Pose Estimation of Target Satellite for Proximity Operations

A. Cropp, Dr P. Palmer, Dr C.I. Underwood

Surrey Space Centre, University of Surrey, Guildford, Surrey GU2-7XH, UK

Tel: +44 (0) 1483 – 300800 x2692

Fax: +44 (0) 1483 – 259503

Email: a.cropp@eim.surrey.ac.uk

Abstract. Close-proximity operations are increasingly a topic of interest, where satellites manoeuvre within a very small distance of other spacecraft. A high degree of accuracy is required in estimating the relative position and orientation of the other spacecraft, in order to conduct such manoeuvres safely. Traditionally, active systems such as radar or more recently Differential GPS have been used for relative position estimation, yet these give little information on the orientation of the other satellite. Passive imaging can provide a large amount of information on the location and orientation of the Target, with high spatial resolution. Imaging requires only low-powered cameras, which can be made available on a wider range of satellites, and does not require any functionality from the other spacecraft.

A robust autonomous close-range relative orientation and location (pose) estimation system is proposed, based on computer vision. Using a single image, and utilising knowledge of the Target spacecraft, an estimation of the Target's six relative rotation and translation parameters are found from a distance in the order of 10 metres. Such position and rotation estimates over time will allow relative orbit parameter estimation, and enable close-proximity operations such as docking and remote inspection.

1. Introduction

Spacecraft Rendezvous and Docking (RVD) has been performed routinely with humans in the loop, but recently interest has been shown in performing this complicated task autonomously. Human control from the ground requires large-bandwidth communications in order to have continuous data with low delays. Ground visibility constraints with Low-Earth-Orbit (LEO) satellites means that direct communication can only come at certain times, unless expensive relay communications are involved (if even available). Rendezvous and docking requires manoeuvres which may position satellites so that communications antennas are no longer optimally aligned with the ground-station receivers, or possibly the target satellite may occlude the ground station all together. In effect, there are RVD situations where human control from the ground is simply not available, and autonomous control is required.

There are current automated docking systems available – the Progress re-supply vehicles have been docking with Mir for a number of years automatically. However, in these situations, the target to be docked with is in a known position, with a known orientation. This paper deals with the case of a target in an unknown orientation and location. It is also assumed that the target will not attempt to aid in the docking attempt, for example by communicating with the docking craft or by carrying out any manoeuvres. Lack of communication will

restrict the use of systems such as Differential GPS. In effect, the target is passively non co-operating. This may be because the target is damaged, or is still under construction, or is busy.

In order to dock with the target, the position and velocities need to be measured relative to the inspection craft, as well as the orientation and angular velocity. To carry out these measurements, it is assumed that the inspection vehicle uses passive imaging as the main source of information. The hardware required is simple and should be readily available even for small satellites with a very low mass and power budget (the SNAP-1 nanosatellite from Surrey Space Centre has 4 cameras on board, and weighs less than 7 kg). Images contain a large amount of information, with a high spatial resolution. In order to reduce this problem to manageable levels, it is assumed that the inspection craft has prior knowledge of the target.

This paper presents a robust autonomous close-range pose estimation system, based on computer vision. Using a single monocular image, and utilising knowledge of the target spacecraft, estimations of the target's orientation and location with respect to the camera are found.

There are several key differences between satellite-based machine vision systems such as the one proposed here, and their equivalent terrestrial systems. There is no complex background other than the Earth in space, which should be easily recognizable and therefore easy to remove. Only one object of interest is in the field of view, ie. the target, and many satellites are based on simple polyhedral shapes, which can be easily modelled. Lighting conditions in space usually generate high-contrast images, producing highly defined shadows.

Unlike previous satellite-based computer vision techniques [1,2], this work does not require specialised markers to be present on the target, nor does it need to search for specific (and possibly occluded) structures that must be attached to the target in advance. Instead, a more general system is proposed, that makes use of the target's own structure in estimating pose.

Our method is based on the fact that a detailed model of the target satellite is available. This is reduced to a line model of the principal observable features (see Figure 1 – right), with the origin at the Centre of Gravity (CoG) of the target, and the same scale as the target. In pose estimation, the aim is to match the model lines with lines detected in the image of the target. Such matching will enable the calculation of target orientation as well as the location of the target CoG with respect to the camera, assuming that the camera calibration parameters are known.

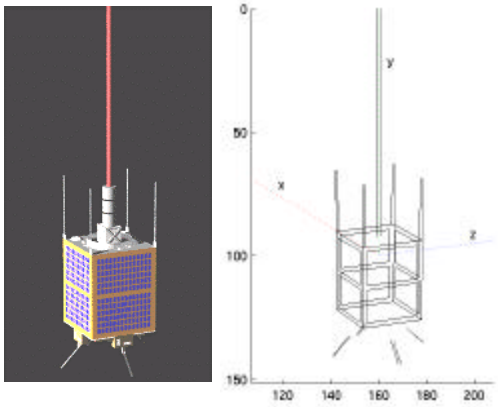

Figure 1: Example image of target microsatellite (left), with Line Model of target under same perspective projection (right)

This paper deals with the initial stage of testing, where the target is a Surrey Space Centre generic UoSAT microsatellite, which is a simple box-shaped satellite with extruding antennas. We present results from target images generated synthetically using the freeware script-driven rendering software POV-Ray (www.povray.com). The system will be tested using in-orbit data from the SNAP-1 Nanosatellite Inspection Mission, which was recently launched on 28th June 2000.

2. Coordinate Frames

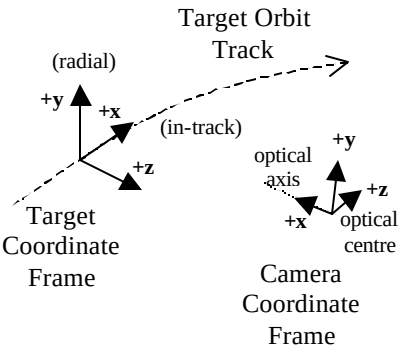

Figure 2: Target and Camera coordinate frames

Figure 2 shows the two coordinate frames in use. The target frame origin is the target's CoG, while the target direction vectors are fixed to the in-track orbital direction (**+x**), the radial direction (**+y**), and the out-of-plane direction (**+z**), rather than the actual orientation of the target. Hence the target may be rotated with respect to the target coordinate frame, but the translation is always 0. The second co-ordinate frame is the camera coordinate frame, where the camera in question is the passive imager attached to the inspection craft. It is assumed that the camera calibration parameters are known, and hence the machine vision system calculates rotations and translations with respect to the known camera. The transformation from camera to docking satellite coordinates is fixed and assumed known. The **+x** axis in the camera coordinates is the optical axis of the camera, while the **+y** and **+z** axis represent the 'camera up' and 'camera right' directions respectively.

The rigid transformation between the two coordinate frames is in the form of a rotation and translation:

$$\mathbf{p}_c = R\,\mathbf{p}_t + \mathbf{t}_c \qquad \textbf{(2.1)}$$

where $\mathbf{p}_c$ is a 3-dimensional point in the camera coordinate frame, R is a 3x3 orthogonal rotation matrix, $\mathbf{p}_t$ is the same point described in target coordinates, and $\mathbf{t}_c$ is a translation in camera coordinates. It is the values of the three rotation angles that make up R and the three translation values of $\mathbf{t}_c$ that are required in pose estimation.

Figure 3 shows the projection from camera coordinates to image coordinates, where f is the focal length of the camera (the distance from the optical centre to the image centre). The vectors **+u** and **+v** are the image plane coordinate vectors. Point **q** is a point in 3-dimensional space, while $\mathbf{q}_i$ is the 2-dimensional point projected onto the image plane.

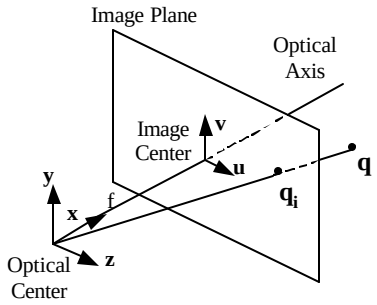

Figure 3: Pinhole Camera Model

Given the point $\mathbf{q}=[x,y,z]$, the image point $\mathbf{q}_i$ can be found as follows:

$$\mathbf{q}_i = [u, v] = \frac{f}{x}[z, y] \qquad \textbf{(2.2)}$$

The object of this pose estimation of the target is to take the 2-dimensional image coordinates of a target image, and convert this to an estimate of the rotation and translation from camera to target coordinates.

Due to the scaling effects of x in Eqn. **(2.2)**, the 3D position of **q** cannot be recovered from $\mathbf{q}_i$ alone. For example, two similar objects, one twice as big yet twice as far away, will appear the same from the camera's point of view. If, however, the object's size is known beforehand, some effort can be made in removing the unknown scalar ambiguity.

3. Pose Estimation System Description

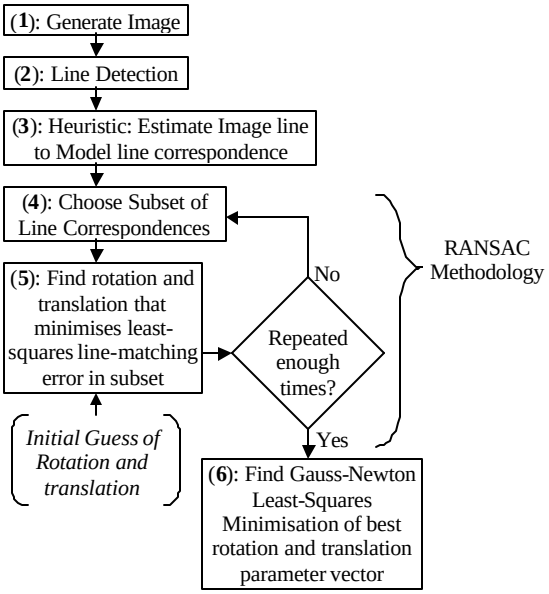

Figure 4: System Diagram

Figure 4 shows the pose estimation system diagram. Block (**1**) involves synthetic image generation. A UoSAT microsatellite was chosen as the target, and was rendered using POV-Ray. The rendered image contains many of the same complexities as real images, from a machine vision point of view, such as specularity, shadowing, and self-occlusion. Rendered images are too perfect in appearance, and therefore a 2D convolution process was performed using a flat 3x3 mask. This convolution produced a blurring effect on the images, distorting small-scale (high-frequency) objects.

Because the rendering programme is script-driven, an exterior programme such as Matlab can edit the position and orientation values of the target. POV-Ray can then be called to generate the necessary image, which is then imported back into Matlab. Hence, a simulation environment is created that can generate a visual response, for example when modelling the orbital drift between two near-by satellites.

Block (**2**) detects the major straight lines using a Hough-transform line detector that detects to sub-pixel accuracy [3]. Figure 5 (left) shows an example of a target image with unknown rotation and translation. The shadows have merged with the background, so that one face of the target is not visible (the image is displayed as a negative). Figure 5 (right) shows the lines detected in the target image.

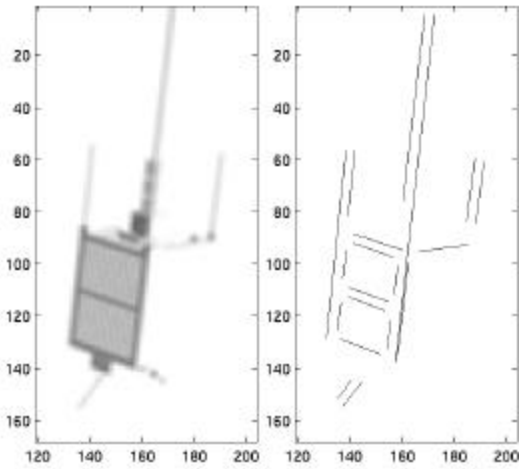

Figure 5: Negative target Image (left), recovered lines (right)

Block (**3**) uses heuristics to try to categorise the detected lines, in order to generate a list of correspondences between the lines detected in the image and lines in the model. These heuristics were generated from scratch, and search for empirically observed patterns of lines. The development of these heuristics was time-consuming, and the heuristics are very specific to the target under observation. However, without heuristics, the number of possible combinations of matches would be too great to process in real time. Errors in line detection can cause line merges or broken lines, which would further increase the number of combinations in a brute-force matching method. The use of heuristics allows a form of input for human observational and generalised problem-solving abilities.

Some assumptions were made in the creation of these heuristics. In order to

reduce the development time, it was assumed that the target vertical would be within 30° of the camera vertical. This assumption will be relaxed in the future, which will require an expansion of current heuristics, rather than a new set, and should pose no greater problem that the initial creation of the current heuristics. These current heuristics are intended mainly to demonstrate the validity of this approach.

It is possible to calculate the rotation and translation of the target given a set of correct line-to-line correspondences [4,5], assuming the target model is available (see Section 4 below). In order to remove inaccurate correspondences from the list created in block (**3**), a small subset is chosen at random in block (**4**), which is then used to calculate the rotation and translation in block (**5**). An initial guess of the pose is used block (**5**), but the results are insensitive to errors in the initial guess. The block (**4**-**5**) sequence is repeated several times, each time choosing a new correspondence subset at random. This process is similar to the Random Sample Consensus (RANSAC) method [6]. The minimum number of line correspondences is taken to be 4, and the maximum loop count is set to 20.

In order to judge the quality of the match, the model is first projected onto the image using the estimated rotation and translation. For each possible combination of image and model line, a value is assigned depending of the difference in position, angle, length, and end-point location. Each image line is assigned the model line with the highest value (best fit), and the average value for all the image lines is taken as the quality of fit for that match. The best matches are then used in block (**6**), where the Gauss-Newton minimisation of the pose is performed [7,8] on the parameter vector. The output of block (**6**) gives another set of estimates for rotation and translation that are tested for quality of match, and the match with the best score overall gives the final estimate for that image.

4. Estimating R and t from Line-to-Line Correspondence

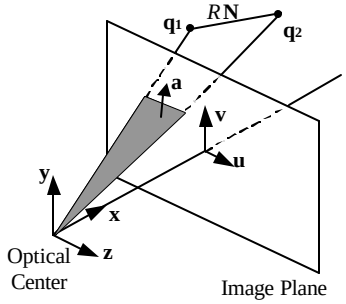

Figure 6: Interpretation Plane

Figure 6 demonstrates the Interpretation plane, formed from the 3D points $\mathbf{q_1}$, $\mathbf{q_2}$, and the Optical centre, with a normal vector **a**. The vector difference between the two points $\mathbf{q_1}$ and $\mathbf{q_2}$ form a line in 3D space, which is projected onto the image plane. The projected 2D line is also parallel with the Interpretation plane. The points $\mathbf{q_1}$ and $\mathbf{q_2}$ are the result of a rotation and translation transformation:

$$\mathbf{q_1} = R\,\mathbf{p_1} + \mathbf{t} \qquad \textbf{(4.1)}$$

$$\mathbf{q_2} = R\,\mathbf{p_1} + \mathbf{t} \qquad \textbf{(4.2)}$$

where $\mathbf{p_1}$ and $\mathbf{p_2}$ are the end-points of the same 3D line before transformation. The vector **a** is perpendicular to both vectors $\mathbf{q_1}$ and $\mathbf{q_2}$, hence (without noise):

$$\mathbf{a}^T(R\mathbf{p}+\mathbf{t}) = 0 \qquad \textbf{(4.3)}$$

where **p** is any point on the line including $\mathbf{p_1}$ and $\mathbf{p_2}$. In Figure 6, the vector **N** represents the unity direction vector of the line formed by points $\mathbf{p_1}$ and $\mathbf{p_2}$:

$$\mathbf{N} = \frac{\mathbf{p}_2 - \mathbf{p}_1}{\left\|\mathbf{p}_2 - \mathbf{p}_1\right\|} \qquad \textbf{(4.4)}$$

N represents the direction vector of the line before transformation, and *R* represents the rotation transformation. Therefore, the transformed vector *R***N** is also perpendicular to the Interpretation plane, and hence:

$$\mathbf{a}^T(R\mathbf{N}) = 0 \qquad \textbf{(4.5)}$$

Without noise, Eqn. **(4.5)** is true. **N** represents the (known) model line before transformation, and the rotation matrix *R* represents the (unknown) rotation from model to camera coordinates. The vector **a** can be calculated, as the image points and the camera calibration parameters are known.

With noise, Eqns. **(4.3)** and **(4.5)** are no longer satisfied. In order to find *R*, the sum-of-squares error function F_1 is minimised:

$$F_1 = \sum_{i=1}^{n} \left(\mathbf{a}_i^T R \mathbf{N}_i\right)^2 \qquad \textbf{(4.6)}$$

where *n* is the number of line correspondences. *R* is a non-linear function of three rotation angles, and so a linearisation procedure must be applied to *R* 4·5 . Once *R* is estimated, the translation vector **t** is found as the least-squares minimisation of F_2:

$$F_2 = \sum_{i=1}^{n} \sum_{j=1}^{2} \left(\mathbf{a}_i^T \left(R\mathbf{p}_i^j + \mathbf{t}\right)\right)^2 \qquad \textbf{(4.7)}$$

Where *j*=1..2 are the two end-points of line *i*. Hence, given a set of correct line correspondences, the rotation and translation can be found.

5. Results

Figure 7 demonstrates the estimated pose given the correct correspondences. The left of Figure 7 shows the detected image lines, while the right is the projected line model, with the image lines shown as dotted lines. The error in rotation in this case was less than 5°, and translation error was less than 5% of the true distance to the target. The majority of this translation error was in the direction of the optical (x) axis. Errors in the y-axis and z-axis were less than 0.2% and 0.1% respectively. In this case, the range to target was 10 metres, with a field of view of 24°x18°, a resolution of 320x200 pixels in the image plane, and a focal length set to 1.

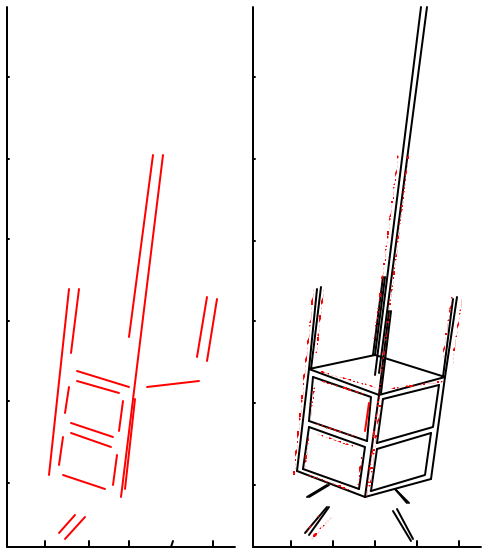

Figure 7: Pose Estimation for Correct Correspondence

Figure 8 demonstrates results of 58 random rotation and translation trials, for 20 iterations each image. Some restrictions were made to the ranges of rotation and translation. For translation, the range was kept low enough to be able to detect lines, yet far enough away that the main body of the target (i.e. not including the antennas and boom) is in view. For rotation, the target vertical was kept within 30° of the camera vertical to comply with the heuristic assumptions, but otherwise the rotations are unfixed. Of the 58 random images, the heuristics failed on 2 images (which meant no pose estimation possible), and a further image produced wildly inaccurate results. These failures were due to too few lines being detected to too large target distance. The remainder are shown in Figure 8. The median rotation error was less than 5°, and the median translation error was less than 2% of the distance to the target. Most of the translation error is along the optical axis. Each image took approximately 1 minute to process on an Intel P75, running un-optimised Matlab code.

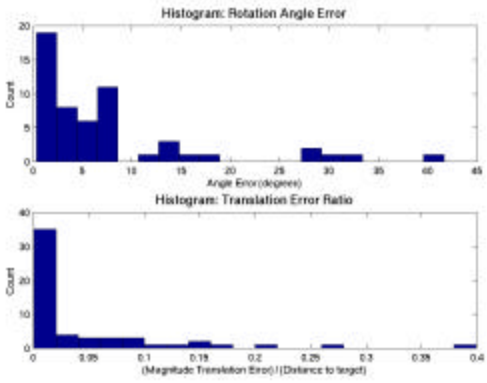

Figure 8: Results

Both histograms in Figure 8 show outliers, the most common cause being a non-unique solution to the pose estimation problem, where the correct correspondence would still give an incorrect estimate. These estimates often have high match quality values despite the error, overriding other weaker but more accurate results. This can be overcome by increasing the number of iterations, which generated more suitable results in most of the outlier cases. However, outliers will be difficult to spot, and increasing the number of iterations throughout will increase required computation time. Insufficient numbers of image lines is another reason for poor results, often caused when the target is too far away for a given set of camera parameters. With very few lines, the heuristics usually fail to find the required patterns. There are also minor errors in the model due to simplifications made, which contribute to small errors in estimation.

6. Conclusions

These results demonstrate reasonable accuracy using a single frame from a monocular camera in near-real time. Only one frame is required for each estimate, so with several consecutive frames, results can be filtered to improve accuracy. This would give an inspection satellite the ability to autonomously determine its own position and velocity relative to the spacecraft under inspection, as well as estimating the orientation of the target. The requirements for this system are not excessive, both in terms of required camera hardware and computational load, and could therefore be considered for many small-satellite inspection-class missions.

7. References

[1]: Ho, C.J. and McClamroch, N.H., "Autonomous Spacecraft Docking using a Computer Vision System", *Proc. 31st Conf. on Decision and Control*, Tucson, Arizona, pp. 645-650, December 1992

[2]: Philip, N.K., Kumar, E.K., and Ananthasayanam, M.R., "Sliding Observer for a Robust Relative Position and Attitude Estimation During the Final Phase of an Autonomous Docking Mission", *Proc. 49th Int'l Astronautical Congress*, Melbourne, Australia, October 1998

[3]: Palmer, P.L., Kittler, J., and Petrou, M., "An Optimizing Line Finder Using a Hough Transform Algorithm", *Computer Vision and Image Understanding*, vol. 67, no. 1, pp. 1-23, July 1997

[4]: Liu, Y., Huang, T.S., Faugeras, O.D., "Determination of Camera Location from 2-D to 3-D Line and Point Correspondences", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 12, no. 1, pp. 28-37, January 1990

[5]: Lee, C. and Haralick, R.M., "Statisitcal estimation for exterior orientation from line-to-line correspondences", *Image and Vision Computing*, vol. 14, pp. 379-388, 1996

[6]: 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, pp. 381-395, 1981

[7]: McLauchlan, P.F., "A Batch/Recursive Algorithm for (3D) Scene Reconstruction", *To appear in Proc. (IEEE) Conf. On Computer Vision and Pattern Recognition*, 2000

[8]: Bjorck, A., "Numerical Methods for Least Squares Problems", SIAM, Philadelphia, 1996