VISUAL POSE ESTIMATION SYSTEM FOR AUTONOMOUS RENDEZVOUS OF SPACECRAFT

Mark A. Post[1], Xiu T. Yan[2], Junquan Li[3], and Craig Clark[4]

[1]*Lecturer, Space Mechatronic Systems Technology Laboratory. Department of Design, Manufacture and Engineering Management, University of Strathclyde, Glasgow, United Kingdom*
[2]*Professor, Space Mechatronic Systems Technology Laboratory. Department of Design, Manufacture and Engineering Management, University of Strathclyde, Glasgow, United Kingdom*
[3]*Marie Curie Experienced Researcher, Clyde Space Ltd. Glasgow, United Kingdom*
[4]*CEO, Clyde Space Ltd. Glasgow, United Kingdom*

ABSTRACT

In this work, we consider a tracker spacecraft equipped with a short-range vision system that must visually identify a target and determining its relative angular velocity and relative linear velocity using only visual information from an onboard camera. By means of visual feature detection and tracking across rapid, successive frames, features detected in two-dimensional images are matched and triangulated to provide three-dimensional feature maps using structure-from-motion techniques. Triangulated points are organized by means of orientation histogram descriptors and used to identify and track targets over time. The state variables with respect to the camera system are extracted as a relative rotation quaternion and relative translation vector that are tracked by an embedded unscented Kalman filter. Inertial measurements over periods of time can then be used to determine the relative movement of tracker and target spacecraft. This method is tested using laboratory images of spacecraft movement with a simulated spacecraft movement model.

Key words: Satellite; Pose Estimation; Vision.

1. INTRODUCTION

Visual Pose Estimation technology has attracted a lot of interest for spacecraft navigation as an enabling technology for rendezvous and docking manoeuvres. Guidance and Control of a spacecraft has been studied extensively, but in order for such systems to work effectively between spacecraft close to each other, the relative position, attitude and and velocity between each spacecraft must be robustly estimated. The desired result is that two satellites will be able to reliably and autonomously rendezvous with each other, but visual position estimation for satellites in orbit is far from a solved problem.

Traditionally, RF radar trades off precision for wide range of operation, and is not as suitable for uncooperative or small targets. The TriDAR system used a LIDAR and Iterative Closest Point system outside the ISS without approach or autonomy [RLB12]. Recent automated rendezvous and docking systems make use of optical, laser ranging, and LIDAR systems [HCDS14] [PHAR12] and visually-aided systems have been tested in proximity operations with NASA's Space Shuttle, JAXA's ETS-VII satellite [Oda00], and other satellites such as the DART mission [RT04].

However, the complexity, size, and power requirements of current LIDAR systems are still out of reach for small satellites and nanosatellites, and there is great potential in the use of multiple-view imaging and feature mapping since only one camera may be necessary. Many pose estimation techniques have been proposed for this, and typically focus on shape tracking and recognition, feature detection and triangulation [Sha14], or a combination of shape and features [TBB11]. The SPHERES experiment uses SURF feature matching with stereo vision for navigation inside the ISS [TSSO+14].

In this work, we propose a different approach to the monocular visual estimation problem: recognition and tracking of features for ego-motion from a sequence of images, which can then be inserted into a point cloud, which in turn provides a way to recognize the position of the target. This method is derived from structure-from-motion computer vision methods used in robotics and in photo-tourism reconstructions from large image sets, and requires that only rigid transformations are present between images. To speed the development process and minimize coding errors and complexity, we make use of the open-source OpenCV (Open Computer Vision) and PCL (Point Cloud Library) for most of the computer vision programming.

2. APPROACH AND TRACKING

To allow a tracker spacecraft to to identify and estimate the movement of a target spacecraft, we approach this problem as illustrated in Fig. 1 First, we build up a feature set of points located in three dimensions by triangulation of keypoints on successive images of the target in the "Approach" phase. We then locate the camera relative to the matched points by Perspective-n-Point (PnP) solution during the "Track" phase. By projecting the keypoints into three dimensions, we build up a point cloud of the target over many more images in the "Observe" phase, which can then be matched in shape to a point cloud model, and the pose of the model accurately obtained by three-dimensional keypoint correspondences in the "Identify" phase.

Feature-based vision methods reduce complete images to a set of distinct, reproduceable "features" that are represented by small numerical sequences. We apply ORB (Oriented FAST and Rotated BRIEF) point descriptors for 2-D feature matching with high rotation invariance [RRKB11]. We then use structure-from-motion methods to triangulate these points in space.

2.1. System Overview

A flowchart of the process we propose is shown in Fig. 2, with details on each step provided in the following sections. A sequence of images can be captured or cached, features extracted using two-dimensional point descriptors that are stored in memory and matched in pairs to obtain a list of images with features, and also a list of features tracked across images. This list of feature correspondences is used to track the movement of keypoints across several poses, and.if the triangulation is not good enough, a more different pose containing those features is selected. Using a pose solution, the points and camera are projected into global coordinates. The resulting scene point cloud can then be compared with a model cloud to identify the target by choosing a set of keypoints and extracting histogram descriptors for each with respect to point normals. By matching descriptors between the scene and model, the model and its pose can be found within the scene. An OptiTrak Trio optical tracking system is currently used as an external high-speed reference for pose estimation. The pose estimates are then filtered over time using an Unscented Kalman Filter to reduce noise. Sensor fusion of the triangulation and correspondence tracker-target measurements is planned, but has not been implemented yet.

2.2. Keypoint Detection and Matching

A method of keypoint detection must be used to obtain keypoints from a sequence of images. The FAST keypoint detector (Features from Accelerated Segment Test) is frequently used for keypoint detection due to its speed, and is used for quickly eliminating unsuitable matches in ORB. Starting with an image patch p of size $31x31$, each pixel is compared with a Bresenham circle built 45 degrees at a time by $x_{n+1}^2 = x_n^2 - 2y(n) - 1$. The radius of the surrounding circle of points is nominally 3, but is 9 for the ORB descriptor, which expands the patch size and number of points in the descriptor. If at least 75% of the pixels in the circle are contiguous and more than some threshold value above or below the pixel value, a feature is considered to be present [RD05]. The ORB algorithm introduces an orientation measure to FAST by computing corner orientation by intensity centroid, defined as

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \quad where \;\; m_{pq} = \sum_{x,y} x^p y^q I(x,y). \tag{1}$$

The patch orientation can then be found by $\theta = \text{atan2}(m_{01}, m_{10})$ and is Gaussian smoothed. ORB then applies the BRIEF feature descriptor $f_n(p) = \sum_{1 \le i \le n} 2^{i-1} \tau(p; a_i, b_i)$, a bit string result of binary intensity tests τ, each of which is defined from the intensity $p(a)$ of a point at a relative to the intensity $p(b)$ at a point at b by [RD05]

$$\tau(p; a, b) = \left\{ \begin{array}{ll} 1 & : p(a) < p(b) \\ 0 & : p(a) \ge p(b) \end{array} \right\} \tag{2}$$

The descriptor is also steered according to the orientations computed for the FAST keypoints by rotating the feature set of points (a_i, b_i) in $2xn$ matrix form by the patch orientation θ to obtain the rotated set F[RRKB11].

$$F = R_f \left(\begin{array}{ccc} a_1 & \cdots & a_n \\ b_1 & \cdots & b_n \end{array} \right). \tag{3}$$

The steered BRIEF operator used in ORB then becomes $g_n(p, \theta) = f_n(p) \vee (a_i, b_i) \in F$. A lookup table of steered BRIEF patterns is constructed from this to speed up computation of steered descriptors in subsequent points.

Keypoints are then matched between two images in the sequence by attempting to find a corresponding keypoint a' in the second image that matches each point a in the first image, which can be done exhaustively by an XOR operation between each descriptor and a population count to obtain the Hamming distance. However, The FLANN (Fast Library for Approximate Nearest Neighbor) search algorithm built into OpenCV is used in current work as it performs much faster while still providing good matches [ML09].

The more features in common between these images, the more potentially good matches M_f can be found, but it is essential that matches be correct correspondences or a valid transformation between the two images will be impossible. The matches M_f are first

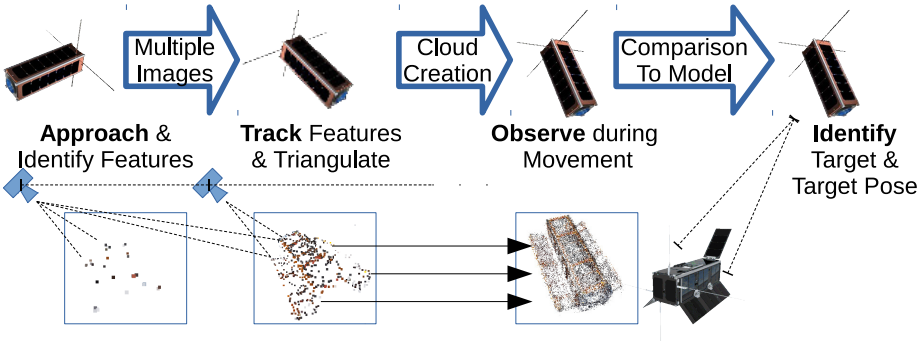

Figure 1. Process of Ego-Motion and Target Pose Estimation

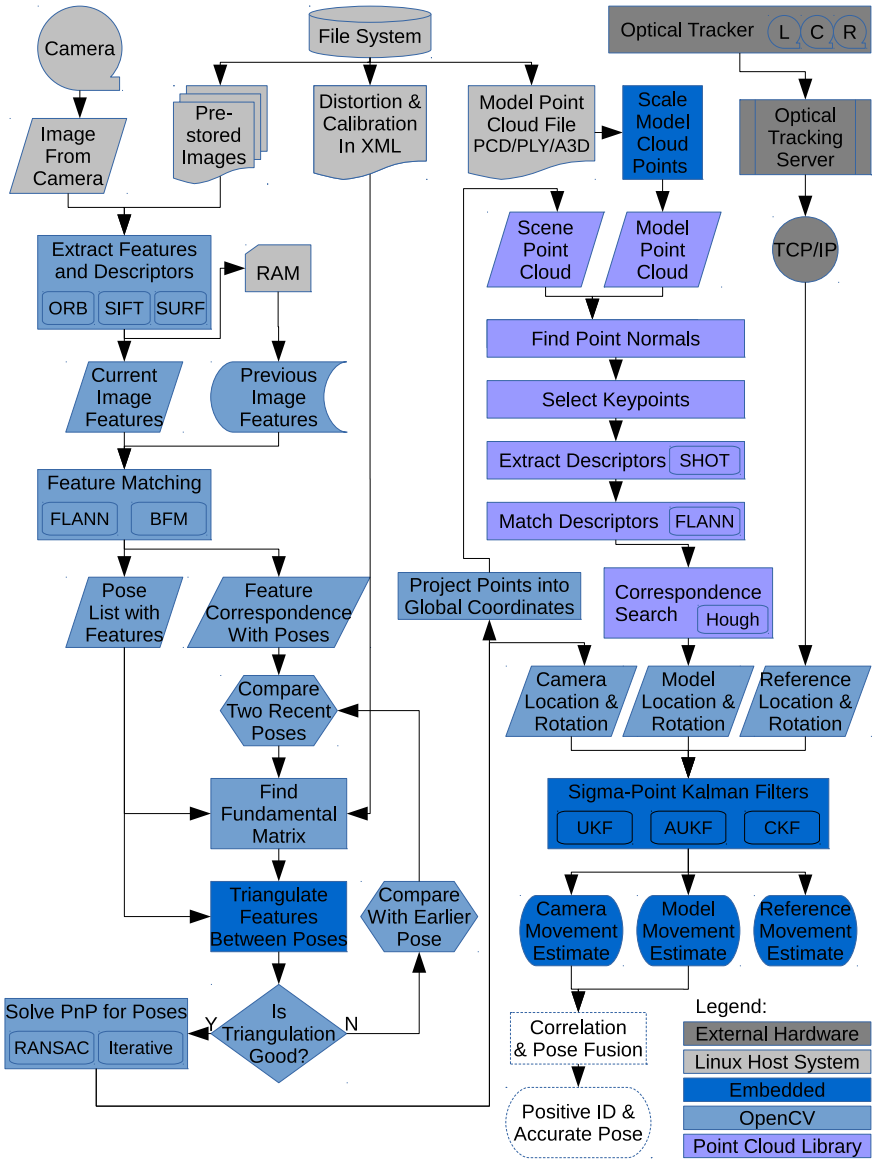

Figure 2. Flowchart of Optical Pose Estimation System

coarsely pruned of bad pairings by finding the maximum distance between points d_{max} and then removing all matches that have a coordinate distance d_a of more than half the maximum distance between features using $M_g = M_f(a)|d_a < d_{max}/2$.

2.3. Three-Dimensional Projection

To obtain depth in a 3-D scene, an initial baseline for 3-D projection is first required using either stereoscopic vision, or two sequential images from different angles.. The Fundamental Matrix $\mathbf{F}$ is the transformation matrix that maps each point in a first image to a second image, and the set of "good" matches M_g is used where each keypoint a_i in the first image is expected to map to a corresponding keypoint a_i' on the epipolar line in the second image by the relation $a_i'^T \mathbf{F} a_i = 0,\ i = 1, \ldots, n$ [LF95]. For three-dimensional space, this equation is linear and homogeneous and the matrix F has nine unknown coefficients, so $\mathbf{F}$ can be uniquely solved for by using eight keypoints with the method of Longuet-Higgins [LH87]. However, due to image noise and distortion, linear least squares estimation (i.e. $\min_F \sum_i (a_i'^T \mathbf{F} a_i)^2$) or RANSAC [FB81] must be used to ensure that a "best" solution can be estimated. We use RANSAC for its speed to estimate $\mathbf{F}$ for all matches M_g and estimate the associated epipolar lines [FH03]while removing outliers more than 0.1 from their epipolar line from M_g to yield a final, reliable set of keypoint matches M_h. To perform a projection into un-distorted space, a calibration matrix $\mathbf{K}$ is needed, either from calibration with a known pattern such as a checkerboard [Har97], or estimated for a size $w \times h$ image as

$$\mathbf{K} = \begin{pmatrix} \max(w,h) & 0 & w/2 \\ 0 & \max(w,h) & h/2 \\ 0 & 0 & 1 \end{pmatrix}. \qquad (4)$$

A camera matrix is defined as $\mathbf{C} = \mathbf{K}[\mathbf{R}|\mathbf{t}]$ with the rotation matrix $\mathbf{R}$ and the translation vector $\mathbf{t}$ defining the pose of the camera in space, and for two images, we define two camera matrices $\mathbf{C1}$ and $\mathbf{C2}$. To localize a point in un-distorted space, we formulate the so-called essential matrix $\mathbf{E} = \mathbf{t} \times \mathbf{R} = \mathbf{K}^T \mathbf{F} \mathbf{K}$ that relates two matching undistorted points $\hat{x}$ and $\hat{x}'$ in the camera plane as $\hat{a}_i'^T \mathbf{E} \hat{a}_i = 0,\ i = 1, \ldots, n$ [HS97]. In this way, $\mathbf{E}$ includes the "essential" assumption of calibrated cameras [Shi12b], and is related to the fundamental matrix by $\mathbf{E}$

After calculating $\mathbf{E}$, we can find the location of a second camera $\mathbf{C2}$ by assuming for simplicity that the first camera is uncalibrated and located at the origin ($\mathbf{C1} = [I|0]$). We decompose $\mathbf{E} = \mathbf{t} \times \mathbf{R}$ into its component $\mathbf{R}$ and $\mathbf{t}$ matrices by using the singular value decomposition of $\mathbf{E}$ [HZ04]. We start with the orthogonal matrix $\mathbf{W}$ and and singular value decomposition (SVD) of $\mathbf{E}$, defined as

$$\mathbf{W} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{SVD}(\mathbf{E}) = \mathbf{U} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \mathbf{V}. \qquad (5)$$

The matrix $\mathbf{W}$ does not directly depend on $\mathbf{E}$, but provides a means of factorization for $\mathbf{E}$. Detailed proofs can be found in [HZ04] and are not reproduced here, but there are two possible factorizations of $\mathbf{R}$, namely $\mathbf{R} = \mathbf{U}\mathbf{W}^\mathbf{T}\mathbf{V}^\mathbf{T}$ and $\mathbf{R} = \mathbf{U}\mathbf{W}\mathbf{V}^\mathbf{T}$, and two possible choices for $\mathbf{t}$, namely $\mathbf{t} = \mathbf{U}(0,0,1)^T$ and $\mathbf{t} = -\mathbf{U}(0,0,1)^T$. Thus when determining the second camera matrix $\mathbf{C2} = \mathbf{K}[\mathbf{R}|\mathbf{t}]$, we have four choices in total.

it is now possible to triangulate the original un-distorted point positions in space with $\mathbf{E}$ and a pair of matched keypoints $[\mathbf{a} = (a_x, a_y), \mathbf{b} = (b_x, b_y)] \in M_h$ using iterative linear least-squares triangulation [HS97]. A point in three dimensions $\mathbf{x} = (x_x, x_y, x_z, 1)$ written in the matrix equation form $\mathbf{A}\mathbf{x} = 0$ results in four linear nonhomogeneous equations in four unknowns for an appropriate choice of $\mathbf{A}_{4x4}$. To solve this, we can write the system as $\mathbf{A}\mathbf{x} = \mathbf{B}$, with $\mathbf{x} = (x_x, x_y, x_z)$, and $\mathbf{A}_{4x3}$ and $\mathbf{B}_{4x1}$ as defined by Shil [Shi12a]. The solution $\mathbf{x}$ by SVD is transformed to un-distorted space by $\hat{\mathbf{x}} = \mathbf{KC1x}$, assuming that the point is neither at 0 nor at infinity. This triangulation must be performed four times for each combination of $\mathbf{R}$ and $\mathbf{t}$ and tested by perspective transformation with $\mathbf{C1}$ and $\hat{\mathbf{x}}_z > 0$ to ensure the resulting points p_i are in front of the camera.

2.4. Image Selection

Using adjacent pairs of images in a closely-spaced time sequence allows feature points to be tracked more reliably between images, as there is less chance of conditions or change in angle causing a feature to change significantly. However, the disadvantage of using closely-spaced images for pose estimation is that a very small angular difference between two images will prevent triangulation solutions, like very distant points. Therefore, we track, match, and store keypoints between closely-spaced images, but only triangulate with images that are well-separated that contain tracked keypoints between the two. Unusable images in the matching process are most commonly due to:

- Not enough feature points being matched to obtain $\mathbf{F}$ or triangulate
- Inaccurate estimates of rotation $\mathbf{R}$ and translation $\mathbf{t}$
- Inaccuracy of the fundamental matrix $\mathbf{F}$, preventing decomposition to $\mathbf{E}$, $\mathbf{R}$, and $\mathbf{t}$

If two few features are matched between image P_t at time step t and P_{t-1}, the next image to be obtained P_{t+1} is used with P_{t-1}, if it fails then P_{t+2} is used, and so on

until a predefined "reset" limit. Valid matches from the new image P_t or later are added the the existing tracked keypoint list to associate feature numbers across the sequence of images. When obtaining the fundamental matrix $\mathbf{F}$, only keypoints that have been associated between both images are used.

2.5. Position Estimation

To finding the ego-motion of the tracker's camera relative to feature points represents the Perspective & Point (PnP) problem. For this, we apply the OpenCV implementation of the EPnP algorithm [MNLF07]. For the n-point cloud with points $\mathbf{p}_1 \dots \mathbf{p}_n$, four control points c_i define the world coordinate system and are chosen with one point at the centroid of the point cloud and the rest oriented to form a basis. Each reference point is described in world coordinates (denoted with ^{w}) as a linear combination of c_i with weightings α_{ij}. This coordinate system is consistent across linear transforms, so they have the same combination in the camera coordinate system (denoted with ^{c}. The known two-dimensional projections $\mathbf{u_i}$ of the reference points $\mathbf{p_i}$ are linked to these weightings by $\mathbf{K}$ considering that the projection involves scalar projective parameters w_i, leading to the following.

$$\mathbf{p}_i^w = \sum_{j=1}^{4} \alpha_{ij} \mathbf{c}_j^w,\ \mathbf{p}_i^c = \sum_{j=1}^{4} \alpha_{ij} \mathbf{c}_j^c,\ \sum_{j=1}^{4} \alpha_{ij} = 1 \quad (6)$$

$$\mathbf{K}\mathbf{p}_i^c = w_i \begin{pmatrix} \mathbf{u}_i \\ 1 \end{pmatrix} = \mathbf{K} \sum_{j=1}^{4} \alpha_{ij} \mathbf{c}_j^c \quad (7)$$

The expansion of this equation has 12 unknown control points and n projective parameters. Two linear equations can be obtained for each reference point to obtain a system of the form $\mathbf{Mx} = 0$, where the null space or kernel of the matrix $\mathbf{M_{2nx12}}$ gives the solution $\mathbf{x} = [\mathbf{c}_1^{cT}, \mathbf{c}_2^{cT}, \mathbf{c}_3^{cT}, \mathbf{c}_4^{cT}]$ to the system of equations, which can be expressed as $\mathbf{x} = \sum_{i=1}^{m} \beta_i \mathbf{v}_i$. The set $\mathbf{v_i}$ is composed of the null eigenvectors of the product $\mathbf{M}^T\mathbf{M}$ corresponding to m null singular values of $\mathbf{M}$. The method of solving for the coefficients $\beta_1 \dots \beta_m$ depends on the size of m, and four different methods are used in the literature [MNLF07] for practical solution.

Let the translation and rotation in world coordinates of the previous pose be $\mathbf{t}_w(t-1)$ and $\mathbf{R}_w(t-1)$, and that of the current pose be $\mathbf{t}_w(t)$ and $\mathbf{R}_w(t)$, for which we need to find the current camera matrix in world coordinates $\mathbf{C}_w(t)$. The relative transformation between the camera positions $\mathbf{t}(t)$ and $\mathbf{R}(t)$ is used to incrementally advance the current pose (assumed to be attached rigidly to the camera) as $\mathbf{C}_w(t) = [\mathbf{R}_w(t-1)\mathbf{R}(t)|\mathbf{R}(t)\left(\mathbf{t}(t) + \mathbf{t}_w(t-1)\right)]$., and feature points are incrementally projected into world coordinates with $x' = \left(\mathbf{R}_w(t-1)\mathbf{R}(t)\right)^T x + \mathbf{R}_w(t-1)\left(\mathbf{t}(t) + \mathbf{t}_w(t-1)\right)$. Orientation is stored as a quaternion from the elements r_{ij} of $\mathbf{R}_w$.

$$\mathbf{q} = \begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{1+r_{00}+r_{11}+r_{22}}}{2} \\ \frac{r_{21}-r_{12}}{2\sqrt{1+r_{00}+r_{11}+r_{22}}} \\ \frac{r_{02}-r_{20}}{2\sqrt{1+r_{00}+r_{11}+r_{22}}} \\ \frac{r_{10}-r_{01}}{2\sqrt{1+r_{00}+r_{11}+r_{22}}} \end{bmatrix} \quad (8)$$

3. OBSERVATION AND IDENTIFICATION

The PnP solution across a sequence of images allows us to track the pose of the tracker spacecraft relative to features on the target spacecraft. However, in most cases it is necessary to identify what the actual orientation of the target is with respect to a known geometric model, or to identify specific parts of the target for interaction or analysis. For this task, we use the positional correspondences of three-dimensional keypoints selected from the constructed point cloud with respect to keypoints selected from a reference model point cloud that can be obtained in advance or on-line from another sequence of images with known relative pose. Model recognition is done on a per-pose basis with accumulated points in the point cloud once a sufficient number of images has been acquired during the "Observation" phase. This makes it possible to match parts of a structure without requiring the entire structure to have keypoints, for example if the target is in partial shadow. We use an Unscented Kalman filter (UKF) for reducing noise over time for pose estimates. Separate filtering is performed for the pose estimates obtained from PnP solutions and target pose estimation, both translation and quaternion rotation, using a fast embedded UKF implementation with adaptive statistics [LPL13].

3.1. Object Pose Estimation

A set of three-dimensional keypoints are chosen from both the scene and the model by picking individual points from the cloud separated by a given sampling radius. Normals are calculated for these keypoints relative to nearby points so that each keypoint has a repeatable orientation. The keypoints are then associated with three-dimensional SHOT (Signature of Histograms of OrienTations) descriptors [STDS14]. SHOT descriptors are calculated by grouping together a set of local histograms over the volumes about the keypoint, where this volume is divided into by angle into 32 spherically-oriented spatial bins. Within a given radius of the keypoint, point counts from the local histograms are binned as a cosine function $\cos(\theta_i) = \mathbf{n}_u \cdot \mathbf{n}_{v_i}$ of the angle θ_i between the point normal within the corresponding part of the structure $\mathbf{n}_{v_i}$ and the feature point normal $\mathbf{n}_u$. This has the beneficial effects of creating a general rotational invariance since angles are relative to local normals, accumulating points into different bins as a result of small differ-

ences in relative directions, and creating a coarse partitioning that can be calculated fast with small cardinality.

Comparing the scene keypoint descriptors with the model keypoint descriptors to find good correspondence matches is done using a FLANN search on a k-dimensional tree (k-d tree) structure, similarly to the matching of image keypoints. Additionally, the BOrder Aware Repeatable Directions algorithm for local reference frame estimation (BOARD) is used to calculate local reference frames for each three-dimensional SHOT descriptor [PDS11] to make them independent of global coordinates for rotation and translation invariance. Once a set of nearest correspondences and local reference frames is found, clustering of correspondences to given cluster sizes is performed by pre-computed Hough voting to make recognition of shapes more robust to partial occlusion and clutter [TDS10].

Evidence of a particular pose and instance of the model in the scene is initialized before voting by obtaining the vector between a unique reference point C^M and each model feature point F_i^M and transforming it into local coordinates by the transformation matrix $R_{GL}^M = [L_{i,x}^M, L_{i,y}^M, L_{i,z}^M]^T$ from the local x-y-z reference frame unit vectors $L_{i,x}^M$, $L_{i,y}^M$, and $L_{i,z}^M$. This precomputation can be done offline for the model in advance and is performed by calculating for each feature a vector $V_{i,L}^M = [L_{i,x}^M, L_{i,y}^M, L_{i,z}^M] \cdot (C^M - F_i^M)$. For online pose estimation, Hough voting is performed by each scene feature F_j^S that has been found by FLANN matching to correspond with a model feature F_i^M, casting a vote for the position of the reference point C^M in the scene. The transformation $R^M S_L$ that makes these points line up can then be transformed into global coordinates with the scene reference frame unit vectors, scene reference point F_j^S and scene feature vector $V_{i,L}^S$ as $V_{i,G}^S = [L_{j,x}^S, L_{j,y}^S, L_{j,z}^S] \cdot V_{i,L}^S + F_j^S$. The votes cast by $V_{i,G}^S$ are thresholded to find the most likely instance of the model in the scene, although multiple peaks in the Hough space are fairly common and can indicate multiple possibilities for model instances. Due to the statistical nature of Hough voting, it is possible to recognize partially-occluded or noisy model instances, though accuracy may be lower.

3.2. Processing Times

To profile the processing requirements of the described algorithms on a system that could potentially be embedded into a satellite, the algorithm was run on a $667MHz$ ARM Cortex-A9 processor with pre-defined images of a CubeSat engineering model in VGA resolution and precomputed point clouds, and raw timing statistics gathered for the processing time of each algorithm. Tests 1 and 2 were performed with 6524 model points and 5584 scene points from 220 images, and tests 3 and 4 were performed with 6524 model points and 1816 scene points from 32 images. Tests 1 and 3 were performed with a descriptor radius of 0.05 and cluster size of 0.1, and Tests 2 and 4 were performed with a descriptor radius of 0.1 and cluster size of 0.5. Tab. 1 and Tab. 2 show the timing information obtained for each of the described algorithms in these cases.

Table 3. Correspondences and Error resulting from varying Descriptor Radius and Cluster Size

Estimate	Descr. Radius	Cluster Size	Correspondences	Trans. Error	Rotation Error
1	2.0	1.0	507	1%	2%
2	2.0	0.1	507	7%	3%
3	0.5	1.0	45	3%	4%

3.3. Identification Accuracy

To illustrate the accuracy of pose estimation while varying the descriptor radius and cluster size and therefore processing times, a set of pose estimation tests were performed using a CubeSat engineering model as a target for pose identification. In three examples of target identification shown in Fig. 3, Fig. 4, and Fig. 5, high-density model points are in yellow with selected keypoints in green, and low-density scene keypoints are shown in blue. The model instance found in the scene is overlaid in red from a high-density model composed of 26339 points, while the scene is composed of 1960 points triangulated from 52 images. The number of keypoints was reduced by radius to 2042 in the model and 1753 in the scene. The descriptor radius and cluster size for these estimates, with the resulting number of correspondences and rounded cumulative errors in translation and rotation are shown in Tab. 3.

As more scene points are added over time, accuracy can increase, but only if they are consistent with the existing scene. We can see from these results that increasing the size of the SHOT descriptor will increase the number of keypoints available and result in better accuracy and higher likelihood of identifying a shape, but also will require longer processing times. Cluster sizes must be set appropriately for the point cloud size, as a cluster size too small or too large will prevent valid instances from being found, and result in decreased accuracy.

4. CONCLUSIONS

In this work, we have described a feature-based visual identification system that allows a tracker spacecraft to track relative movement to a target and ultimately acquire pose estimates using point cloud techniques. Using projective geometry, we perform three-dimensional reconstruction of features on the target from a sequence of images taken with a single camera. The patent-free

Table 1. Timing for Features, Triangulation and PnP

Test Number	Feature Detect.	Feature Matching	Feature Selection	Fundam. Matrix	Essential Matrix	Triangu-lation	PnP RANSAC	Ego-Motion	Total Time
1-2	0.12	0.058	0.015	0.083	0.0017	0.038	0.0033	0.0005	0.32
3-4	0.12	0.061	0.010	0.048	0.0014	0.025	0.0026	0.0004	0.27

Table 2. Timing for Correspondence and Identification

Test Number	Model Normals	Scene Normals	Model Sampling	Scene Sampling	Model Keypoints	Scene Keypoints	FLANN Search	Cluster-ing	Total Time
1	0.17	0.15	0.027	0.020	1.26	0.84	107.7	0.92	112.1
2	0.17	0.15	0.029	0.024	3.37	2.19	118.0	2.00	127.2
3	0.17	0.043	0.031	0.0083	3.31	0.37	42.5	0.63	48.4
4	0.17	0.041	0.031	0.0078	3.31	0.37	42.6	1.36	49.1

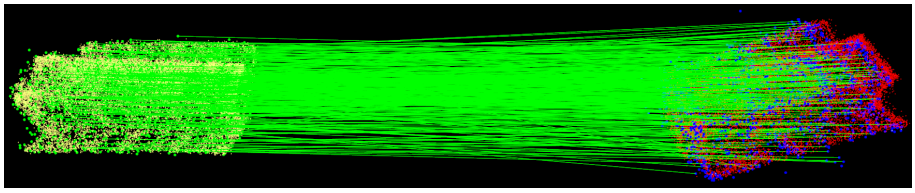

Figure 3. Pose Correspondence for Estimate 1, Descriptor Radius 2.0*, Cluster Size* 1.0

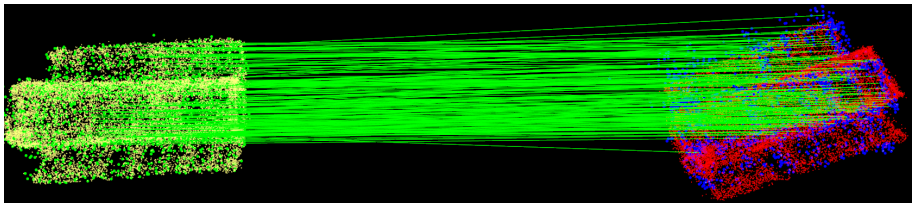

Figure 4. Pose Correspondence for Estimate 1, Descriptor Radius 2.0*, Cluster Size* 0.1

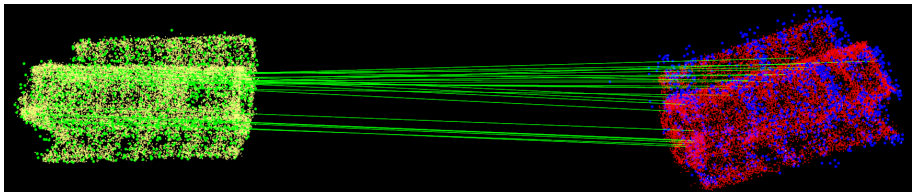

Figure 5. Pose Correspondence for Estimate 1, Descriptor Radius 0.5*, Cluster Size* 1.0

ORB algorithm that combines FAST keypoint detection and BRIEF feature descriptors provides good tolerance to rotation and scaling of features for this purpose. For useful reconstruction, it is important to identify as many features as possible, so target spacecraft with many colors, edges, and shapes generally provide the best results for feature-based systems such as this. It is important to note that this method of motion estimation provides best solutions through post-processing of results. The more images that are included when creating the structure, the better triangulation will be. If processing power and storage is available to include a large number of recent images, such as by observing the target through multiple rotations, a better solution for motion will be obtained. To additionally decrease the processing time if desired, the camera image can be lowered in resolution, or pixels can be under-sampled by choosing only every 2nd pixel or every 4th pixel in a staggered pattern over the image for feature matching [AZK09].

It is intended that even small spacecraft such as nanosatellites with a single camera could take advantage of this system. Work is underway to scale this system to a level suitable for nanosatellite use, which could provide a technology demonstration with a minimum of cost and risk. As the performance of feature tracking depends very heavily on the design of the feature descriptor and method of matching, further comparison of descriptor types for both two-dimensional and three-dimensional matching is warranted. Future work also includes the validation of these methods on a variety of models, and under a broader set of varying conditions to evaluate the robustness of feature-based systems. A wide variety of applications for this technology is also available, including robotic uses and planetary rover navigation and sensing.

REFERENCES

[AZK09] K. Ambrosch, C. Zinner, and W. Kubinger. Algorithmic considerations for real-time stereo vision applications. In *MVA09*, page 231, 2009.

[FB81] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981.

[FH03] C. L. Feng and Y. S. Hung. A robust method for estimating the fundamental matrix. In *In International Conference on Digital Image Computing*, pages 633–642, 2003.

[Har97] Richard Hartley. Self-calibration of stationary cameras. *International Journal of Computer Vision*, 22(1):5–23, 1997.

[HCDS14] Heather Hinkel, Scott Cryan, Christopher DSouza, and Matthew Strube. Nasa's automated rendezvous and docking/capture sensor development and its applicability to the ger. 2014.

[HS97] Richard I. Hartley and Peter Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2):146 – 157, 1997.

[HZ04] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[LF95] Q.-T. Luong and O.D. Faugeras. The fundamental matrix: theory, algorithms, and stability analysis. *International Journal of Computer Vision*, 17:43–75, 1995.

[LH87] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. In Martin A. Fischler and Oscar Firschein, editors, *Readings in computer vision: issues, problems, principles, and paradigms*, pages 61–62. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1987.

[LPL13] J. Li, M. A. Post, and R. Lee. A novel adaptive unscented kalman filter attitude estimation and control system for a 3u nanosatellite. In *12th biannual European Control Conference*, Zurich, Switzerland, 17-19 July 2013.

[ML09] Marius Muja and David G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Application (VISSAPP'09)*, pages 331–340. INSTICC Press, 2009.

[MNLF07] F. Moreno-Noguer, V. Lepetit, and P. Fua. Accurate non-iterative o(n) solution to the pnp problem. In *IEEE International Conference on Computer Vision*, Rio de Janeiro, Brazil, October 2007.

[Oda00] Mitsushige Oda. Experiences and lessons learned from the ets-vii robot satellite. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 1, pages 914–919. IEEE, 2000.

[PDS11] Alioscia Petrelli and Luigi Di Stefano. On the repeatability of the local reference frame for partial shape matching. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2244–2251. IEEE, 2011.

[PHAR12] Jose Padial, Marcus Hammond, Sean Augenstein, and Stephen M Rock. Tumbling target reconstruction and pose estimation through fusion of monocular vision and sparse-pattern range data. In *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2012 IEEE Conference on*, pages 419–425. IEEE, 2012.

[RD05] Edward Rosten and Tom Drummond. Fusing points and lines for high performance tracking. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1508–1515 Vol. 2, 2005.

[RLB12] Stéphane Ruel, Tim Luu, and Andrew Berube. Space shuttle testing of the tridar 3d rendezvous and docking sensor. *Journal of Field Robotics*, 29(4):535–553, 2012.

[RRKB11] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary R. Bradski. Orb: An efficient alternative to sift or surf. In *ICCV 2011*, pages 2564–2571, 2011.

[RT04] Michael Ruth and Chisholm Tracy. Video-guidance design for the dart rendezvous mission. In *Defense and Security*, pages 92–106. International Society for Optics and Photonics, 2004.

[Sha14] S. Sharma. Pose estimation of uncooperative spacecraft using monocular vision. In *Invited Student Presentation at Stanford's 2014 PNT Challenges and Opportunities Symposium, Kavli Auditorium, SLAC*, 10 2014.

[Shi12a] Roy Shil. Simple triangulation with opencv from harley & zisserman. Online, January 2012.

[Shi12b] Roy Shil. Structure from motion and 3d reconstruction on the easy in opencv 2.3+. Online, February 2012.

[STDS14] Samuele Salti, Federico Tombari, and Luigi Di Stefano. Shot: unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding*, 125:251–264, 2014.

[TBB11] T Tzschichholz, T Boge, and H Benninghoff. A flexible image processing framework for vision-based navigation using monocular imaging sensors. In *Proceedings of the 8th international ESA conference on guidance, navigation & control systems. Karlovy Vary, Czech Republic*, 2011.

[TDS10] Federico Tombari and Luigi Di Stefano. Object recognition in 3d scenes with occlusions and clutter by hough voting. In *Image and Video Technology (PSIVT), 2010 Fourth Pacific-Rim Symposium on*, pages 349–355. IEEE, 2010.

[TSSO+14] Brent E Tweddle, Timothy P Setterfield, Alvar Saenz-Otero, David W Miller, and John J Leonard. Experimental evaluation of on-board, visual mapping of an object spinning in micro-gravity aboard the international space station. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 2333–2340. IEEE, 2014.