Article

Visual Monocular 3D Reconstruction and Component Identification for Small Spacecraft

Mark Post [1,†,‡], **Junquan Li** [2,‡]

[1] University of Strathclyde, Glasgow, United Kingdom; mark.post@strath.ac.uk
[2] Space Innovation Robotics Ltd, Glasgow, United Kingdom; junquanl@spaceibot.com
* Correspondence: mark.post@strath.ac.uk; Tel.: +44-0141-574-5274
† Current address: Department of Design, Manufacture and Engineering Management, University of Strathclyde, 75 Montrose St. Glasgow, United Kingdom, G1 1XJ
‡ These authors contributed equally to this work.

Abstract: A monocular vision pose estimation and identification algorithm used on a small spacecraft for future orbital servicing is studied in this paper. A tracker spacecraft equipped with a short-range vision system is proposed to recover the 3D structural model of a space target in orbit and automatically identify its solar panels and main body using only visual information from an onboard camera. The proposed reconstruction and identification framework is tested using structure-from-motion and point cloud identification methods. The Efficient Perspective-n-Points (EPnP) descriptor is used for pose estimation. Triangulated points are used for component segmentation by means of orientation histogram descriptors. Experimental results based on laboratory images of a spacecraft model show the effectiveness and robustness of our approach.

Keywords: spacecraft; structure from motion; monocular vision; component detection; structure analysis

1. Introduction

Space object 3D reconstruction, pose estimation and identification is very important for spacecraft orbital servicing and space situational awareness based on satellite imaging. Structure from Motion technology and pose estimation has attracted a lot of interest as an enabling technology for detecting, tracking, cataloguing and identifying satellites and spacecraft in recent years. Structure from Motion (SfM) is a method for obtaining 3-D structures using only monocular feature matches between multiple images at multiple angles, which can include lines (e.g. Canny edge detection), corners (e.g. Harris corner detection, and other types of features. It also represents a natural progression into point cloud techniques from feature-based ego-motion estimation between pairs of images. 3D reconstruction and identification has been studied extensively, but in order for such systems to work effectively on small spacecraft with only single visual sensors, the implementation of point cloud building, image feature point matching, sparse reconstruction, identification strategy and dimensional analysis information must be considered.

High performance optical imaging sensors-such as radar, lidar, visible and infrared are used for detecting, tracking and identifying objects in orbit. Table 1 shows a list of existing on-orbit servicing missions and demonstrations. 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 [1]. Recent automated rendezvous and docking systems make use of optical, laser ranging, and LIDAR systems [2] [3] and visually-aided systems have been tested in proximity operations with NASA's Space Shuttle, JAXA's ETS-VII satellite [4] as well as other satellites such as the DART mission [5]. The Rendezvous Lidar System (RLS) has also been tested on the XSS-11 spacecraft for rendezvous operations. However, the complexity, size,

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 January 2018 doi:10.20944/preprints201801.0195.v1

Table 1. Summary of On-Orbit Servicing Missions

Name	Mission	Sensors	Time	Status
ESA GSV [14]	Vision Inspection, Robot Operation and Debris	Vision Cameras	1994-1996	Concept
DLR ESS [15]	Vision Inspection, Satellite Capture, Docking/Release	Laser Range Finder and Stereo Vision Cameras	1994-1997	Concept
ESA ROGER [16]	Web Capture Satellite, Space Object and Debris	Stereo Cameras, Laser Range Finder and Zoom Camera	2003	Concept
ESA OLEV [17]	Life extension for Servicing Commercial GEO Spacecrafts	Stereo Cameras and Far field Camera	2002	Demonstration
DLR DEOS [18]	On-Orbit Servicing Mission	Vision Camera and LIDAR	2004	Ongoing Mission
NICT OMS [19]	Orbital Maintenance System	SVGA resolution COTS C-MOS Imager/ARM 9 Processor and Star Sensor	2005	Concept
USA XSS11[20]	Autonomous Rendezvous and Proximity Manoeuvres	Space-borne Scanning Lidar and Combined Vision Camera and Star Tracker by SAIC	2005	On-Orbit Demonstration
DAPRA HRV [21]	Servicing Hubble Space Telescope	Vision Camera and RADAR	2005	Concept
JAXA SDMR [22]	Testing debris using Tether for Small Satellite	GPS, Vision Sensors and Star Tracker	2006	Demonstration
NRL FREND [23]	Non-cooperative Capture Servicing and recycling	Solid State LIDAR and Stereo Vision Cameras	2007	Demonstration Ground Testing
CSA ACTS [24]	Autonomous capture and servicing of satellites	Laser Camera System from Neptec	2006	Demonstration
DAPRA RSGS [25]	Robotic Servicing of Geosynchronous Satellites	Under Design	2017	Concept

and power requirements of current LIDAR systems are still out of reach for small spacecraft, 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 [6] have been proposed for this, and typically focus on shape tracking and recognition, feature detection and triangulation [7], or a combination of shape and features [8]. The SPHERES experiment uses SURF feature matching with stereo vision for navigation inside the ISS [9]. Images of space objects using visible cameras are low resolution and lack texture information. These methodologies are related to computer vision challenges in terms of extreme lighting conditions, as specular reflection and hard shadows can lead to mission failure. A lot of studies have been done using Kalman-filter and other classic vision algorithms with 3D vision sensors for spacecraft on-orbit servicing [10] [11]. There are a few related works that handle satellite recognition, pose estimation, 3D reconstruction and identification using vision only as well as using structure from motion [12] [13].

Based on the authors' previous work [12], 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

 doi:10.20944/preprints201801.0195.v1

coding errors and complexity, we make use of the open-source OpenCV (Open Computer Vision) and PCL (Point Cloud Library) libraries for most of the computer vision programming. We consider the situation of a rendezvous zone where spacecraft are separated by several meters or tens of meters, with the intention of matching velocity and attitude for rendezvous. Precise manoeuvring and capture requires the use of short-range sensing on the satellite itself. Outside of the range where optical sensors are useful, other sensors can be used for coarse positioning and estimation such as GNSS and telemetry from ground tracking stations. The flexibility of visual-only pose estimation also means that it has many potential applications in other fields such as planetary rover navigation, but the movement of hardware complexity to software complexity in vision systems requires a corresponding increase in computing resources. Hardened computing hardware for space can take between several seconds to several minutes for simple image recognition tasks. The Mars Exploration Rovers required 42 seconds to process a single image pair for navigation with no recognition task [26]. In this work, the ORB descriptor is used with FLANN matching as an open alternative to SIFT and SURF for feature detection. Point Cloud Library provides the framework for processing, storage, and visualization of the point cloud, and a review of multiple-view geometry used to create a point cloud from multiple poses is provided. We also add the components identification and dimensional analysis. The proposed vision pose estimation and identification system shows good performance in experimental results. This work is intended to be applied and evaluated on a real mission in the near future.

The contributions of this work are summarized as follows. We review the following machine vision methods and how they are implemented:

1. ORB descriptor 2D feature detection and matching between images
2. Multi-view feature triangulation and PnP solution for ego-motion (structure-from-motion)
3. Characterization of point cloud shapes using the 3D SHOT descriptor
4. Point cloud correspondence using FLANN and Hough voting for object and partial object recognition

We also perform the following laboratory tests using an engineering model of a small satellite sequentially imaged at multiple angles to simulate observation of a tumbling target by a tracker satellite:

5. Identification and dimensional analysis of small satellite components by comparing a component model to a scene
6. Comparison of the effects of variation in SHOT parameters to pose identification accuracy
7. Investigation of the effects of partial spacecraft occlusion on pose identification accuracy
8. Evaluation of timing required for processing on a representative embedded processor

The structure of this paper is as follows: Section 1 provides the background and value of our work. Overall workflow and principals are described in Section 2. The results and discussion are shown in Section 3. The conclusions are given in Section 4.

2. Overview of Framework

To allow a tracker spacecraft to to identify and estimate the movement of a target spacecraft, we approach this problem as illustrated in Figure 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 and Analyze” phase. In the end, the tracker spacecraft with robot arms and end effectors is intended to perform a projected “Capture and Servicing” phase.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 January 2018 doi:10.20944/preprints201801.0195.v1

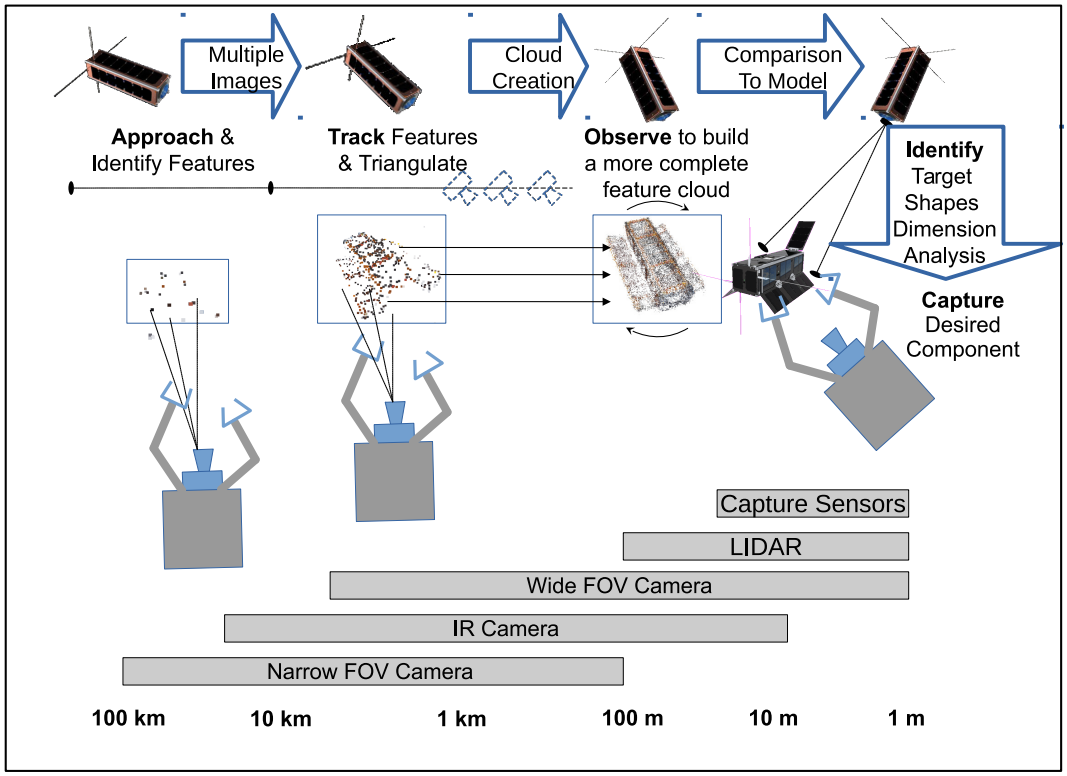

Figure 1. Process of Orbit Servicing for Small Satellite

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 [27]. We then use structure-from-motion methods to triangulate these points in space.

2.1. 3D Reconstruction from Camera

A flowchart of the process we propose is shown in Figure 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.

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 points, 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

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 January 2018 doi:10.20944/preprints201801.0195.v1

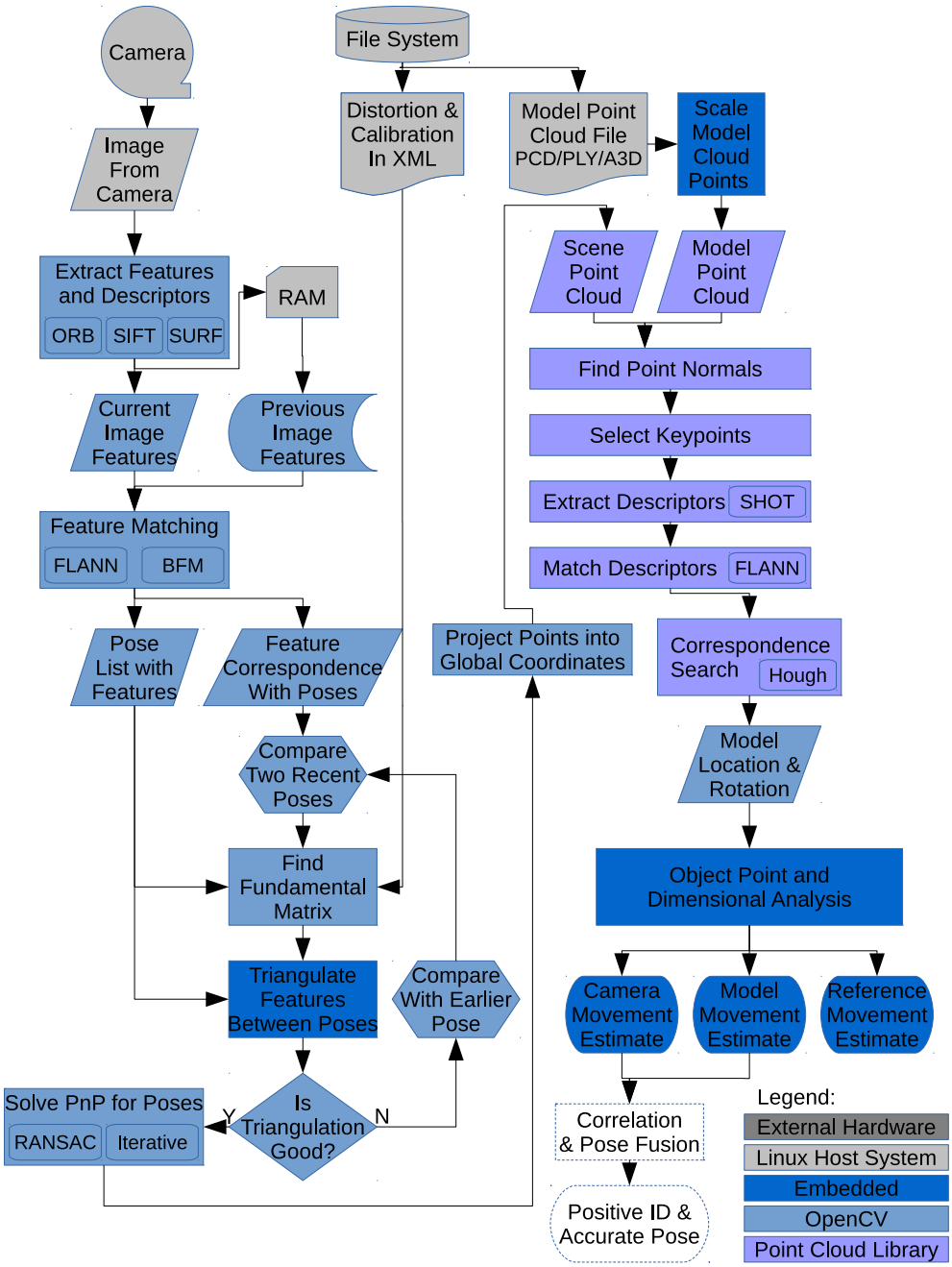

Figure 2. Flowchart of Vision Detection and Analysis System

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 January 2018 doi:10.20944/preprints201801.0195.v1

or below the pixel value, a feature is considered to be present [28]. The ORB algorithm introduces an orientation measure to FAST by computing corner orientation by intensity centroid, defined as

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

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 $f_n(p) = \sum_{1 \leq i \leq n} 2^{i-1} \tau(p; a_i, b_i)$, 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 [28]

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

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[27].

$$F = R_f \begin{pmatrix} a_1 & \cdots & a_n \\ b_1 & \cdots & b_n \end{pmatrix}. \tag{3}$$

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 [29].

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 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 **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 $a_i'^T \mathbf{F} a_i = 0, \; i = 1, \ldots, n$ [30]. For three-dimensional space, this equation is linear and homogeneous and the matrix F has nine unknown coefficients, so **F** can be uniquely solved for by using eight keypoints with the method of Longuet-Higgins [31]. 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 [32] must be used to ensure that a "best" solution can be estimated. We use RANSAC for its speed to estimate **F** for all matches M_g and estimate the associated epipolar lines [33]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 **K** is needed, either from calibration with a known pattern such as a checkerboard [34], or estimated for a size $w \times h$ image as

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 January 2018 doi:10.20944/preprints201801.0195.v1

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

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 **C1** and **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 $\hat{a}_i'^T\mathbf{E}\hat{a}_i = 0$, $i = 1, \ldots, n$ [35]. In this way, $\mathbf{E}$ includes the "essential" assumption of calibrated cameras [36], and is related to the fundamental matrix by $\mathbf{E}$.

After calculating $\mathbf{E}$, we can find the location of a second camera **C2** by assuming for simplicity that the first camera is uncalibrated and located at the origin (**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 $\mathbf{E}$ [37]. We start with the orthogonal matrix $\mathbf{W}$ and and singular value decomposition (SVD) of $\mathbf{E}$, defined as

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

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 [37] 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 **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 [35]. 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 [38]. 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 **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.

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.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 January 2018 doi:10.20944/preprints201801.0195.v1

2.5. Pose 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 [39]. 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.

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

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

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 $\mathbf{x} = [\mathbf{c}_1^{c\,T}, \mathbf{c}_2^{c\,T}, \mathbf{c}_3^{c\,T}, \mathbf{c}_4^{c\,T}]$ 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 [39] 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 $\mathbf{C}_w(t) = [\mathbf{R}_w(t-1)\mathbf{R}(t)|\mathbf{R}(t)\,(\mathbf{t}(t) + \mathbf{t}_w(t-1))]$., and feature points are incrementally projected into world coordinates with $x' = (\mathbf{R}_w(t-1)\mathbf{R}(t))^T\, x + \mathbf{R}_w(t-1)\,(\mathbf{t}(t) + \mathbf{t}_w(t-1))$. Orientation is stored as a quaternion from the elements r_{ij} of $\mathbf{R}_w$.

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

2.6. Object Pose Estimation

Object pose estimation focuses on the 3D reconstruction of the object and from the 3D point cloud features are extracted to detect and classify the detect. There are four main segmentation methods: local features [40] [41] [42], global features [43], graph matching [44] and machine learning [45] [46]. The use of point clouds in the presence of noise, varying mesh resolutions or poorly textured objects, clutter, and occlusion are very challenging [47] [48]. Segmentation in unstructured environments is difficult [49]. The image data for spacecraft and satellites in orbit are also often distorted and partially occluded due to shadowing. Using 3D point cloud-based recognition methods emphasizes overall shape and configuration over texture and can tolerate a degree of distortion and occlusion. We test the proposed system of using 3D keypoint descriptors by using images of a real satellite model.

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

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 January 2018 doi:10.20944/preprints201801.0195.v1

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. These 3D keypoints (not to be confused with the 2D keypoints used for triangulation) provide a means to compare models on a per-pose basis with accumulated points in the scene 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. It also allows us to match parts of the target separately given a sufficient number of points in the part that we are matching to.

2.7. Target Identification

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. In the case that multiple matches are identified, a criteria for determining which one is the most appropriate is necessary. We choose the match with the largest number of corresponding keypoints as the most likely correct match.

2.8. Satellite Component Identification

The remote capture of spacecraft is a highly sensitive operation that is carefully planned beforehand to minimize the chance of error. For this reason, an automated grasp planner is not a good fit for orbital capture of a known spacecraft. Rather, the exact point on the spacecraft should be specified beforehand using three-dimensional models, and the grasp planned based on the model and knowledge of the spacecraft's structure. The grasping operation can then be executed based on the position and motion of the target component. It is also necessary to verify the extents of the component and the whole spacecraft to ensure that no accidental contact is made during the grasping operation, which could cause both target and chaser to spin and separate before the grasp is completed.

Satellite components are identified by first preparing exemplar point clouds, such as a model of a solar panel, that can be stored and used for reference by the tracker spacecraft. These model point clouds are then located in the actual reconstructed 3-D scene point cloud created by the tracker spacecraft. We focus on the solar panels as an example of external satellite components that are easy to grasp and manipulate in a rendezvous operation, and the body of the spacecraft that indicates overall positioning. Solar panels may also not remain at a precise angle with respect to the spacecraft body, and therefore must be identified in isolation from the spacecraft body to ensure accuracy. The identification process begins with a set of three-dimensional keypoints being chosen from both the scene and the model by randomly choosing individual points from the cloud separated by a given sampling radius r_k. Normals are calculated for these keypoints relative to nearby points so that each

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 January 2018 doi:10.20944/preprints201801.0195.v1

keypoint has a repeatable orientation. The keypoints are then associated with three-dimensional SHOT point descriptors.

SHOT descriptors [41] 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 r_d 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 differences in relative directions, and creating a coarse partitioning that can be calculated fast with small cardinality [50].

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 [51] 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 set by a parameter r_c is performed by pre-computed Hough voting to make recognition of shapes more robust to partial occlusion and clutter [52]. At least a threshold of n_{thresh} votes in Hough space is needed to estimate a valid pose.

3. Results

3.1. 3D Reconstruction and Identification

To test the identification of small satellite components, we use an engineering model of a small satellite with full-length fold-out solar panels, shown in Figure 3. This satellite serves as an example target for a simulated tracker satellite.

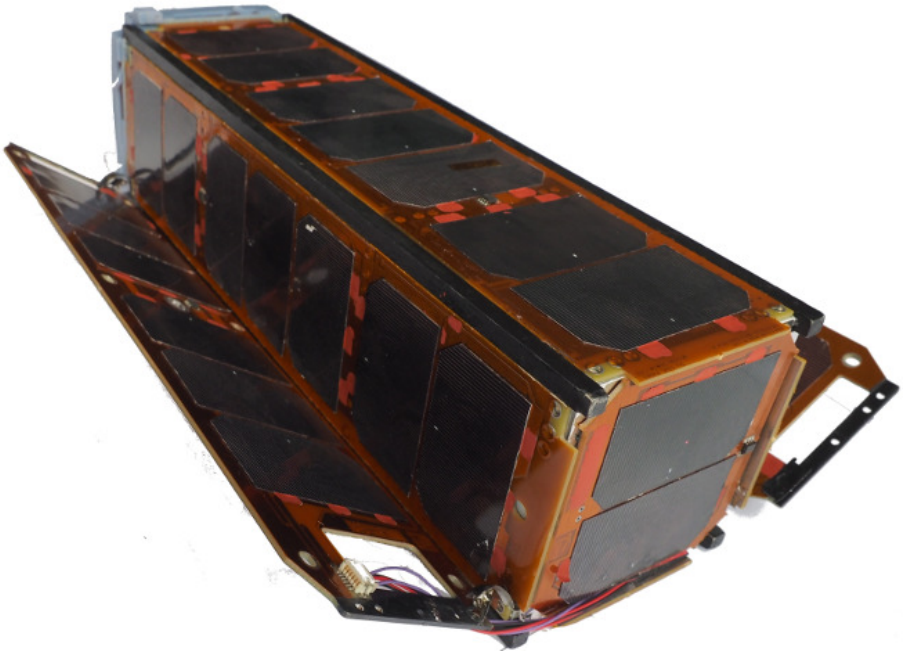

Figure 3. Small satellite engineering model

A three-dimensional point cloud of this satellite was created in the laboratory by simulating what a tracker satellite in close proximity would observe as the target tumbles at low relative rotational speed. Rather than rotating the target, the camera was robotically moved at low speed in an arc around

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 January 2018 doi:10.20944/preprints201801.0195.v1

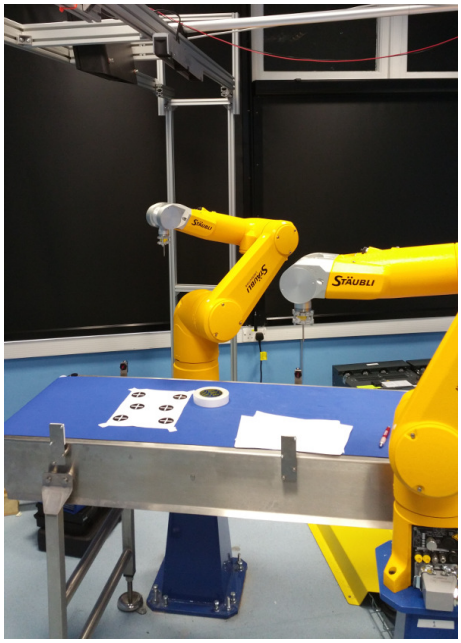

Figure 4. Robot Arm used for positioning camera

Figure 5. Olympus OM-D Camera used for imaging of satellite model

the target satellite in 10 degree increments with the background obscured by a paper screen to prevent unrelated features from being detected and images taken of the target. A high-intensity light source was used to simulate direct sunlight. The robot arm used for motion of the camera is shown in Figure 4 and the Olympus OM-D E-M1 camera in Figure 5. Through 10 complete rotations around the satellite at slightly different angles of view, a sufficiently dense point cloud was triangulated for component recognition. All images were converted to VGA resolution (640x480 pixels) to decrease processing time and demonstrate the feasibility of low-resolution point cloud recognition. Figure 6 shows the scene as reconstructed by the simulated tracker spacecraft.

Using the same process, point clouds were obtained of a solar panel and the satellite body itself. Figure 7 shows the point clouds generated for these components.

3.2. 3D Component Identification

Each of the component point clouds shown in Figure 7 was sequentially matched with the scene of the small satellite in Figure 6. To illustrate the matching process, the point cloud matched for each component is marked in yellow with keypoints indicated in green, and the scene is in full colour with keypoints marked in blue. The points of the matched component within the scene are indicated in red to show where the component's location has been identified.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 January 2018 doi:10.20944/preprints201801.0195.v1

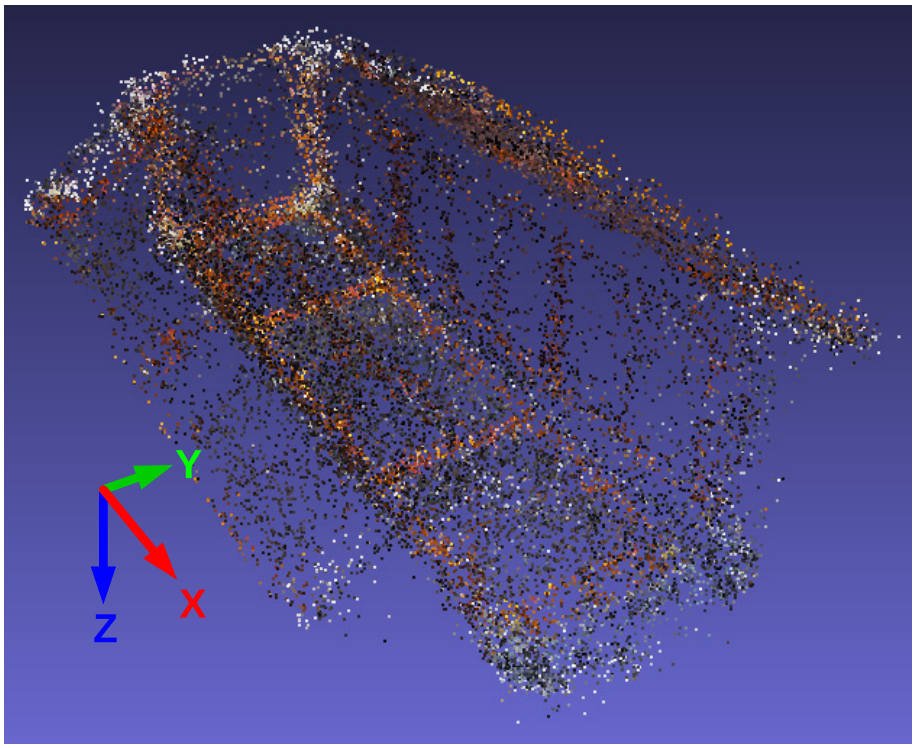

Figure 6. Scene reconstructed from tracker spacecraft using structure-from-motion

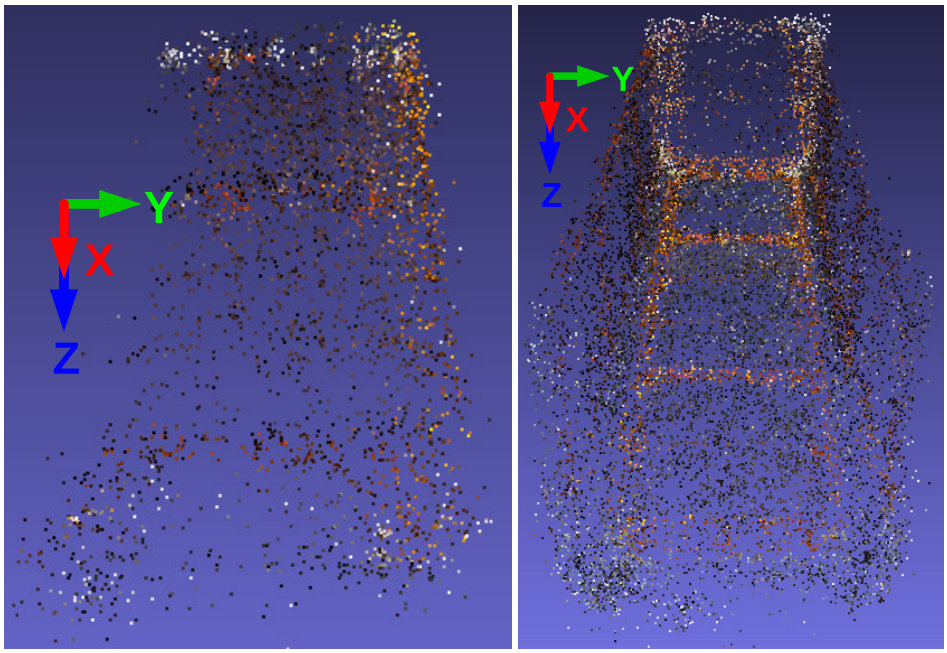

Figure 7. Spacecraft component point clouds: solar panel (left), spacecraft body (right)

First, the solar panels were matched. Figure 8 shows the best match for the solar panel model, which corresponds with the left-side solar panel in the scene. Figure 9 shows a lower-likelihood match, which corresponds to the right-side solar panel in the model. Using the model of the satellite body, Figure 10 shows the body of the satellite identified.

The parameters used for the SHOT descriptors in these tests were a model and scene sampling radius of $r_k = 0.025m$, reference frame and descriptor radius of $r_d = 0.5m$, cluster size of $r_c = 0.25m$,

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 January 2018 doi:10.20944/preprints201801.0195.v1

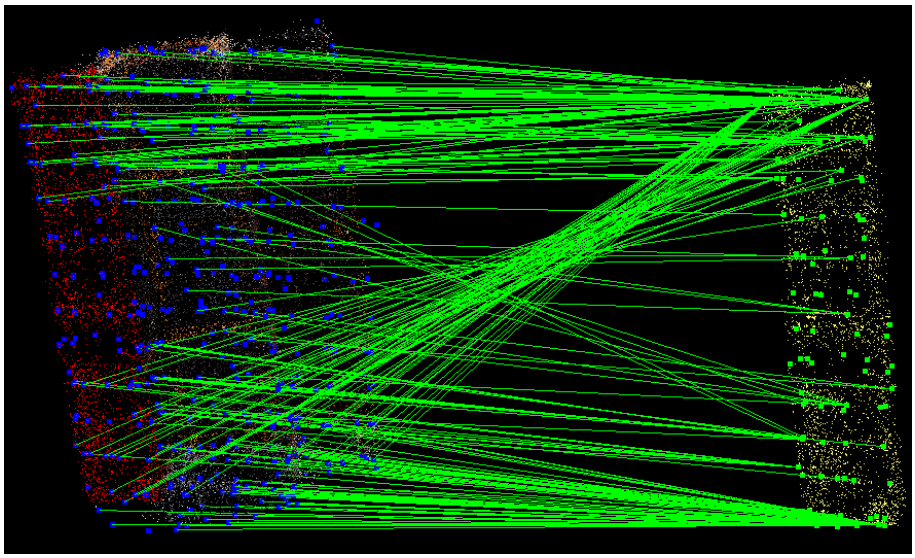

Figure 8. Matched location (red points) for left solar panel component (yellow points)

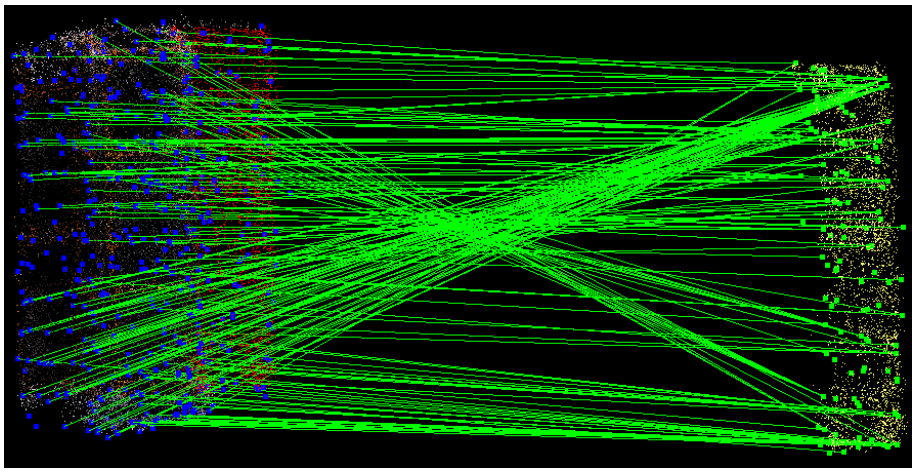

Figure 9. Matched location (red points) for right solar panel component (yellow points)

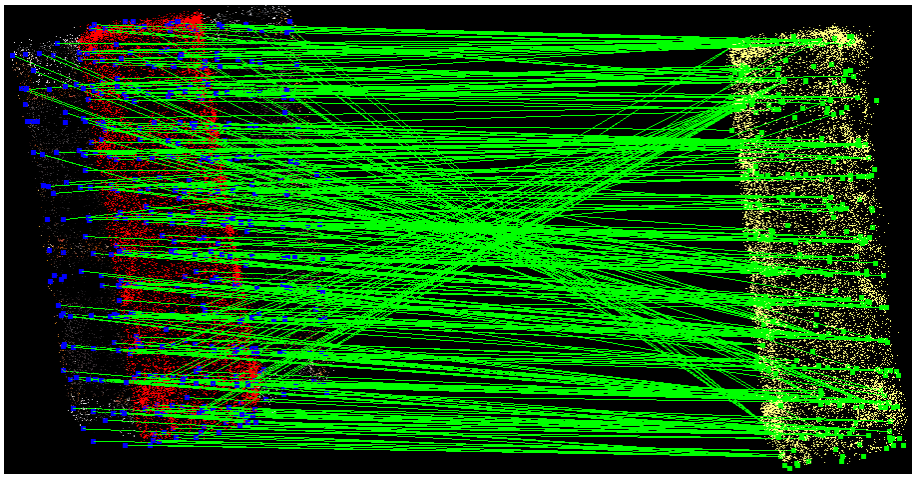

Figure 10. Matched location (red points) for body component (yellow points)

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 January 2018 doi:10.20944/preprints201801.0195.v1

and clustering threshold of $n_{thresh} = 5$. The number of correspondences and percentage of error observed in both rotation and translation is shown in Table 2. As there are less keypoints in smaller components such as the solar panels, they exhibit higher error in correspondence. Increasing the number of keypoints (and computational time) serves to mitigate this problem.

Table 2. Correspondences and Error resulting from varying Descriptor Radius and Cluster Size

Component	Corresp -ondences	Translation Error	Rotation Error
Left Solar Panel	243	3%	5%
Right Solar Panel	186	3%	6%
Spacecraft Body	375	2%	3%

3.3. Dimensional Analysis

For each component identified on the spacecraft, we in addition estimate its size for purposes of planning and grasping for the chaser spacecraft. Table 3 shows the dimensions of the components estimated during the identification process, compared with actual measurements of size. The measurements of size in each direction are performed with respect to the coordinate axes for each component model, and simply indicate the extents of the scene points that have been matched with the model. For the spacecraft considered here, this is suitable since all components are rectangular in form except for the entire satellite as a unit. The detected dimensions of each component are larger than their actual values because the scene points exhibit some degree of statistical variation due to numerical inaccuracies during the triangulation process, and this must be accounted for in planning and control of capture operations as well. This is particularly true for the Z axis measurement of the thin solar panels. The closer and more accurately the chaser spacecraft can observe the target, the smaller these triangulation errors will be, since triangulation error increases with distance..

Table 3. Dimensional Analysis of Spacecraft Components

Component	Size X (m)	Size Y (m)	Size Z (m)	Actual X (m)	Actual Y (m)	Actual Z (m)
Left Solar Panel	0.307	0.103	0.025	0.300	0.100	0.002
Right Solar Panel	0.309	0.117	0.032	0.310	0.100	0.002
Body	0.336	0.112	0.120	0.315	0.100	0.100
Satellite	0.337	0.230	0.115	0.315	0.264	0.100

3.4. Parameter Effects on Spacecraft Pose Identification Accuracy

To illustrate the accuracy of pose estimation while varying the descriptor radius r_d and cluster size r_c and therefore processing times, a set of pose estimation tests were performed. These tests use a point cloud of the complete spacecraft that was generated from a different series of images so that a different point cloud with the same shape could be matched against the scene. In three examples of target identification shown in Figure 11, Figure 12, and Figure 13, 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.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 January 2018 doi:10.20944/preprints201801.0195.v1

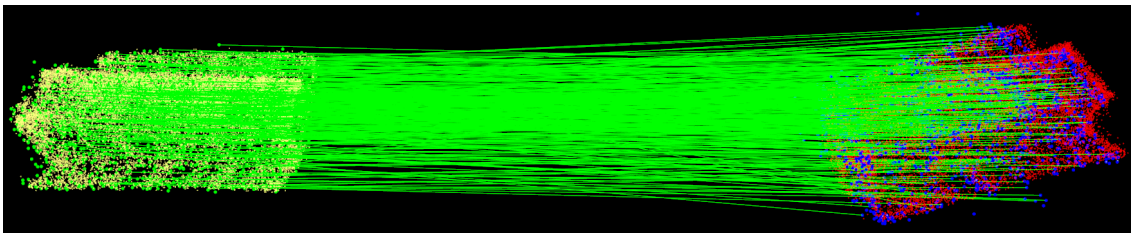

Figure 11. Pose Correspondence for Estimate 1, Descriptor Radius 0.5m, Cluster Size 0.25m

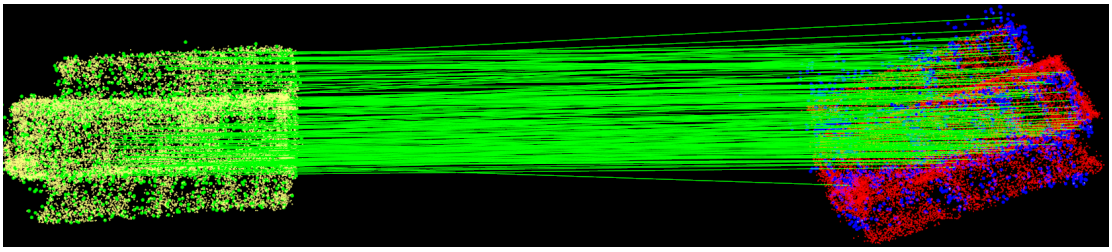

Figure 12. Pose Correspondence for Estimate 1, Descriptor Radius 0.5m, Cluster Size 0.025m

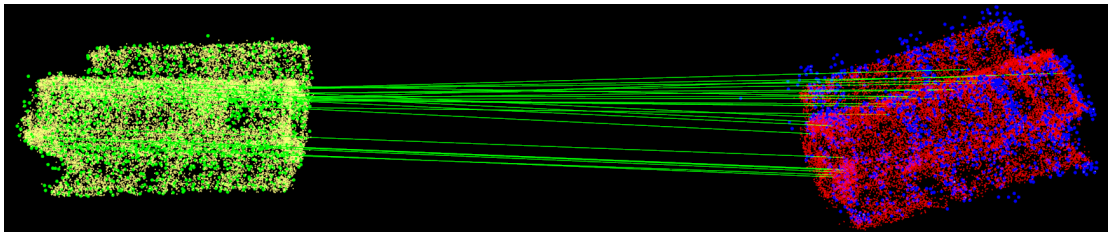

Figure 13. Pose Correspondence for Estimate 1, Descriptor Radius 0.125m, Cluster Size 0.25m

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 Table 4.

Table 4. Correspondences and Error resulting from varying Descriptor Radius and Cluster Size

Estimate	Descr. Radius (m)	Cluster Size (m)	Corresp -ondences	Translation Error	Rotation Error
1	0.5	0.25	507	1%	2%
2	0.5	0.025	507	7%	3%
3	0.125	0.25	45	3%	4%

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.

The patent-free 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

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 January 2018 doi:10.20944/preprints201801.0195.v1

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 [53].

3.5. Occlusion Effects on Spacecraft Pose Identification Accuracy

In the space environment, it is common that components are partially or fully occluded by shadows, which can be cast by either the chaser spacecraft or other components of the target spacecraft. These shadows are total in an airless environment and prevent any features from being detected in a shadowed scene. To evaluate the effects of partial shadowing on the small spacecraft model, features were removed from the scene point cloud used in previous tests so that along the length of the spacecraft, the first 25%, 50%, and then 75% of features are in shadow, as shown in Figure 14, Figure 15, and Figure 16 respectively. All tests use a descriptor radius $r_d = 0.5m$ and a cluster Size $r_c = 0.25m$.

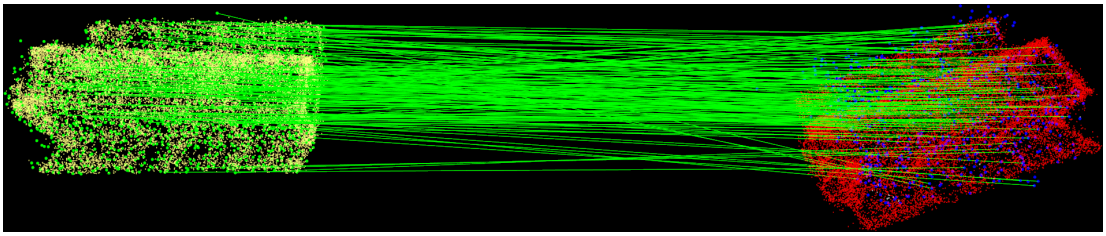

Figure 14. Pose Correspondence for 25% of the scene in shadow

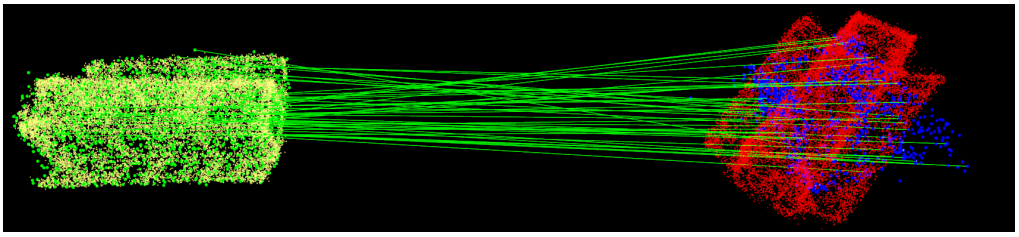

Figure 15. Pose Correspondence for 50% of the scene in shadow

Figure 16. Pose Correspondence for 75% of the scene in shadow

The effects of this occlusion on the model-to-scene correspondence and pose estimation accuracy are summarized in Table 5. A small amount of shadow over a quarter of the scene has a tolerable but noticeable effect on both translation and rotation estimation. However, with half of the target shadowed, translation error increases in a linear fashion while rotation error increases much more

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 January 2018 doi:10.20944/preprints201801.0195.v1

quickly due to the high sensitivity of rotation estimation to the observed point cloud shapes in the scene. With three-quarters of the target in shadow, no pose estimate can be found as the scene point cloud no longer bears a similar enough shape to the model.

Table 5. Correspondences and Error resulting from varying Descriptor Radius and Cluster Size

Percent Shadowed	Translation Error	Rotation Error
25%	4%	8%
50%	8%	21%
75%	No Pose	No Pose

3.6. Timing and Profiling

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 over the VGA images of the satellite engineering model used above, 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. Table 6 and Table 7 show the timing information obtained in seconds for each of the described algorithms in these cases. While accurate matching of large models and scenes can take on the order of minutes, this does not prevent a chaser spacecraft from building a motion model over long periods of time from stored images before acting to rendezvous, and both software and hardware acceleration methods may be used to further improve this performance.

Table 6. Timing for Features, Triangulation and PnP in seconds

Test Num.	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 7. Timing for Correspondence and Identification in seconds

Test Num.	Model Normals	Scene Normals	Model Sampling	Scene Sampling	Model Keypoints	Scene Keypoints	FLANN Search	Clustering	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

4. Conclusions

This study proposes a 3D pose estimation, recognition and identification system for a small spacecraft servicing mission that uses a monocular camera sensor. This study uses Structure from Motion (SFM) to build 3D model from 2D images and a SHOT descriptor to identify surface shape components. The EPnP process estimates object poses and increases the system's ability to identify position and angles. The experimental results show that the proposed system can effectively identify components and poses of a spacecraft model in the lab. Potential application of this system to an orbital demonstration mission with industry partners is under investigation.

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. It is intended that even small spacecraft with a single camera could take advantage of this system. Work is underway to scale this system to a level suitable for small satellite 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, and FPGA acceleration is being developed for this system. Future work also includes the validation of these methods with a variety of different spacecraft and vision hardware, and under a broader set of varying conditions to evaluate the robustness of feature-based systems.

Author Contributions: All authors contributed equally to this manuscript. Dr Post is the principal author of this manuscript and is responsible for programming and experimental testing. Dr Li contributed background research and analysis of the results. Both are responsible for the writing of the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ruel, S.; Luu, T.; Berube, A. Space shuttle testing of the TriDAR 3D rendezvous and docking sensor. *Journal of Field Robotics* **2012**, *29*, 535–553.
2. Hinkel, H.; Cryan, S.; DSouza, C.; Strube, M. NASA's Automated Rendezvous and Docking/Capture Sensor Development and Its Applicability to the GER. *NASA Report* **2014**.
3. Padial, J.; Hammond, M.; Augenstein, S.; Rock, S.M. Tumbling target reconstruction and pose estimation through fusion of monocular vision and sparse-pattern range data. Multisensor Fusion and Integration for Intelligent Systems (MFI), 2012 IEEE Conference on. IEEE, 2012, pp. 419–425.
4. Oda, M. Experiences and lessons learned from the ETS-VII robot satellite. Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on. IEEE, 2000, Vol. 1, pp. 914–919.
5. Ruth, M.; Tracy, C. Video-guidance design for the DART rendezvous mission. Defense and Security. International Society for Optics and Photonics, 2004, pp. 92–106.
6. Zhang, H.; Jiang, Z.; Elgammal, A. Satellite recognition and pose estimation using homeomorphic manifold analysis. *IEEE Transctions on Aerospace and Electronic Systems* **2015**, *51*, 785–793.
7. Sharma, S. Pose Estimation of Uncooperative Spacecraft using Monocular Vision. Stanford's 2014 PNT Challenges and Opportunities Symposium, Kavli Auditorium, SLAC, 2014.
8. Tzschichholz, T.; Boge, T.; Benninghoff, H. A flexible image processing framework for vision-based navigation using monocular imaging sensors. Proceedings of the 8th international ESA conference on guidance, navigation & control systems. Karlovy Vary, Czech Republic, 2011.
9. Tweddle, B.E.; Setterfield, T.P.; Saenz-Otero, A.; Miller, D.W.; Leonard, J.J. Experimental evaluation of on-board, visual mapping of an object spinning in micro-gravity aboard the International Space Station. Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on. IEEE, 2014, pp. 2333–2340.
10. Aghili, F.; Kuryllo, M.; Okouneva, G.; English, C. Fault-tolerant position/attitude estimation of free-floating space objects using a laser range sensor. *IEEE Sensors Journal* **2011**, *11*, 176 – 185.

11. Flores-Abad, A.; Ma, O.; Pham, K.; Ulrich, S. A review of space robotics technologies for on-orbit servicing. *Progress in Aerospace Sciences* **2014**, *68*, 1 – 26.
12. Post, M.A.; Yan, X.T.; Li, J.; Clark, C. Visual Pose Estimation System for Autonomous Rendezvous of Spacecraft. 13th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA 2015) - Noordwijk, Netherlands, 2015, pp. 1–9.
13. Zhang, H.; Wei, Q.; Jiang, Z. 3D reconstruction of space objects from multi-Views by a visible sensor. *Sensors* **2017**, *7*, 1 – 16.
14. Depeuter, W.; Visentin, G.; Fehse, W. Satellite servicing in GEO byrobotic service vehicle. *ESA Bulletin-European Space Agency* **1994**, *7*, 22 – 25.
15. Hirzinger, G.; Landzettel, K.; Brunner, B.; Fischer, M.; Preusche, C.; Reintsema, D.; Albu-Schäffer, A.; Schreiber, G.; Steinmetz, B.M. DLR's robotics technologies for on-orbit servicing. *Advanced Robotics* **2004**, *18*, 139 – 174.
16. B, B.; Kerstein, L. ROGER-robotic Geostationary Orbit Restorer. 54th International Astronautical Congress of the International Astronautical Federation, Bremen, Germany, 2003.
17. Kaisera, C.; Sjöbergb, F.; Delcurac, J.M.; Eilertsen, B. SMART-OLEV an orbital life extension vehicle for servicing commerial spacecrafts in GEO. *Acta Astronautica* **2008**, *63*, 400 – 410.
18. Rupp, T.; Boge, T.; Kiehling, R.; Sellmaier, F. Flight Dynmacis Challenges of Germanon-Orbit Servicing Mission. International Symposium on Space Flight Dynamics Toulouse France September 27- October 2, 2009.
19. Kimura, S.; Nagai, Y.; Yamamoto, H.; Masuda, K.; Abe, N. Approach for on Orbit Maintenance and Experiment Plan using 150kg Class Satellites. IEEE Aerospace Conference Big Sky USA March 5-12, 2005.
20. Richards, R.; Tripp, J.; Pashin, S.; King, D.; Bolger, J.; Nimelman, M. Advances in Automous Orbital Rendezvous Technology: The XSS-11 Lidar Sensor. Proceedings of the 57th IAC/IAF/IAA (International Astronautical Congress) Valencia Spain October 2-6, 2005.
21. Thienel, J.K.; Sanner, R.M. Hubble space telescope angular velocity esitmatin during the robotic servicing mission. *Journal of Guidance Control and Dynamics* **2007**, *30*, 29 – 34.
22. Nishida, S.I.; Kawamoto, S.; Okawa, Y.; Terui, F.; Kitamura, S. Space debris removal system using a small satellite. *Acta Astronautica* **2009**, *65*, 95 – 102.
23. Thomas, D.; Sean, D. Overview and Performance of the Front-end Robotics Enabling Near-term Demonstration. AIAA Infotech Aerospace Conference Seattle USA April 6-9, 2009.
24. Rekleitis, I.; Martin, E.; Rouleau, G.; L'Archevêque, R.; Parsa, K.; Dupuis, E. Autonomous capture of a tumbling satellite. *Journal of Field Robotics* **2007**, *24*, 275 – 296.
25. Roesler, G. Robotic Servicing of Geosynchronous Satellites DARPA. Online, 2016.
26. Goldberg, S.B.; Maimone, M.W.; Matthies, L. Stereo vision and rover navigation software for planetary exploration. Aerospace Conference Proceedings, 2002. IEEE. IEEE, 2002, Vol. 5, pp. 5–2025.
27. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G.R. ORB: An efficient alternative to SIFT or SURF. ICCV 2011, 2011, pp. 2564–2571.
28. Rosten, E.; Drummond, T. Fusing points and lines for high performance tracking. Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, 2005, Vol. 2, pp. 1508–1515.
29. Muja, M.; Lowe, D.G. Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration. International Conference on Computer Vision Theory and Application (VISSAPP'09). INSTICC Press, 2009, pp. 331–340.
30. Luong, Q.T.; Faugeras, O. The Fundamental matrix: theory, algorithms, and stability analysis. *International Journal of Computer Vision* **1995**, *17*, 43–75.
31. Longuet-Higgins, H.C. A computer algorithm for reconstructing a scene from two projections. In *Readings in computer vision: issues, problems, principles, and paradigms*; Fischler, M.A.; Firschein, O., Eds.; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1987; pp. 61–62.
32. Fischler, M.A.; Bolles, R.C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **1981**, *24*, 381–395.
33. Feng, C.L.; Hung, Y.S. A Robust Method for Estimating the Fundamental Matrix. In International Conference on Digital Image Computing, 2003, pp. 633–642.
34. Hartley, R. Self-Calibration of Stationary Cameras. *International Journal of Computer Vision* **1997**, *22*, 5–23.
35. Hartley, R.I.; Sturm, P. Triangulation. *Computer Vision and Image Understanding* **1997**, *68*, 146 – 157.

36. Shil, R. Structure from Motion and 3D reconstruction on the easy in OpenCV 2.3+. Online, 2012.
37. Hartley, R.I.; Zisserman, A. *Multiple View Geometry in Computer Vision*, second ed.; Cambridge University Press, ISBN: 0521540518, 2004.
38. Shil, R. Simple triangulation with OpenCV from Harley & Zisserman. Online, 2012.
39. Moreno-Noguer, F.; Lepetit, V.; Fua, P. Accurate Non-Iterative O(n) Solution to the PnP Problem. IEEE International Conference on Computer Vision Rio de Janeiro, Brazil, 2007.
40. Johnson, A.E.; Hebert, M. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transctions on Pattern Analysis and Machine Intelligence* **1999**, *21*, 433 – 449.
41. Tombari, F.; Salti, S.; Stefano, I.D. Unique Signatures of Histograms for Local Surface Description. European Conference on Computer Vision, Springer Berlin Heidelberg, 2010, pp. 356–369.
42. Rusu, R.B.; Blodow, N.; Beetz, M. Fast Point Feature Histograms for 3D Registration. IEEE Robotics and Automation Conference ICRA Kobe May 12-17, 2009, pp. 3212–3217.
43. Rusu, R.B.; Blodow, N.; Beetz, M. Fast 3d Recognition and Pose using the Viewpoint Feature Histogram. IEEE/RSJ Internaltional Conference on Intelligent Robots and Systems, 2010, pp. 2155–2162.
44. Hao, W.; Wang, Y. Structure-based object detection from scene point clouds. *Neurocomputing* **2016**, *191*, 148 – 160.
45. Wang, Z.; Zhang, L.; Fang, T.; Mathiopoulos, P.T.; Tong, X.; Qu, H.; Xiao, Z.; Li, F.; Chen, D. A multiscale and hierachical feature extraction method for terrestrial laser scanning point cloud classification. *IEEE Transactions on Geoscience and Remote Sensing* **2014**, *53*, 2409 – 2425.
46. Chen, J.; Fang, Y.; Cho, Y.K. Performance evaluation of 3D descriptors for object recognition in construction applications. *Automation in Construction* **2018**, *86*, 44 – 52.
47. Yang, J.; Zhang, Q.; Xiao, Y.; Cao, Z. TOLDI: An effective and robust approach for 3D local shape description. *Pattern Recognition* **2017**, *65*, 175 – 187.
48. Wohlhart, P.; Lepetit, V. Learning Descriptors for Object Recognition and 3D Pose Estimation. *CoRR* **2015**, *abs/1502.05908*.
49. Choi, C.; Christensen, H.I. RGB-D object pose estimation in unstructured environments. *Robotics and Autonomous Systems* **2016**, *75*, 595 – 613.
50. Salti, S.; Tombari, F.; Di Stefano, L. SHOT: unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding* **2014**, *125*, 251–264.
51. Petrelli, A.; Di Stefano, L. On the repeatability of the local reference frame for partial shape matching. Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, 2011, pp. 2244–2251.
52. Tombari, F.; Di Stefano, L. Object recognition in 3D scenes with occlusions and clutter by Hough voting. Image and Video Technology (PSIVT), 2010 Fourth Pacific-Rim Symposium on. IEEE, 2010, pp. 349–355.
53. Ambrosch, K.; Zinner, C.; Kubinger, W. Algorithmic Considerations for Real-Time Stereo Vision Applications. MVA09, 2009, p. 231.