}_j = 0 \tag{9}$$

$$\sum_{j=1}^{4} f \alpha_{ij} \hat{\beta}_j - \alpha_{ij} v_i \hat{\gamma}_j = 0 \tag{10}$$

Hence, the $2n$ linear equations which result from writing (Eqs. (9) and 10) for $i=1,2,\ldots,n$, can be used to estimate the 12 unknowns in the form of a 12×1 vector, $\hat{x}$, by solving a linear system of the form $M\hat{x}=0$. Note that M is a $2n\times 12$ matrix formed by arranging the coefficients of the $2n$ linear equations. When the image points are perfect data from a true perspective projection of the model points, the solution is calculated as a 12×1 vector, $\hat{x}$, in the null-space of the 12×12 matrix, M^{TM}. However, the matrix may have as many as four linearly dependent columns due to measurement noise and/or the focal length of the camera. There could be four possible solutions expressed as linear combination of the eigenvectors of this matrix. Out of the four possible solutions, the one with the lowest reprojection error is selected. Typically, this is inaccurate when the focal length is large and the matrix is poorly conditioned. In this scenario, even a small amount of measurement noise could lead to highly inaccurate estimates of the solution.

2.3. Newton–Raphson Method

The Newton–Raphson Method (NRM) [16,17] iteratively solves the true perspective equations, given by Eq. (1), for an estimate of the pose. It requires an input guess of the pose and a minimum of three corresponding 3D model and 2D image points. For each iteration, the reprojection error is linearized about the current pose estimate, $\vec{x}=[T_{BC}\ \varphi_{BC}]^T$. Note that $\vec{x}$ is a 6×1 vector containing the three components of the translation vector, T_{BC} (see Fig. 3), and the three Euler angles, φ_{BC}, representing the rotation from frame B to frame C:

$$E_{RE,i} = \frac{\partial p_i}{\partial r_i}\frac{\partial r_i}{\partial T_{BC}}\Delta T_{BC} + \frac{\partial p_i}{\partial r_i}\frac{\partial r_i}{\partial \varphi_{BC}}\Delta\varphi_{BC} \tag{11}$$

The derivatives of the perspective equations with respect to T_{BC} and φBC are evaluated in the form of a $2n\times 6$ Jacobian matrix, $J=[J_1,J_2,\ldots,J_n]^T$:

$$J_i = \begin{bmatrix} \frac{\partial p_i}{\partial r_i}\frac{\partial r_i}{\partial T_{BC}} & \frac{\partial p_i}{\partial r_i}\frac{\partial r_i}{\partial \varphi_{BC}} \end{bmatrix} \tag{12}$$

For each iteration, the least squares solution by using the reprojection error from the previous iteration, E_{RE}, and the Jacobian matrix provides an update to the pose estimate, $\Delta\vec{x}$. Note that E_{RE} is a $2n\times 1$ vector formed by writing Eq. (11) for $i=1,2,\ldots,n$:

$$\Delta\vec{x} = (J^T J)^{-1} J^T E_{RE} \tag{13}$$

The pose estimate is refined until either $|\Delta\vec{x}|$ falls below 10^{-10} or 50 iterations are reached. To examine robustness to initialization with coarse pose estimates, we conducted a simulation with synthetic 3D model and 2D image points. For a fixed set of six 3D model points, we varied the initial guess for R_{BC} 10^4 times and generated 10^4 corresponding sets of six image points. Each time the initial guess for R_{BC} was generated from random values of the three Euler angles drawn from the uniform distributions of $[-180^\circ, 180^\circ]$, $[-180^\circ, 180^\circ]$, and $[0^\circ, 180^\circ]$. After

computing a pose estimate for each corresponding set of six 3D model points and six image points, it was observed that given the true value of T_{BC}, NRM typically converged to the correct solution when the initial guess for R_{BC} was within 60 ° of the true value of R_{BC}.

3. Framework for comparative assessment

For initial pose estimation, P*n*P solvers will be called multiple times to find the correct correspondence hypothesis. Since there exist multiple ways to generate correspondence hypotheses, we decouple their performance from that of P*n*P solvers by using a minimal number of matches between the 3D model and the 2D image in these simulations as inputs. Monte-Carlo simulations are used to rigorously inspect performance of the P*n*P solvers in the context of initial pose estimation. It is imperative to define performance criteria, test cases and generate simulation input data that exhaustively represent scenarios of spaceborne applications. The simulations made use of vectorized implementations of all P*n*P solvers and were carried out on a 2.4 GHz Intel Core i5 processor. In order to assess potential implementation of these solvers on a state-of-the-art spaceborne microprocessor running at clock-rates of 30–300 MHz, the profiling results have been scaled appropriately. Since Coplanar PosIt only accepts coplanar 3D model points while PosIt only accepts non-coplanar 3D model points, for each simulation we check coplanarity of the input and switch to the more suitable of the two solvers. Results for this combination of solvers are labeled as "PosIt+".

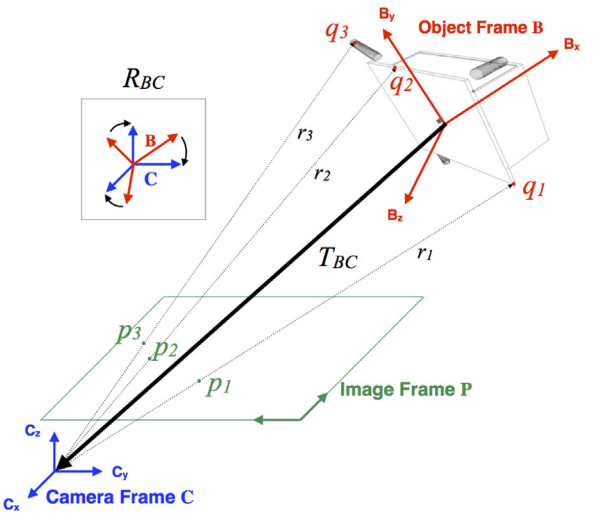

Fig. 3. Geometric representation of the Perspective-*n*-Point problem.

3.1. Simulation input generation

The three inputs of the P*n*P solvers are n 3D model points extracted from a wire-frame model, camera intrinsic parameters, and n 2D image points extracted from a virtual image of the wire-frame model. For NRM, an input guess of the pose is also required. We select a random attitude guess within $\pm$ 60° of the true attitude and a random position guess with a magnitude within $\pm$ 30% of the magnitude of the true position.

Object 3D Model: The 3D model needs to carry a high number of features to reduce ambiguities associated with the symmetry of spacecraft but also be simplified enough to boost the efficiency of the search algorithms during feature correspondence and pose estimation. With this consideration, a wire-frame model derived from a high fidelity CAD model of the Tango spacecraft from the PRISMA mission is used in this paper (see Fig. 4). The model is input in MATLAB as a stereolithographic file from which information about surfaces and edges is generated. Duplicate edges as

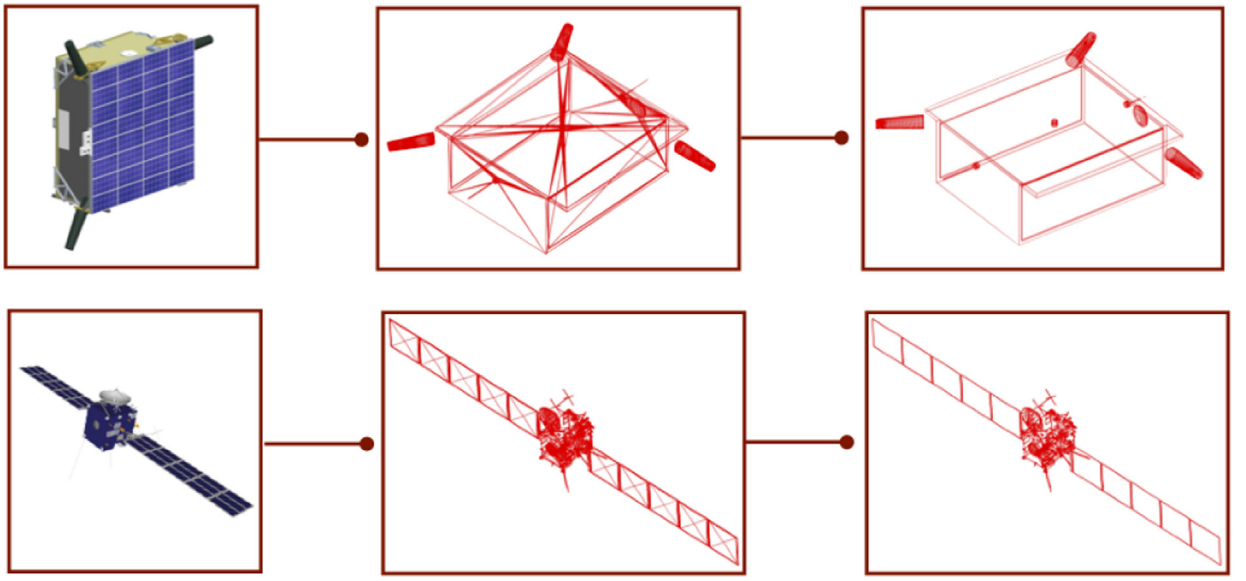

Fig. 4. Typical model reduction steps to obtain simplified wire-frame model from CAD model of Tango spacecraft (top). Output of the same steps for Rosetta spacecraft (bottom).

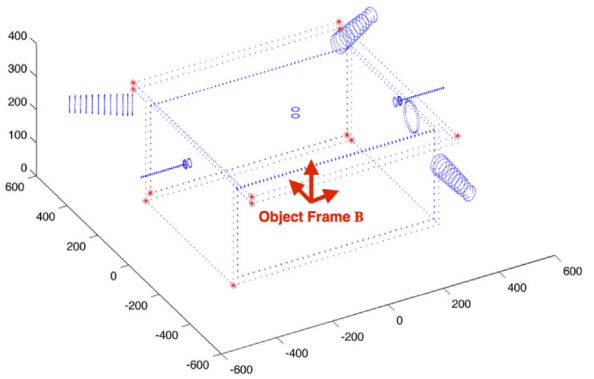

Fig. 5. Sample simulation input of 3D model points (shown in red). The discretized 3D model (shown in blue) has been plotted as a reference. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

well as edges on flat surfaces are removed and the remaining edges are discretized as 3D points. Based on an empirically determined probability distribution, n 3D points are selected at random for each simulation and used as an input to the PnP solvers (see Fig. 5). Although not used in the comparative assessment of the PnP solvers in our work, we present the results of our model reduction procedure on a publicly available CAD model of the Rosetta spacecraft [21] to show the wide applicability of our method (see Fig. 4). The model reduction procedure can potentially be used for pre-launch validation of pose estimation techniques when a CAD model of the target spacecraft is available, or during orbit when the 3D model of the target space resident object can be generated using Structure-from-Motion (SfM) techniques.

Camera Model: A virtual pinhole camera model is adopted to model the close-range vision camera embarked on the servicer spacecraft of the PRISMA mission [16]. The effective focal length is $20{,}187 \cdot 10^{-6}$ m for both axes of the image sensor which produces images 752 px $\times$ 580 px in size. It is assumed that the images are produced with zero skewness and zero distortion. Origin of the camera frame C coincides with the origin of the image frame P, with the optical axis aligned with the z-axis. The internal calibration matrix of the camera is as follows:

$$K = \begin{bmatrix} 2347 & 0 & 0 \\ 0 & 2432 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{14}$$

Typically, the camera will be calibrated on-ground but it can also be calibrated on-board by treating the intrinsic parameters as five additional unknowns of pose estimation.

Object 2D Image: Simulation input for n image points is obtained through true perspective projection (Eq. (1)) of the selected n 3D model points using the virtual camera (see Fig. 6).

3.2. Performance criteria

Four criteria are used to give a quantitative definition to accuracy, speed, and robustness of the PnP solvers. These criteria are computational runtime of the solver, image plane error, translation error, and rotation error in the pose estimate output by the solver in comparison to the ground truth used to generate simulation input.

Computational runtime: MATLAB command *tic* is used to start a stopwatch timer when a PnP solver is called whereas the command *toc* is used to stop the timer. Thus, the time elapsed on the timer is reported as the computational runtime to estimate the efficiency of the solvers relative to each other. This runtime is then used to provide a coarse first-order approximation of the runtime on a spaceborne microprocessor with a clock-rate of 30 MHz. Since we are interested in implementing a realtime pose estimation architecture on-board a spacecraft microprocessor, it is essential for PnP solvers to consume minimal computational resources.

Image Plane error: Also referred to as reprojection error, it is defined as the mean Euclidean distance in pixels between input 2D image points p_i and the corresponding virtual 2D image points p_i' constructed from the 3D model points and the pose estimate from the PnP solver

$$E_{2D} = \frac{1}{n}\sum_{i=1}^{n} |p_i - p_i'| \tag{15}$$

Translation error: As a measure of accuracy in the estimation of relative position, we define translation error as the percentage difference between the magnitudes of the true translation vector and the estimated translation vector

$$E_T = \frac{|\vec{T}_{true}| - |\vec{T}_{est}|}{|\vec{T}_{true}|} \cdot 100 \tag{16}$$

Rotation error: Unit quaternion representation of the true rotation matrix, q_{true}, and the estimated rotation matrix, q_{est}, is used to compute the rotation error. We use quaternion algebra to compute a unit quaternion, q_{diff}, which represents the relative rotation between q_{true} and q_{est}. The rotation error is expressed as an equivalent angle and reported in degrees. Note that $q_{diff}(4)$ represents the scalar component of the unit quaternion:

$$E_R = 2\ \cos^{-1}(q_{diff}(4)) \tag{17}$$

3.3. Test cases

We develop four different test cases to represent the range of possible input to PnP solvers. Our main idea is to decouple the effects of the Earth's shadow, inter-spacecraft distance, background interaction, and measurement noise on the performance of the PnP solvers. Different levels of the Earth's shadow on the target spacecraft as well as the spacecraft orientation relative to the sun will vary the number of features detected through image processing. We simulate this effect by varying the number of feature correspondences, n, being passed as an input to the PnP solvers. Measurement noise due to the image sensor

characteristics is simulated by adding random Gaussian noise with zero mean and a standard deviation, σ_P, to the simulation input of the image points being passed into the P*n*P solvers. Background interaction may lead to features being detected not on the target spacecraft. This effect is simulated by changing the percentage of outlier image points, $\tilde{p}$. An outlier image point in the simulations is defined as an image point with a measurement noise greater than $5\sigma_P$.

To simulate these effects, we require values of n, σ_P, and $\tilde{p}$ which are representative of images taken during spaceborne applications of monocular vision. In order to make these values as representative as possible, we refer to the series of images captured by visual navigation cameras of the Orbital Express [22] and PRISMA [23] missions. We performed edge detection followed by Hough transform [24] to simulate the output of a state-of-the-art object detection subsystem. Typical results are shown in Fig. 8 with intermediate steps shown in Fig. 7.

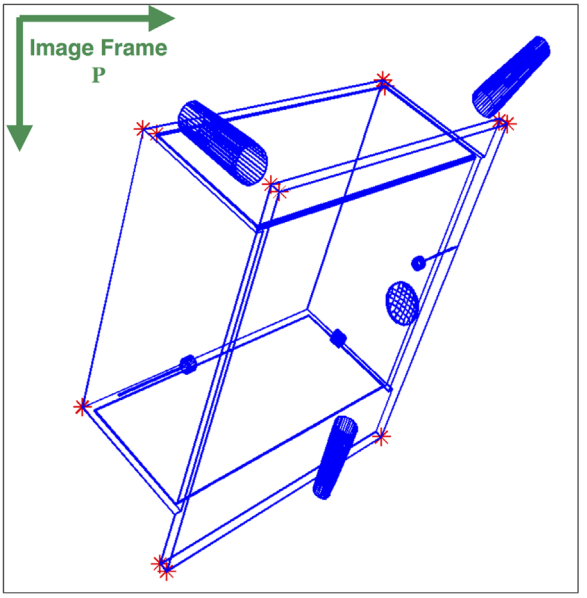

Fig. 6. Sample simulation input of 2D image points (shown in red). The 3D model (shown in blue) has been plotted as a reference. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

These results are used to empirically determine the range of n, σ_P, and $\tilde{p}$ used in the following test cases.

Test Case 1: *number of feature correspondences*. We are interested to gauge the performance of P*n*P solvers with an input of minimal number of feature correspondences as these solvers will be used on small sets of input to statistically identify outliers and identify the correct feature correspondence. As our first test case, we perform simulations with varying number of feature correspondences from a minimum of three to a maximum of twelve correspondences. For each value of n, 10^4 simulations are run, each with a different input set of 3D model points. The true pose is kept constant for each simulation in this test case to negate the effect of the geometry of the space resident object. Also note that feature correspondences are perfect, i.e., the input set of the 3D model points is correctly mapped to the corresponding input set of the 2D image points.

Test Case 2: *pixel location noise*. The 2D image points can be modeled as a sum of the true pixel locations and the pixel location noise. The pixel location noise can be modeled as a Gaussian distribution in two dimensions with a standard deviation of σ_P which is shown to be proportional to image intensity noise σ_I for state-of-the-art object detection algorithms [25]. Since we can estimate σ_I through a principal component analysis of homogeneous grayscale image patches [26], we can estimate the expected values of σ_P in a spaceborne application. Hence, as our second test case, we perform 10^4 simulations with varying values of pixel location noise. Noise is characterized as a Gaussian distribution with a mean of zero and a standard deviation of σ_P. This noise is added to the 2D image points generated from the true perspective projection of six 3D model points. The true pose is kept constant and feature correspondences are perfect for all simulations in this test case. For a fair comparison, we limit the number of iterations for PosIt and NRM so that computational runtime is comparable to the closed-form solver EP*n*P.

Test Case 3: *outliers*. Feature correspondence of input 3D model points and 2D image points is prone to contain outliers due to errors in the object detection subsystem or simply due to the geometry of the space resident. An object detection subsystem based on edges can output partial edges due to illumination conditions and/or false edges due to background interaction (see Fig. 8). Such an output from the object detection subsystem can result in an incorrect feature correspondence. Moreover, a probabilistic approach

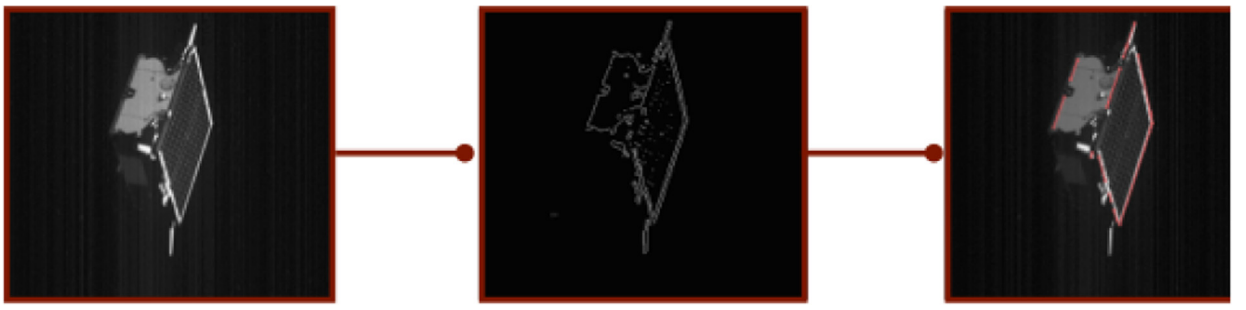

Fig. 7. Intermediate steps of object detection subsystem based on edges: preprocessing (left), feature extraction (middle), feature description (right).

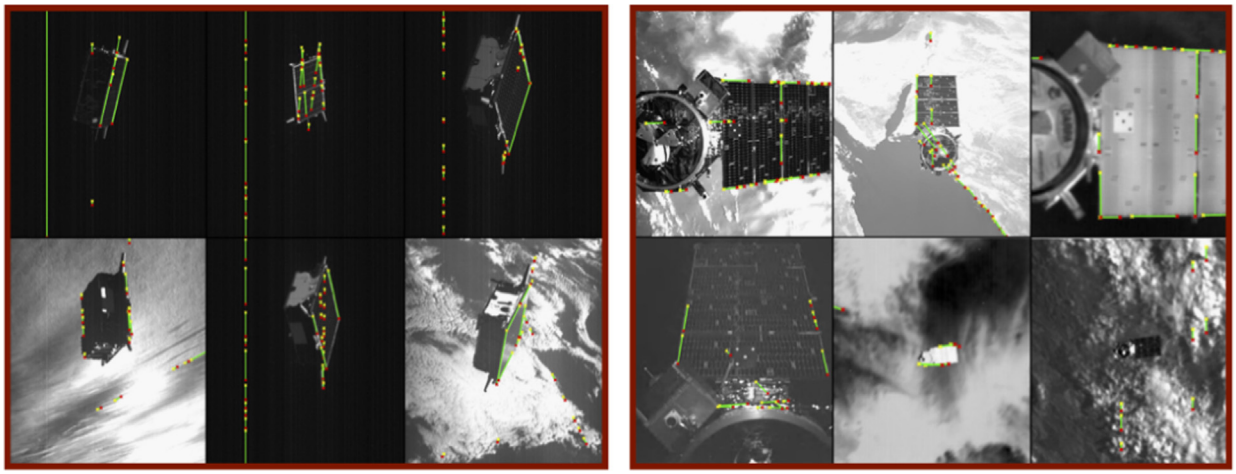

Fig. 8. Typical output of an object detection subsystem overlaid on actual space imagery from the PRISMA[23] mission (left) and Orbital Express[22] mission (right).

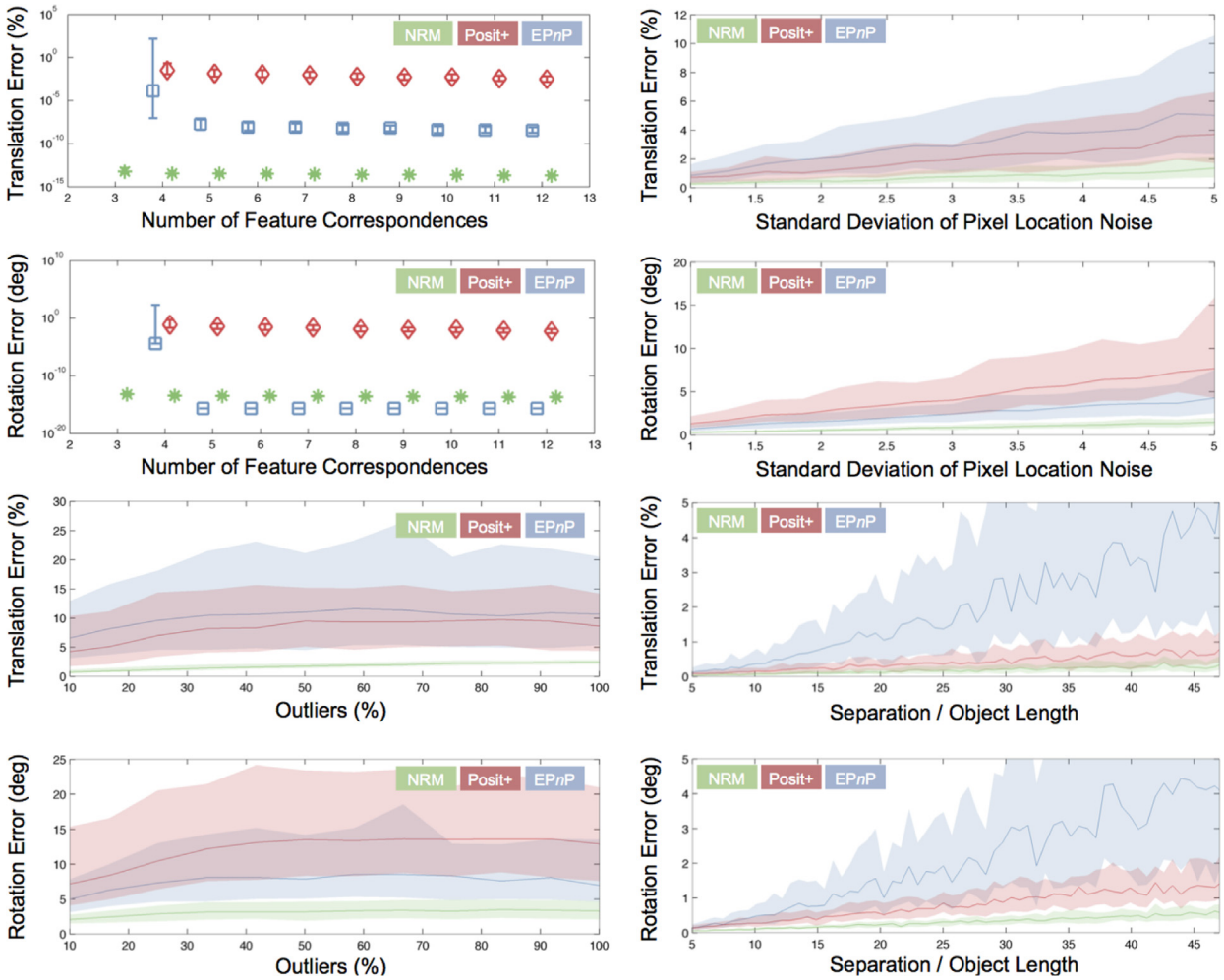

Fig. 9. Results for translation error and rotation error from simulations of all test cases: number of feature correspondences (top left), pixel location noise (top right), outliers (bottom left), and distance along optical axis (bottom right).

to feature correspondence is prone to errors even when object detection is perfect, for example, when the geometry of the space resident contains indistinguishable or symmetric surfaces. A robust P*n*P solver should negate the effects of incorrect feature correspondences (outliers) present in its input and produce an accurate pose estimate. Hence, as our third test case, we perform 10^4 simulations with varying number of outliers. Outliers are 2D image points with pixel location error of 10 px. This is in addition to the pixel location noise characterized by Gaussian distribution of a mean of zero and a standard deviation of 2 px. This noise is added to the 2D image points generated from the true perspective projection of twelve 3D model points. As in Test Case 2, the true pose is kept constant for all simulations and iterations for NRM and PosIt are limited.

Test Case 4: *distance along optical axis*. With fixed camera characteristics, increasing the separation decreases the spread in the distribution of the 2D features on the image plane making it difficult for the object detection subsystem to resolve features. Hence, as our fourth test case, we perform 10^4 simulations with varying relative separation of the space resident object and the camera in the direction of the optical axis of the camera. Pixel location noise is added to the 2D image points generated from the true perspective projection of six 3D model points. Noise is characterized by a Gaussian distribution with a mean of zero and standard deviation of 2 px. The relative attitude is kept constant and feature correspondences are perfect for all simulations in this test case. As in Test Case 2, the number of iterations for NRM and PosIt is limited.

4. Results & Discussion

For each test case, mean and variance of performance criteria for all simulations are represented as markers/lines and bars/shaded regions, respectively (see Figs. 9 and 10). Variance is reported as the interquartile range, i.e., the difference between the upper and lower quartile values of performance criteria.

4.1. Test Case 1: number of feature correspondences

NRM is the most computationally expensive solver due to its use of least squares to invert the Jacobian matrix at each iteration. Typically, least squares in NRM is an operation of $O(c^2n)$ complexity where c is a constant equal to 6, the number of unknowns for the P*n*P problem. This is highly expensive in comparison to an $O(c^3n)$ overall complexity of EP*n*P and $O(24n)$ overall complexity for PosIt. Note that for $3 \leq n \leq 6$, NRM requires fewer and fewer iterations to converge as n is increased. This leads to an overall decrease in computational runtime even though complexity per iteration increases. For this test case, PosIt has the highest errors in comparison to other solvers but it improves in accuracy for increasing values n. Even though EP*n*P provides a pose estimate for $n=4$, it has the largest variance in the translation error and the rotation error at this value of n.

4.2. Test Case 2: pixel location noise

With an increase in pixel location noise in the input 2D image points, all solvers exhibit a decrease in accuracy. NRM has the best performance due to its use of least squares, which is intentionally well suited to handle Gaussian noise. PosIt and EP*n*P have approximately the same variance in rotation error as well as translation error and exhibit a linear increase in errors with an increase in noise. But unlike EP*n*P which provides a closed-form solution, errors of PosIt can be reduced at the expense of computational runtime if more iterations are allowed. As with the first test case, PosIt's runtime is the lowest and NRM's runtime is the highest.

4.3. Test Case 3: outliers

Recall that an outlier image point was earlier defined as an image point with a random pixel location error of greater than $5\sigma_P$, where σ_P is fixed at 2 px to characterize the measurement noise. All solvers exhibit a linear increase in errors when outliers made up less than 40% of the input 2D image points. For higher percentages, errors hold approximately constant values as the performance degrades to levels where additional outliers have no significant effect on accuracy. However, the slope of the increase in errors is proportional to the pixel location error used to generate the outliers. Comparatively, NRM has the lowest errors while PosIt and EP*n*P have similar levels of error. Absolutely, all solvers have sub-optimal performance in the context of initial pose estimation where feature correspondences are unknown and input of 2D image points can contain high pixel location errors.

4.4. Test Case 4: distance along optical axis

As the distance along the optical axis is increased, EP*n*P is the worst affected. One of the intermediate steps in EP*n*P is the calculation of a basis of the null space of a matrix containing all 2D image points. With increasing distance along the optical axis and apparent shrinkage of the image, this matrix tends to become sparse and null space estimation becomes increasingly challenging. However, as the distance is increased, SOP tends to be a better approximation of true perspective projection. Hence, PosIt, which is based on the SOP approximation has lower rotation and translation error than EP*n*P. However, NRM has the lowest translation and rotation error as even at large separations, least squares is well suited to handle noisy input of 2D image points.

4.5. Decision matrix

To conclude the comparative assessment, we construct a qualitative decision matrix (see Fig. 11) with P*n*P solvers indicated on the rows and the test cases indicated on the columns. Each cell is color coded to represent the solvers' weighted sum of relative performance as measured by the four performance criteria. Recall that these performance criteria were earlier defined as the computational runtime, the image plane error, the translation error, and the

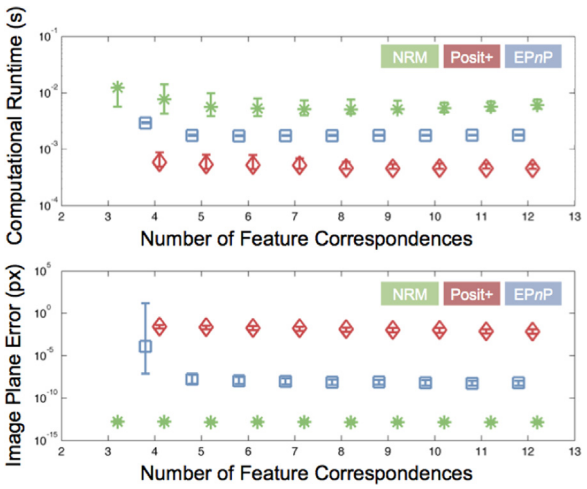

Fig. 10. Results for computational runtime and image plane error from simulations of test case 1.

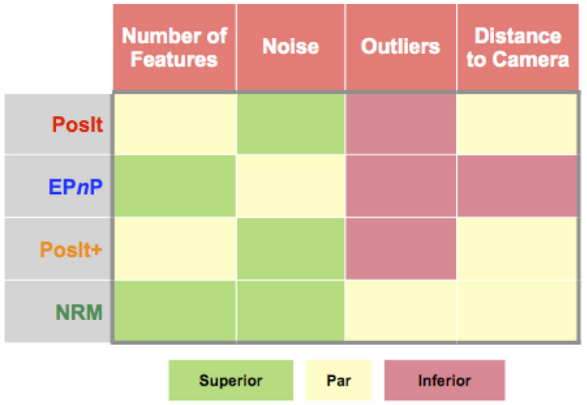

Fig. 11. Comparative assessment results for simulations from all test cases as a qualitative decision matrix.

rotation error. We give equal weights to the difference of each performance criteria from the mean value of the respective performance criteria, for all test cases and for all P*n*P solvers. We then use terms "superior", "par", and "inferior" to represent this difference of performance criteria.

We see that the use of PosIt+ makes the PosIt solver applicable to both coplanar and non-coplanar points without having a significant impact on computational runtime. EP*n*P has the best performance when pixel locations are free from noise or outliers. NRM has the lowest errors across all test cases. However, it requires an input guess of the pose and it also has an order of magnitude higher computational time across all test cases. In the presence of feature outliers, all solvers have sub-optimal levels of rotation and translation error.

5. Conclusions & way forward

Preliminary results indicate that the runtime of each call of a P*n*P solver is on the order of 10 ms when embedded in a spaceborne microprocessor (clock-rate of 30 MHz). In a typical scenario, we can expect about 10^3 calls of a P*n*P solver which makes their current performance unsatisfactory for a spaceborne application of real-time pose estimation. Accuracy of the P*n*P solvers is acceptable only when feature correspondence is perfect and is sub-optimal in the presence of feature outliers. Hence, an iterative statistical approach will be necessary to achieve pose convergence in real-world applications where multiple feature correspondence hypotheses need to be validated. The strength of each P*n*P solver lies in different regimes and calls for a strategy to exploit the identified synergies. Future work will implement a P*n*P mode switcher based on the obtained decision matrix to yield a fast and robust solution to the perspective equations under diverse operational scenarios. For example, since EP*n*P has the lowest runtime, it can be used during the first few iterations of pose estimation when large number of correspondence hypotheses need to be validated. However, in later iterations when the search space for correct feature correspondence has been reduced to a few ambiguous hypotheses, NRM can be used due to its better accuracy in the presence of outliers. If the pose estimate is still ambiguous, the architecture could acquire and process subsequent images and validate each ambiguous pose estimate by exploiting principles of relative orbit dynamics and kinematics. The interplay between the initial pose estimation and the object detection subsystem needs to be explored further. The gradual inclusion of more features (in number and in type) and the data editing process need to be studied further. Since we now have a measure of process errors in a variety of test cases, it has to be understood how such errors could be properly incorporated in a filtering scheme. This comparative assessment not only provides a framework to review initial pose estimators but its conclusions will also serve as a cornerstone for the design of pose estimators for future missions involving rendezvous and proximity operations.

References

[1] L.T. Castellani, et al., PROBA-3 mission, Int. J. Space Sci. Eng. 1 (4) (2013) 349–366.
[2] A.F.R. Laboratory, Fact sheet: automated navigation and guidance experiment for local space (ANGELS) ⟨http://www.kirtland.af.mil/shared/media/document/AFD-131204-039.pdf⟩, July 2014 (accessed 30.09.2014).
[3] S.D'Amico, et al., PRISMA, in: M.D'Errico (Ed.), Distributed Space Missions for Earth System Monitoring, Springer New York, New York, NY, ISBN: 978-1-4614-4540-1, 978-1-4614-4541-8, 2013, pp. 599–637.
[4] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2) (2004) 91–110.
[5] D. Grest, T. Petersen, V. Kr'uger, A comparison of iterative 2D-3D pose estimation methods for real-time applications, in: A.-B. Salberg, J. Hardeberg, R. Jenssen (Eds), Image Analysis, vol. 5575, Lecture Notes in Computer Science, Springer Berlin Heidelberg, http://dx.doi.org/10.1007/978-3-642-02230-2_72, 2009, pp. 706–715, ISBN: 978-3-642-02229-6.
[6] D.W. Eggert, A. Lorusso, R.B. Fisher, Estimating 3-D rigid body transformations: a comparison of four major algorithms, Mach. Vis. Appl. 9 (5–6) (1997) 272–290.
[7] A. Yilmaz, O. Javed, M. Shah, Object tracking: a survey, ACM Comput. Surv. 38 (December (4)), 13es, http://dx.doi.org/10.1145/1177352.1177355, 2006, ISSN: 03600300.

[8] D. Nistèr, Preemptive RANSAC for live structure and motion estimation, Mach. Vis. Appl. 16 (December (5)), http://dx.doi.org/10.1007/s00138-005-0006-y, 2005, pp. 321–329, ISSN: 0932-8092, 1432-1769.
[9] J. Feldman, Perceptual grouping by selection of a logically minimal model, Int. J. Comput. Vis. 55 (1) (2003) 5–25.
[10] S. D'Amico, et al., Noncooperative rendezvous using angles-only optical navigation: system design and flight results, J. Guid. Control Dyn. 36 (November (6)), http://dx.doi.org/10.2514/1.59236, 2013, pp. 1576–1595, ISSN: 0731-5090, 1533-3884.
[11] M.A. Fischler, R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Commun. ACM 24 (6) (1981) 381–395.
[12] D.F. Dementhon, L.S. Davis, Model-based object pose in 25 lines of code, Int. J. Comput. Vis. 15 (1–2) (1995) 123–141.
[13] Vincent Lepetit, Pascal Fua, Monocular model-based 3D tracking of rigid objects: A Survey, Found. Trends® Comp. Graph. Visi. 1 (1) (2005) 1–89, http://dx.doi.org/10.1561/0600000001.
[14] Y. Zheng, et al., Revisiting the PnP problem: a fast, general and optimal solution. in: IEEE International Conference on Computer Vision, http://dx.doi.org/10.1109/ICCV.2013.291, December 2013, pp. 2344–2351, ISBN: 978-1-4799-2840-8.
[15] C.-P. Lu, G.D. Hager, E. Mjolsness, Fast and globally convergent pose estimation from video images, IEEE Trans. Pattern Anal. Mach. Intell. 22 (6) (2000) 610–622.
[16] S. D'Amico, M. Benn, J.L. Jørgensen, Pose estimation of an uncooperative spacecraft from actual space imagery, Int. J. Space Sci. Eng. 2 (2) (2014) 171–189.
[17] A. Cropp, P. Palmer, C.I. Underwood, Pose estimation and relative orbit determination of a nearby target microsatellite using passive imagery (Ph.D. thesis), University of Surrey, 2001.
[18] G. Campa, et al., A comparison of pose estimation algorithms for machine vision based aerial refueling for UAVs, in: 14th Mediterranean Conference on Control and Automation, 2006 (MED'06), IEEE, Ancona, Italy, 2006, pp. 1–6.
[19] Oberkampf, Denis, Dementhon, F. Daniel, Davis, S. Larry, Iterative pose estimation using coplanar feature points, Comput. Vis. Image Underst. 63 (3) (1996) 495–511.
[20] V. Lepetit, F. Moreno-Noguer, P. Fua, EPnP: an accurate O(n) solution to the PnP problem, Int. J. Comput. Vis. 81 (February (2)), http://dx.doi.org/10.1007/s11263-008-0152-6, 2009, pp. 155–166, ISSN: 0920-5691, 1573-1405.
[21] Eyes of the Solar System, NASA/JPL-Caltech, Rosetta 3D Model, July 2012.
[22] D.A. Whelan, et al., Darpa orbital express program: effecting a revolution in space-based systems, in: International Symposium on Optical Science and Technology. International Society for Optics and Photonics, 2000, pp. 48–56.
[23] T. Karlsson, et al., PRISMA Mission control: transferring satellite control between organisations, in: SpaceOps 2012 (2012).
[24] R.O. Duda, P.E. Hart, Use of the Hough transformation to detect lines and curves in pictures, Commun. ACM 15 (1) (1972) 11–15.
[25] P. Tissainayagam, D. Suter, Assessing the performance of corner detectors for point feature tracking applications, Image Vis. Comput. 22 (August (8)), http://dx.doi.org/10.1016/j.imavis.2004.02.001, 2004, pp. 663–679, ISSN: 02628856.
[26] X. Liu, M. Tanaka, M. Okutomi, Single-image noise level estimation for blind denoising, IEEE Trans. Image Process. 22 (December (12)) (2013) 5226–5237, http://dx.doi.org/10.1109/TIP.2013.228340. ISSN: 1057-7149, 1941-0042.