Image-based attitude determination of co-orbiting satellites using deep learning technologies

Ben Guthrie[a,*], Minkwan Kim[a], Hodei Urrutxua[b], Jonathon Hare[a]

[a] *University of Southampton, University Road, Southampton SO17 1BJ United Kingdom*
[b] *Universidad Rey Juan Carlos, Camino del Molino 5, 28943 Fuenlabrada Madrid Spain*

Abstract

Active debris removal missions pose demanding guidance, navigation and control requirements. We present a novel approach which adopts deep learning technologies to the problem of attitude determination of an uncooperative debris satellite of an a-priori unknown geometry. A siamese convolutional neural network is developed, which detects and tracks inherently useful landmarks from sensor data, after training upon synthetic datasets of visual, LiDAR or RGB-D data. The method is capable of real-time performance while improving upon conventional computer vision-based approaches, and generalises well to previously unseen object geometries, enabling this approach to be a feasible solution for safely performing guidance and navigation in active debris removal, satellite servicing and other close proximity operations. The performance of the algorithm, its sensitivity to model parameters and its robustness to illumination and shadowing conditions, are analysed via numerical simulation.

Keywords: active debris removal, spacecraft attitude determination, deep learning, image processing

1. Introduction

In the past 60 years, the amount of debris in the Low Earth Orbit (LEO) has been increasing steadily [1, 2]. This poses a threat to current space missions,

[*] Corresponding author
Email address: b.f.guthrie@soton.ac.uk (Ben Guthrie)

as highlighted by the 2003 collision between Iridium 33 and Cosmos 2251 [3]. In fact, the Kessler syndrome states that, even if all space launches were to be stopped, the amount of debris would continue to increase [1]. Therefore, it is clear that a method of actively removing debris is required.

Active debris removal (ADR) is a research area which has been a high priority for recent years. The recent RemoveDebris mission was the first to demonstrate many of the required technologies in a space environment, though there remain several challenges to be met before a real debris object can be removed. These challenges include legal aspects, cost, mission design and technological issues. In particular, there are strict requirements on the guidance, navigation and control system [4].

Guidance, navigation and control (GNC) for proximity operations and rendezvous in orbit is very technologically challenging. Due to the time scales and the criticality of this section of an ADR mission, there is increasing interest in the GNC system to be able to perform in a fully autonomous manner. Additionally, it is of crucial importance that the proximity operations do not lead to a collision, thereby adding to the space debris problem. As such, the debris removal satellite must be capable of accurately and robustly determining the position, attitude and tumbling angular velocity of the target.

The GNC system for an ADR mission faces particularly significant challenges where the target parameters are unknown and the target is uncooperative. For such a scenario there are strict GNC requirements that have not yet been conclusively demonstrated in flight in a fully autonomously manner. These may be enumerated as: 1) identification of geometric and physical characteristics of an unknown co-orbiting object; 2) measurement of the target-chaser relative rototranslational state; 3) guidance and navigation around an uncooperative co-orbital object; and 4) capture, stabilisation and de-orbit of an uncooperative spacecraft.

For contactless debris removal missions the chaser would need to safely operate within a few meters of the target [5], whereas for contact methods the spacecraft should be able to dock or berth, thus imposing stringent requirements also to their relative velocity, within 1 cm/s in range rate and 4° in relative angular rate [6]. In addition, the short time scales and high risk of collision make autonomous operations of the GNC system a very desirable feature. However, the uncooperative (and often unknown) nature of the target introduces a further difficulty, thus preventing the straightforward use of readily flight-proven technologies, which rely on prior information about a cooperative satellite, often equipped with docking systems. The control system of the chaser satellite requires the synchronisation of attitude motion with the debris target [7], or alternatively active detumbling of the debris [8], in order to reduce the relative angular rate to within the stated threshold. In either case, an accurate estimate of the attitude and rotational state of the target is required. This is the main focus of our work.

There have not yet been any active debris removal missions to date, except for a small number of ADR demonstration missions with the aim of testing some of the required technologies. The RemoveDebris mission was the first in-orbit demonstration of ADR technologies, including net and harpoon capture mechanisms and a vision-based navigation system [9]. The navigation system consisted of visual, infrared and LiDAR cameras and the measurements were verified using GPS. On-orbit servicing is a similar problem requiring close-range operations, where there have only very recently been successful missions [10, 11]. The Mission Extension Vehicle (MEV) by Northrop Grumman demonstrated the first docking with a satellite which was not built with docking in mind. However, in these missions the target was fully known and cooperative (either fitted with retroreflectors or, in the case of MEV, targeting a liquid apogee engine of known dimensions and properties). Again, in the MEV mission visual, infrared and LiDAR sensors were used. Despite these few demonstration missions, the technical readiness level (TRL) of several ADR technologies is as yet too low to enable any real debris removal missions, so we instead look to similar missions which have demonstrated autonomous GNC in a real environment. Navigation around asteroids and cometary bodies, where the large delay in communications feedback necessitates an autonomous GNC system [12, 13], is a scenario where autonomous GNC systems have been successfully tested. In these cases a 3D model of the asteroid or comet is constructed by extracting key landmarks from images during an observation phase, but often the processing of these landmarks is done off-line.

Visual navigation systems can achieve high accuracies under good conditions, and so are generally accepted as the best solution for autonomous guidance. The selection of the best onboard sensors can vary depending on the applications and requirements. Active LiDAR sensors are one of the most popular choices since they provide a measurement of the depth and are relatively insensitive to illumination conditions, with a wide range of working distances. However, LiDAR sensors have a high power consumption and often a limited field of view. Stereo or RGB-D cameras are other potential options, since these also measure the depth. There are also solutions which propose using a single monocular camera. However, many solutions, such as those of RemoveDebris and Northrop Grumman, use a combination of different sensors.

A conventional approach for visual navigation about an unknown target consists of extracting landmarks from an image which describe the object's pose and tracking the motion of these landmarks to estimate the rotational state, potentially using complex processing techniques such as optical flow or simultaneous localisation and mapping [14]. The pose descriptors are processed within a filtering scheme; several of these filters are compared by Pesce et al [15], the most common being the extended Kalman filter (EKF) [16, 17, 18]. Unfortunately, these algorithms can be computationally intensive, they have no colour saliency and tend to have difficulty in situations with difficult illumination conditions [19, 20]. Some of the complexity can be removed if the geometry of the target is known beforehand [21, 22], though this will not always be true in the most general case.

This paper focuses on the estimation of the instantaneous rotational state of an unknown and uncooperative target from image landmarks, which would then be combined with a filtering system to reconstruct the attitude. It investigates whether deep learning technologies can improve upon the performance of the conventional algorithms, by training a model on datasets under these challenging conditions, as deep learning methods have the advantage of being resistant to non-linearities in the data, such as those caused by varying lighting conditions. Once trained, these models also tend to be fast to run, making them suitable for real-time applications. The downsides are the requirement for large, labelled datasets and the risk of overfitting to the training data.

The domains of image processing and computer vision have seen significant advances in recent years attributed to the deep learning revolution, particularly for object detection and classification tasks [23, 24]. However, in contrast, the applications of deep learning to visual navigation have seen less research, although this area has begun to experience more interest recently [25]. Particularly, most work in this domain has been focused on ground-based problems [26, 27]; in comparison, research on space-based guidance applications has been more limited.

There has been increased interest recently into the applications of machine learning to GNC in space. Sharma and D'Amico [28] present a method for pose estimation with a monocular camera using a convolutional neural network. This work also contributes the Spacecraft Pose Estimation Network (SPEED), which has since been made publicly available through a competition on pose estimation run by the European Space Agency [29]. Other similar research has also looked at pose estimation from monocular images [30, 31], often using finetuned neural network architectures such as ResNet and VGG19. However, in each of these cases, the target spacecraft is known a priori; either the 3D model is provided, or the neural network is trained on a dataset containing a single satellite model. Instead, we aim to solve the more general case, in which the structure of the debris target is entirely unknown. Furthermore, we propose that finetuning a network which has been pretrained on a different image processing task, such as object detection on ImageNet, will result in a network which looks for abstract features, which are not useful in determining the pose of an object.

In this article we propose a novel method for the autonomous, real time estimation of the attitude of an unknown and uncooperative target spacecraft using deep learning technologies. The proposed approach is accurate, robust, and can be used with LiDAR or RGB-D sensors with good results. The same approach is also applicable for monocular cameras; however, the accuracy in this case is heavily impacted by the lack of depth information and so the model requires more refinement before it is able to achieve a similar performance with this restriction. Machine learning technologies have the advantages of being fast and robust to non-linearities, such as varying lighting conditions. However, the drawback of these methods is the requirement for vast amounts of labelled image data, required for training the network. Such a dataset does not exist for this problem, so there has been little investigation into the applications of machine learning in space-based guidance systems. In order to overcome this, synthetic images were generated by simulating the relative motion between two satellites. Thus, the ability of our deep learning model to predict the rotation of a debris satellite is investigated using simulated visual sensor data.

While methods to perform full object rotation state estimation using deep learning have been proven to be difficult [32], we can simplify the problem by only looking at the change in attitude across a time step. Given two images of a satellite as observed by a co-orbital satellite at successive instants of time, we compute the angular velocity of the target satellite. In particular, we employ the simple landmark extraction and tracking approach, aiming to improve upon the performance of conventional algorithms. The model is thus divided into two parts with different purposes: first, landmarks are extracted from the two images and matches between both images are identified; second, the angular velocity and full rotational state are estimated using the matched landmark locations.

For the first part, a convolutional neural network (CNN) is proposed to extract useful landmarks from two images or point clouds, separated by a short time-step; it suffices to look only at high-level features in the form of image landmarks. By making use of the properties of neural networks, we can match landmarks between images implicitly, with no need for complex feature descriptors. In particular, in neural networks the ordering of the output vector is important, a property that can be exploited by proposing that the extracted landmarks from two images passed through the same network can be matched simply by their position in the output vector.

In the second part, the matched landmarks are then used to estimate the rotation of the target over the timestep. The system is encouraged to find landmarks which are inherently useful for the rotation estimation problem, through the use of several loss terms. In order to reject outliers, we also adapt the random sample consensus (RANSAC) algorithm for use within an end-to-end trainable deep learning network.

The remainder of the article is structured as follows: Section 2 discusses the construction of the simulation framework; our proposed approach to divide the model into two parts, namely the landmark detection and matching algorithm, and the rotation estimation algorithm, are discussed in detail in Sections 3 and 4, respectively; the results of the investigations are analysed in Section 5, where our model is compared with conventional techniques; finally, conclusions are summarised in Section 6.

2. Observational Data of in-Orbit Proximity Operations

In order to apply supervised learning techniques to the problem, there is a requirement for a large amount of pose-labelled data. Such a dataset does not currently exist, and would be time-consuming and difficult to generate for real images. Thus, synthetic training data is used as an alternative to real data. This approach enables complete control over all parameters of the data and allows us to generate large datasets quickly.

We have therefore constructed an ad-hoc simulation framework for this purpose. In an active debris removal mission, a debris object (hereafter, the *target*) is approached, observed in close proximity, and eventually captured by a debris removal satellite (hereafter, the *chaser*). During these operations, both satellites are co-orbiting relative to each other, for which the chaser requires of observational data of the target in order to determine its relative position and pose, and characterise its dynamics, geometry and integrity, to feed this information to the GNC subsystem in order to safely perform proximity operations. The aim of this simulation framework is to generate realistic, synthetic optical data from the viewpoint of the chaser. We use Blender[1] for visualisation with python scripts to change the simulation conditions and compute the relative motion of the two satellites.

There are several elements to the simulation: 1) the relative orbital motion between the two satellites must be modelled, as well as their rotational dynamics; 2) the lighting and shadowing conditions must emulate realistic in-flight conditions; and 3) the output video stream should be labelled with the target's pose at each time, and should simulate the output of optical, RGB-D or LiDAR sensors.

2.1. Relative Orbital and Attitude Motion

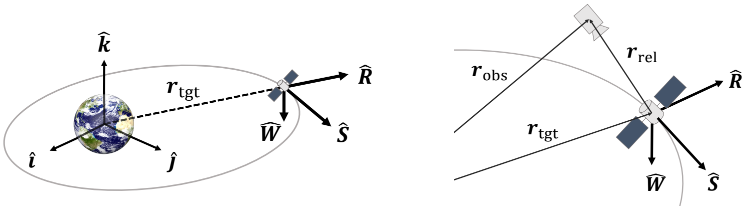
For studying the relative orbital motion, it is customary to use the target-centred, Local-Vertical-Local-Horizontal (LVLH) reference frame, as illustrated in Fig. 1. Following Kaplan's notation [33], the $\hat{\mathbf{R}}$ unit vector is colinear with the position vector, $\hat{\mathbf{W}}$ is normal to the orbit plane, and $\hat{\mathbf{S}}$ completes a right-handed frame, so for a circular orbit $\hat{\mathbf{S}}$ is aligned with the orbital velocity vector. The relative motion of the observing chaser satellite can be described in relation to this new reference frame. The relative position vector of the chaser, $\mathbf{r}_{\text{rel}}$, is given as

$$\mathbf{r}_{\text{rel}} = \mathbf{r}_{\text{obs}} - \mathbf{r}_{\text{tgt}} \tag{1}$$

where $\mathbf{r}_{\text{obs}}$ and $\mathbf{r}_{\text{tgt}}$ are, respectively, the planetocentric position vectors of the chaser (observer) and target satellites.

The equations of relative motion can be linearised which, under different assumptions, yield analytical models of different complexity. The Clohessy-Wiltshire equations [34] are appropriate for modelling close proximity operations

[1] `www.blender.org`



(a) LVLH frame

(b) Relative orbital motion

Figure 1: The target-centred LVLH reference frame.

in a circular orbit, whereas for the case of an elliptic orbit, the Tschauner-Hempel equations [35] are a classical alternative.

In an active debris removal mission, there will often be an observation or monitoring phase in which the chaser moves around the target satellite to characterise its rotational state and other geometric and physical parameters of interest. When the orbital periods of the chaser and target satellites are synced, there are periodic solutions to the relative orbital motion that allow the chaser to circle around the target to perform such activities [36].

The rotational motion of both satellites is also simulated, assuming the target is in a tumbling state, whereas the attitude of the chaser can be controlled. Quaternions are used for attitude parameterisation, which describe the orientation of their body frame coordinates with respect to inertial space [37]. Quaternions are employed due to their built-in redundancy, which not only yields a singularity-free representation, but also conveys error correcting capabilities; additionally, a neural network can easily regress the quaternion directly, as will be discussed later.

For our test case, we consider the chaser to be in the observation phase of a debris removal mission, during which chaser will attempt to keep a circular relative orbit around the target. The target is assumed to be in a circular orbit at an altitude of 500km; thus, the periodic solutions to the Clohessy-Wiltshire solutions result in the monitoring trajectory illustrated in Figure 2. At all points in orbit, we assume that the chaser rotates such that it is always facing towards the target, using its attitude control system. This change in attitude of the chaser over time is considered to be known by the system. On the other hand, the tumbling angular velocity of the target satellite is considered to be constant within its rotating LVLH body frame.

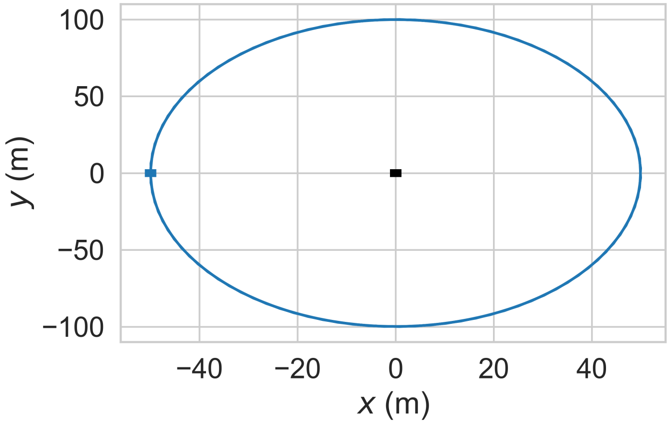


Figure 2: The relative position of the chaser over one orbital period, starting from an initial displacement of $[-50, 0, 0]$ with initial velocity $[0, 0.1109, 0]$, where $0.1109 = -2\omega x_0$

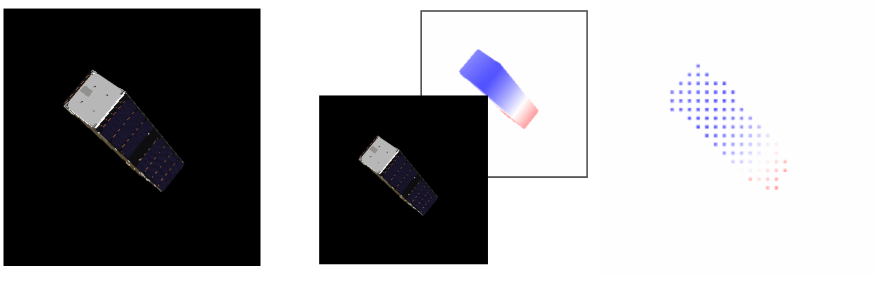
2.2. Lighting and Shadowing Conditions

The unique lighting conditions in space present a key difficulty for optical guidance algorithms, due to the high contrast and reflections, so these must be emulated in the simulation framework. We consider the Sun to be the sole light source, which can be simulated as a distant point source. For simplicity, light reflected from Earth is omitted, and the shadow of the chaser spacecraft upon the target (when both are aligned with the Sun pointing vector) is disregarded.

As is the case in orbit, the direction of the light source (i.e. the sun-pointing vector) changes over the course of the simulation. This is due to the position of the target along its orbit, the direction in which the chaser is facing, and the heliocentric motion of the Earth (Solar ephemerides by Blanco-Muriel et al. [38] are used). We simplify the test cases by selecting inclined orbits, so that spacecraft do not cross the umbra and penumbra regions in the shadow cone of the Earth, and thus both spacecraft are illuminated at all times.

2.3. The Output Video Stream

The output of the simulation not only provides the dynamical states of both spacecraft (and their relative states), but also needs to simulate the outputs of an optical camera, a RGB-D sensor and a LiDAR, all onboard the chaser. Optical and RGB-D sensors both provide a colour image, with the latter also providing the depth at each pixel. A LiDAR sensor also measures the depth of an image, with a lower resolution which depends on the sampling rate of the sensor; this data is generated from the depth image by sampling pixels at a given sampling rate, and returning a point cloud containing sparse 3D information. These forms of sensor data are illustrated in Figure 3.



(a) RGB sensor

(b) RGB-D sensor

(c) LiDAR sensor

Figure 3: Different simulated sensors

The output images must be collected into datasets which can be used in training the neural networks. The datasets consist of a collection of pairs of successive images, with a small time step having passed between the two images. We select the position and attitude parameters of the target and chaser such that there is a large variation in relative angular rate between simulations, from 0.5° per step up to 7°. In order to increase the size and variation of the datasets, we also collect image pairs with a gap of 2, 3 or 4 times the time step, enabling us to investigate the response for up to 28° rotations in a single time-step, although the higher end is likely to be very difficult to solve. We use a dataset of approximately 60,000 image pairs for training the model, varying the debris targets, rotations speeds and illumination conditions. Separate, smaller datasets are created for testing and validation, with different simulation parameters; this data is not seen by the model during training. Each image pair must then be labelled with the rotational state of the target satellite across this time step.

2.4. Using Synthetic Data

Synthetic data presents a number of advantages: 1) it enables the use machine learning in environments where real data would be very time consuming or difficult to process; 2) the size of datasets made up of synthetic data can be much larger than would be achievable with real data, due to the speed of generation; and 3) by simply changing certain parameters of the simulation, it is possible to generate new datasets under different conditions to further test the model. In our simulation framework, all parameters are contained in an editable configuration file, which can be either specified precisely or randomised before generating a dataset of tens of thousands of image pairs, a task which would be unrealistically time-consuming for real data.

However, using synthetic data may lead to difficulties at a later stage, when aiming the model to generalise to real images. The synthetic data will not match exactly to real data – in particular, the camera and sensors will not be as accurate as in the simulation, where the depth of each pixel is known precisely. In addition, it is difficult to assess to which extent the synthetic data matches the real data closely enough so the network trained with synthetic data would perform equally well for in-flight operation use.

In this regard, the capability of the model to generalise to real data could be improved by distorting the data [39], which encourages the model to learn the object features and disregard the inaccuracies in the sensor. In order to determine the generalisability of our model, in a further stage we will be generating realistic experimental data in a lab environment as part of a follow-up development plan.

3. Landmark Detection and Matching

In order to estimate the rotational state of an object, we must track the motion of different areas of the target, which are exhibiting apparent changes due to the object's rotation; from this information an estimate of the rotation can be reconstructed. We therefore divide our approach into two key parts. Firstly, we use a landmark detection and matching algorithm to select specific areas or features on the target and track their motion over time. Then, an estimate of the rotation is determined using the motion of these landmarks. This subdivision of tasks enables the neural network to focus on the specific task of landmark detection, by training this part of the model separately. Finally, we can put both parts together to finetune the response, encouraging the model to find landmarks which are specifically useful for estimating the rotation. This section describes our approach towards the landmark detection and matching and presents the mathematical model of the neural network used.

Often, the landmark detection problem is simplified by fitting the target with retro-reflectors at a known location on the target, which are simple to extract from images. Alternatively, if there exists a-priori knowledge of the object's geometry, features can be predefined based on this and matched with the observed data. Instead, we present an algorithm which requires no knowledge of the debris object and is capable of automatically obtaining a set of matched landmark locations which best define the rotational state of the target, thereby enabling fully autonomous operations.

Our landmark detection network accepts two images and returns K feature maps for each image, each corresponding to a different landmark. A spatial soft-argmax [40] over the feature maps then results in two sets of K three-dimensional points $\{\mathbf{p}_k\}$ and $\{\mathbf{p}'_k\}$; $k = 1, 2, ..., K$. The landmark matching process makes use of a core property of neural networks: the ordering inside the vectors is important. This means that the output at index k in each vector $\mathbf{p}_k$ and $\mathbf{p}'_k$ of the aforementioned sets will describe the same feature in both images. There is therefore no need for complex and costly feature description and matching algorithms; the correspondences are instead determined implicitly from the position in the output vector. An additional advantage of neural networks is their ability to learn features which are inherently useful for the specific problem on which they have been trained.

The network can accept data from optical cameras, RGB-D sensors or a sparse point cloud (such as may be generated from a LIDAR sensor). If a point cloud is provided, it is converted to a dense depth map using the method of Ku et al [41]. Landmarks can be extracted from a three-channel colour image, or directly from a depth map. In the latter case, the depth map is first converted to a three-channel image, with the three channels containing the x, y and z information respectively. Alternatively, the RGB and depth information may be fused together to form a six-channel input image. All types of input data are investigated in the later sections.

3.1. Network Architecture

A convolutional neural network is proposed to detect and match landmarks in the two input images, with the layout shown in Figure 4. This is a siamese network, where each input image passes through the same network to extract image landmarks. The network architecture consists of only two convolutional layers with large kernel sizes and no pooling layers, to generate a set of K feature maps, $\mathbf{M}$. A *shallow* network is proposed, which emphasises features with a low level of abstraction such as corners and edges. In addition, pooling layers lead to a loss of geometrical information, so these should be avoided. The landmark locations are extracted from the feature maps by a spatial soft-argmax layer [40].

The soft-argmax algorithm aims to find the location of the highest peak in the feature map. Unlike a simple argmax, this is fully differentiable which ensures that the network is end-to-end trainable. First, the softmax function is applied to the two-dimensional feature maps to obtain a probability distribution over

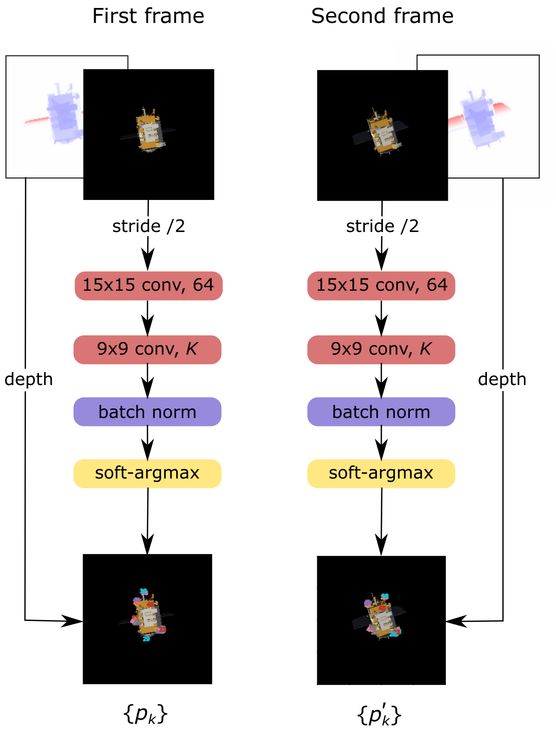


Figure 4: Landmark detection network architecture

the height and width:

$$S_k(u,v) = \frac{\exp(\mathbf{M}_k(u,v)/T)}{\sum_{u=1}^{H}\sum_{v=1}^{W}\exp(\mathbf{M}_k(u,v)/T)} \tag{2}$$

where $\mathbf{M}_k$ is the k-th channel of the feature map, with H and W being the height and width of the map, and $S_k(u,v)$ the value of the probability distribution of feature k at the pixel (u,v). The temperature, T, is a parameter which can be set beforehand or learned by the network. Summing this probability distribution over the height and width provides the (x,y) location of the maximum of each feature map. This is considered to be the position of the landmark at index k, denoted as $\mathbf{p}_k$.

At this point, we have the on-screen (x,y) location of several object landmarks. However, in order to solve for the rotation we instead require the three dimensional (x, y, z) coordinates of each landmark. If RGB-D or LiDAR data is used, the z coordinate is taken directly from the depth map. If depth information is not provided, the network can learn an estimate of the z component, using the same image data as before passed through a few convolutional layers. However, this is not a simple task using the limited information provided, so the accuracy is understandably worse in this case. Once the depth of each landmark has been computed, we must correct for the camera perspective to obtain the 3D point cloud corresponding to each landmark, in the target's body frame; this is trivial given that the camera parameters are known [42, Ch. 6].

The internal architecture of the network, including the number and size of convolution layers, can be easily modified by a configuration file. This enables efficient testing of different architectures. In addition, if the image data used in training is of a sufficiently high resolution, dilation can be used to increase the effective size of the network without obtaining an unreasonably large number of parameters [43].

3.2. Loss Terms

We encourage the network to find landmarks which are inherently useful for the task of rotation estimation, by the choice of several loss terms. To this end, the main loss term is specifically designed to produce landmarks which follow the rotation of the target over the time step. Once a set of learnt landmark locations $\mathbf{p}_k$ are available for a given image, and if the quaternion $\hat{\boldsymbol{\mathfrak{q}}}$ describing the *estimated* or *predicted* rotation between the two considered images is also known, then one would expect that the landmarks in this first image, $\boldsymbol{p}_k$, when rotated by $\hat{\boldsymbol{\mathfrak{q}}}$, would result in the vector of landmarks from the second image, $\mathbf{p}'_k$. Therefore, the loss term L_{rot} is calculated as the difference between this expected result and the observed landmarks in the second image:

$$L_{\text{rot}} = \sum_{k=1}^{K} \left| \boldsymbol{\mathfrak{p}}'_k - \left(\hat{\boldsymbol{\mathfrak{q}}}^* \otimes \boldsymbol{\mathfrak{p}}_k \otimes \hat{\boldsymbol{\mathfrak{q}}} \right) \right|^2 \tag{3}$$

where $\boldsymbol{\mathfrak{p}}_k$ and $\boldsymbol{\mathfrak{p}}_k^{'}$ are the landmark vectors from the first and second image inputs, respectively, expressed in quaternion form, $\hat{\boldsymbol{\mathsf{q}}}^*$ is the conjugate of $\hat{\boldsymbol{\mathsf{q}}}$, $\otimes$ denotes the quaternion product, and the operation $\hat{\boldsymbol{\mathsf{q}}}^* \otimes \boldsymbol{\mathfrak{p}}_k \otimes \hat{\boldsymbol{\mathsf{q}}}$ refers to a rotation of the vector $\mathbf{p}_k$ by the quaternion $\hat{\boldsymbol{\mathsf{q}}}$.

With this loss term alone, there would be a risk that the network could break down by finding a local minimum where all features are at the centre of rotation, in which case L_{rot} would tend to 0. This can be prevented by incorporating an additional separation loss term, L_{sep} [44]. This term encourages the landmarks to spread out over the image by preventing them from clumping together. The separation loss term is given by

$$L_{\text{sep}} = \sum_{k,k'=1;\, k \neq k'}^{K} \exp\left(- \frac{\left|\mathbf{p}_{k'} - \mathbf{p}_k\right|^2}{2\,\sigma_{\text{sep}}^2} \right) \tag{4}$$

where σ_{sep} is a weighting parameter.

We would expect that a higher peak in the feature map will likely correspond to a strong landmark, such as a corner or edge. Therefore, we include a final loss term to look at the concentration of the feature map around the landmark location, termed as L_{conc}. The concentration of each landmark is computed by applying a Gaussian mask centred at the landmark location across the output of the softmax layer. The loss term is then calculated as

$$L_{\text{conc}} = \sum_{k=1}^{K} \exp\left(- \frac{1}{2\,\sigma_{\text{conc}}^2} \sum_{u=1}^{H} \sum_{v=1}^{W} \mathbf{M}_k(u,v)\; G_{(x_k,y_k)}(u,v) \right) \tag{5}$$

where $G_{(x_k,y_k)}$ is a Gaussian mask centered at the landmark location (x_k, y_k) and σ_{conc} is a weighting parameter.

The proposed loss term is the sum of the three terms described above, and therefore presents two adjustable weighting factors that allow for fine control over the output of the landmark detector.

4. Rotation Estimation

Following the detection of image landmarks, the next task is to estimate the rotation of the target from the motion of the landmarks. Given two sets of a minimum of three matched 3D points (i.e. landmarks on two subsequent images that are known to represent the very same feature of the target), an algorithmic method can be devised to determine the best-fit rotation that is compatible with the observed displacement of the landmarks between two successive images. In fact, this information is sufficient to also compute the relative translational motion of the target satellite. However, this has been well covered in past research and we consider it a less challenging problem, which will not provide as valuable an indication of the usefulness of our extracted feature points. We therefore focus our analysis on the sole problem of rotation estimation.

The rotation estimation can be achieved by solving a root-mean-square deviation (RMSD) minimisation problem [45], or alternatively, one can train a fully-connected neural network (FCNN) to estimate the rotation given the 3D points as inputs. Both approaches are implemented in the following, and their performance is analysed and compared.

4.1. Root-Mean-Square Deviation.

This approach provides the rotation which minimises the residual between the observed and predicted landmarks, i.e. the difference between landmark positions observed in an image, and the ones predicted by rotating the landmarks from their observed positions in the preceding image. The residual computation is fully expressed in quaternion notation, so conversions to rotation matrices or other intermediate attitude representations are avoided inside the network. In the quaternion approach [45] the residual E is given by

$$E = \frac{1}{K}\sum_{k=1}^{K}\left(\hat{\mathbf{q}}^{*}\otimes\boldsymbol{\mathfrak{p}}_{k}\otimes\hat{\mathbf{q}}-\boldsymbol{\mathfrak{p}}_{k}^{'}\right)\otimes\left(\hat{\mathbf{q}}^{*}\otimes\boldsymbol{\mathfrak{p}}_{k}\otimes\hat{\mathbf{q}}-\boldsymbol{\mathfrak{p}}_{k}^{'}\right)^{*}. \tag{6}$$

The value of the rotation quaternion which minimises the residual can be found from the correlation matrix, $\mathbf{C}$, between the two sets of image landmarks. This value is equal to the eigenvector corresponding to the maximum eigenvalue of the matrix

$$F = \begin{bmatrix} c_{11} + c_{22} + c_{33} & c_{23} - c_{32} & c_{31} - c_{13} & c_{12} - c_{21} \\ c_{23} - c_{32} & c_{11} - c_{22} - c_{33} & c_{12} + c_{21} & c_{13} + c_{31} \\ c_{31} - c_{13} & c_{12} + c_{21} & -c_{11} + c_{22} - c_{33} & c_{23} + c_{32} \\ c_{12} - c_{21} & c_{13} + c_{31} & c_{23} + c_{32} & -c_{11} - c_{22} + c_{33} \end{bmatrix} \tag{7}$$

where c_{ij} are the components of the correlation matrix $\mathbf{C}$.

4.2. Fully-Connected Neural Network.

As an alternative to the algorithmic RMSD approach, a simple deep learning network can also be proposed to estimate the rotation given the matched landmarks. This method takes the 3D points as inputs which are passed through a FCNN to directly regress the rotation quaternion $\mathbf{q}$ [46].

A simple neural network architecture is investigated, consisting of two hidden layers of 128 units each. The input to the network contains the initial locations of each landmark as well as the change in position over the time step, which is a vector of length $6 \times K$ where K is the number of landmarks. The output is a 4-element vector describing the rotation; a constraint is applied to ensure the quaternion has unit norm. The network is trained to minimise the loss term

$$L_{\text{FCNN}} = \frac{1 - \cos(\theta)}{2} \tag{8}$$

where θ refers to the angular distance between the true quaternion, $\mathbf{q}$, and the predicted quaternion, $\hat{\mathbf{q}}$ [47]:

$$\theta = \arccos(|\mathbf{q} \otimes \hat{\mathbf{q}}|) \tag{9}$$

The FCNN method requires further training on the same datasets. Similarly to the landmark detection network, the data must be labelled with the rotation quaternion between images.

4.3. Relationship Between the Absolute and Relative Rotational Motion

The simulated image pairs are labelled with the true change in attitude of the target satellite. However, the observed change in attitude is a combination of the rotations of both the target and the chaser. Since the chaser is an operational spacecraft, and thus presumably equipped with sufficiently accurate sensors and a fully functional attitude and orbit control system, one can assume that the motion of the chaser satellite is known. Therefore, it is possible to determine the actual rotational state of the target by means of in-orbit observations of its relative motion as observed from the chaser.

Instantaneous rotations[1] between two frames (or 'rotations', for short) are most effectively described in terms of quaternions. Thus, the instantaneous rotation from frame $\mathcal{F}_A$ to frame $\mathcal{F}_B$ is given by the quaternion $\boldsymbol{\mathfrak{q}}_{BA}$. Quaternions have a convenient composition relation that allows to concatenate successive rotations when multiple frames are considered:

$$\boldsymbol{\mathfrak{q}}_{\mathrm{CA}} = \boldsymbol{\mathfrak{q}}_{\mathrm{CB}} \otimes \boldsymbol{\mathfrak{q}}_{\mathrm{BA}} \tag{10}$$

It is important to note that instantaneous rotations and the attitude of an object are deeply related concepts, as the attitude of an object can be described through an instantaneous rotation of its body frame from a given *departure* reference frame; interestingly, note that the departure reference frame can be arbitrarily chosen.

Consequently, in order to describe an object's change in attitude between two successive images (in practice, each taken at successive instants of time), it suffices to find the quaternion that describes the instantaneous rotation exhibited by its body frame from one image to the next. To this ends, body frames must be defined for both, the target and the chaser, and they must be observed at successive instants of time, each of which will provide an image to feed into the landmark detection algorithm.

This is illustrated in Figure 5, which depicts two different instants of time where the attitude of the aforementioned body frames is considered. The frame

[1] The term *instantaneous* is here used to devoid the concept of a rotation from any notion of time dependence, and thus highlight that the considered rotation is simply defined as the difference in attitude between two frames, regardless of kinematic considerations.

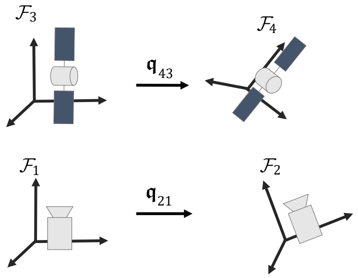


Figure 5: Depiction of the body frames of the target and chaser satellites, before and after a time step δt.

$\mathcal{F}_1$ is defined as a reference frame that is aligned with the chaser's body frame at the initial instant of time, where the first image is obtained; thus, note that in the subsequent rotational motion the chaser's body frame will evolve, whereas $\mathcal{F}_1$ will remain unchanged in the LVLH space. After a time step δt, when the second image is to be obtained, the chaser's body frame would have evolved (following its attitude dynamics) to its current pose; thus, another reference frame can be defined, namely $\mathcal{F}_2$, that is aligned with the chaser's body frame at this instant of time. Therefore, the change of attitude exhibited by the chaser's reference frame in the considered timeframe, is equivalent to the instantaneous rotation from frame $\mathcal{F}_1$ to frame $\mathcal{F}_2$, which can thus be represented by the quaternion $\mathbf{q}_{21}$. Similarly, reference frames $\mathcal{F}_3$ and $\mathcal{F}_4$ can be defined so they match the attitude of the target's body frame at the two considered instants of time, where quaternion $\mathbf{q}_{43}$ establishes the rotation from $\mathcal{F}_3$ to $\mathcal{F}_4$. Finally, in order to describe the attitude of each of these four reference frames, it is useful to define a separate inertial reference frame, $\mathcal{F}_0$, which can be arbitrarily defined and used as a common departure reference frame.

Determining the rotational state of the target requires that the quaternion $\mathbf{q}_{43}$ be known from images at any two successive instants of time. However, since $\mathcal{F}_3$ and $\mathcal{F}_4$ are unknown, the determination of $\mathbf{q}_{43}$ can only be done indirectly. To this end, the only information available in practice is the target's *observed* change in attitude as seen by the chaser, i.e. the relative attitude of the target body frame referred to the chaser body frame. The relative attitude of the target with respect to the chaser is described by quaternions $\mathbf{q}_{31}$ and $\mathbf{q}_{42}$, respectively, for each of the two instants of time considered. The change of the target's attitude relative to the chaser is, precisely, the output that the algorithms presented in Sections 4.1 and 4.2 provide, as well as the same quaternion used in the definition of the loss term L_{rot} presented in Eq. (3). Thus, for the sake of notation consistency, we shall denote this change of the relative attitude by $\hat{\mathbf{q}}$, where no subscript is indicated; from the quaternion composition relation it is straightforward to see that this quaternion is related to $\mathbf{q}_{31}$ and $\mathbf{q}_{42}$ by means of

$$\mathbf{q}_{42} = \hat{\mathbf{q}} \otimes \mathbf{q}_{31}. \tag{11}$$

The rotational state of the chaser satellite with respect to an inertial frame $\mathcal{F}_0$ is assumed to be known with accuracy at all times, thus quaternions $\mathbf{q}_{10}$ and $\mathbf{q}_{20}$ are available, and so is $\mathbf{q}_{21}$. The desired rotation, $\mathbf{q}_{43}$, can thus be written in terms of the known initial state of the chaser body frame, i.e.

$$\mathbf{q}_{43} = \mathbf{q}_{41} \otimes \mathbf{q}_{13} = \mathbf{q}_{41} \otimes \mathbf{q}_{31}^{*} \tag{12}$$

In order to solve for $\mathbf{q}_{43}$, Eq. (12) shows that $\mathbf{q}_{41}$ needs to be expressed in terms of known quantities. Using rotation composition and combining with Eq. (11) yields:

$$\mathbf{q}_{41} = \mathbf{q}_{42} \otimes \mathbf{q}_{21} = \hat{\mathbf{q}} \otimes \mathbf{q}_{31} \otimes \mathbf{q}_{21}, \tag{13}$$

thus Eq. (12) can be rewritten as

$$\mathbf{q}_{43} = \hat{\mathbf{q}} \otimes \mathbf{q}_{31} \otimes \mathbf{q}_{21} \otimes \mathbf{q}_{31}^{*}. \tag{14}$$

Clearly, this expression still contains an unknown quantity, $\mathbf{q}_{31}$, since the true initial state of the target is unknown; finding a work-around requires some additional considerations. At this stage, it must be noted that we have not at any point introduced any assumption nor constraint in the definition of the target body frame, beyond the fact that it needs to be a body frame, i.e. a frame rigidly attached to the target object. Usual choices for a body frame are typically based either on geometric considerations, or on the mass geometry of the object and its principal axes of inertia; however, the orientation of the body frame can actually be arbitrary, as long as it rotates with the body it is attached to. One must also bear in mind that the target object can in principle be of unknown geometry and properties, so for a rotation estimation algorithm to be general, it is actually desirable that it does not depend on a-priori information nor predefined body frames, and thus that it be an automatic, self-starting algorithm that will work on any target object, regardless of its shape or inertia properties.

With these considerations in mind, and since one is free to choose any body frame for the target, a convenient choice is to define a target body frame which is intentionally aligned with the chaser body frame at the very first instant of time. It is easy to see that this choice yields to an initial quaternion $\mathbf{q}_{31}$ associated to an identity rotation matrix, since the body frame of the chaser and the target would both be coincident, i.e. there would be no rotation between them. Hence, Eq. (14) would simply to

$$\mathbf{q}_{43} = \hat{\mathbf{q}} \otimes \mathbf{q}_{21}. \tag{15}$$

Obviously, in subsequent instants of time this would no longer be the case if the target, the chaser or both are rotating, so Eq. (14) would need to be used in all remaining time steps; indeed, as time evolves and the rotation estimation algorithm is thus successively applied at subsequent instants of time, the quaternion $\mathbf{q}_{31}$ for images belonging to successive instants of time would change, but it would now be a known quantity, because it can be computed from the angular velocity by integrating the equations of motion for the relative attitude; indeed, the proposed choice for the target body frame allows to set the initial conditions at the very first instant of time, which allows to start the integration of the equations of motion, which in turn can provide $\mathbf{q}_{31}(t)$ at any subsequent instant of time, and therefore the rotation estimation algorithm would be complete.

Alternatively, note that if one is solely interested in determining the instantaneous rotation between any two subsequent frames (i.e., a quaternion), but not the angular velocity vector, then the arbitrary target body frame could actually be reset or redefined at each time step so it recurrently coincides with the chaser body frame; therefore, in practice Eq. (15) could be used at each time step instead of Eq. (14). Note, however, that with this procedure computing the angular velocity would require extra caution due to the target body frame being re-defined in each time step.

The proposed choice of the target body frame provides a universal algorithm in the sense that it does not require any a-priori or predefined information about the target object; however, there are situations where one needs to specifically select a user-specified target body frame, e.g. based on recognisable features of the target object. For example, this would be the case for a rendezvous and docking manoeuvre, where the location of docking stations, solar arrays and other peripherals of the target are provided in a specific target body frame, and thus it would be required that the target's attitude relative to the chaser be computed using a predefined target body frame. In this case, the presented procedure would still remain valid, with the notable advantage that the attitude of $\mathcal{F}_3$ with respect to $\mathcal{F}_1$ at the initial instant would be provided, and therefore the quaternion $\mathbf{q}_{31}$ is initially known, so Eq. (14) can be used all along without the need to define an ad-hoc arbitrary reference frame.

4.4. Outlier Rejection

Since the landmark matching method used is simplistic in nature, the matches may not always be accurate. For example, one landmark may show the top-right corner of the target; after a rotation, the top-right corner may be a different part of the target which was previously unseen. Therefore, we would like to ignore landmarks which are not useful or are matched poorly.

We achieve this, in part, by assigning a value for the confidence of each pair of detected landmarks. The confidence value is simply a measure of the likelihood that a certain feature corresponds with a strong object landmark. This confidence measure is calculated in a similar way to the loss term L_{conc}, i.e. by applying a Gaussian mask, centred at the landmark location, across the output feature map. This value is used to reject those features with the lowest confidence.

We also implement an outlier rejection algorithm within the end-to-end learnt network. A common choice for outlier rejection is the random sample consensus method, RANSAC [48]. However, this is an iterative algorithm, which makes it unsuitable for use within deep learning networks in its current state. The RANSAC algorithm was therefore adapted to make it parallel and fully differentiable. Figure 6 illustrates an adapted RANSAC algorithm for outlier rejection within a neural network. The steps of the algorithm are as follows:

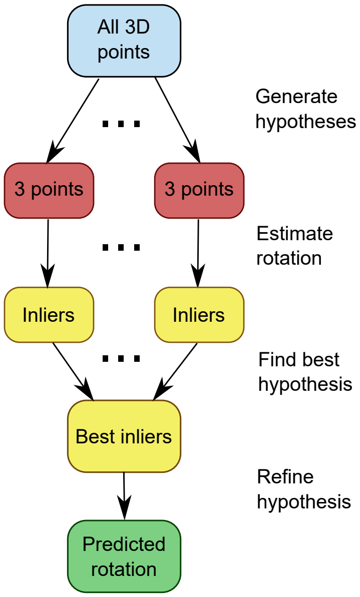


Figure 6: The adapted RANSAC algorithm

1. **Generate hypotheses.** Randomly sample N sets of three landmark pairs from the sets of all landmarks. These form the set of hypotheses H. Each hypothesis H_n contains a set of three points $(\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3)$ randomly sampled from $\{\mathbf{p}_k\}$. Three landmarks are the minimum required to predict a rotation.

2. **Score hypotheses.** For each hypothesis H_n, estimate the rotation using only the sample of three landmarks. Apply this rotation to all $\{\mathbf{p}_k\}$ to obtain $\{\mathbf{p}_k^r\}$. Each landmark pair $(\mathbf{p}_k, \mathbf{p}'_k)$ for which $|\mathbf{p}'_k - \mathbf{p}_k^r|/|\mathbf{p}_k| < \delta$ is an inlier, while all other pairs are rejected as outliers.
3. **Rank hypotheses.** Order hypotheses based on the number of inliers, and select the best one.
4. **Refine hypothesis.** Using all inliers from the best hypothesis, but ignoring outliers, predict the rotation.

Being differentiable and parallelised, this implementation of the RANSAC algorithm may be contained within the end-to-end trainable neural network. This enables the parameters of the algorithm itself, such as the cut-off value δ, to be trained alongside the rest of the network.

4.5. Finetuning

The two sub-networks, for landmark detection and rotation estimation, are trained separately before combining the two together. At this point, we can finetune the entire network with the aim of encouraging it to detect landmarks which are inherently useful to the specific problem at hand.

The finetuning process involves training the entire network against the rotation estimation loss, with a reduced learning rate. We also introduce another loss term for the finetuning, L_{inliers}, which penalises feature sets with small numbers of inliers. This loss term aims to improve the accuracy of the landmark matching between the two images.

5. Performance Analysis on Simulated Test Data

Before looking at the accuracy of the output of the model, it is helpful to visualise each stage of the process. Figure 7 shows the outputs of the inner layers in the network. The convolutional layers are searching for specific features which can be used to describe the rotation. The feature maps are then passed through the spatial soft-argmax to find the landmark points of each feature, which we have overlaid onto the original images. The RANSAC process removes those landmarks which do not appear to match correctly between the two images. Then, only these matched inliers are passed to the rotation estimator.

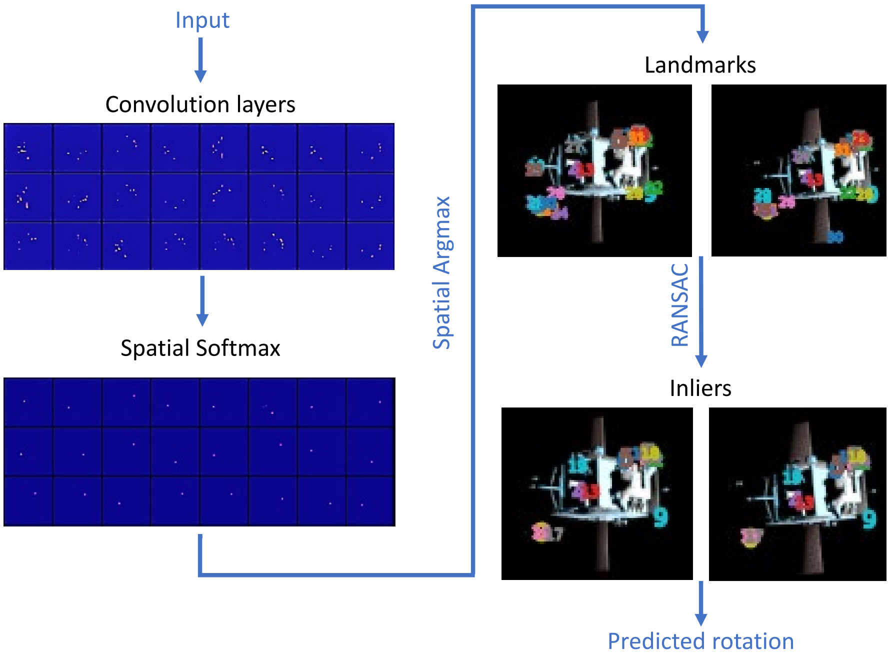


Figure 7: Visualisation of a subset of the outputs of the interior network layers

It can be observed that the the feature maps from the final convolutional layer contain areas of high intensity which correspond with several possible features. The spatial softmax followed by argmax find the landmark locations from each feature map as the point of maximum intensity, where the magnitude of the peak at this point can provide an estimate of the confidence in this landmark. We also see that the RANSAC algorithm successfully removes poorly matched landmarks.

We now look at how the performance of our model changes under different test cases. The use of synthetic test image data facilitates the investigation of our proposed model under different conditions, thus revealing the advantages of it as well as any potential downfalls. We also wish to look at the effects of changing the various parameters of the model itself. However, being a deep-learning based approach, there will always be a significant amount of randomness in the training process. This must be taken into account during numerical analysis.

The loss of the model is defined as the difference in angle between the predicted rotation quaternion and the true value. This angle θ is calculated as in Equation (9), and is equivalent to the angle of the rotation between the final orientation of the target and the predicted orientation following rotation by the estimated quaternion.

We look at four different 3D models for the target satellite, as illustrated in Figure 8. The first three of these (LRO, ICECube and MiRaTa) are real satellite models, taken from NASA's catalogue of 3D models. ICECube and MiRaTa both have simple geometries, but this leads to the downside of having fewer strong features and more rotational ambiguities. The LRO model has a slightly more complex geometry but more strong image features. Finally, the skull model is also included as this represents the most difficult test case, with a complex geometry and few strong features. In each test case, the chaser satellite is considered to be in a circular monitoring trajectory about the target, as described in Section 2.1

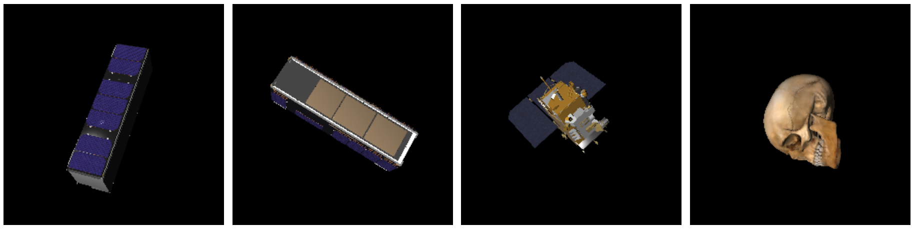
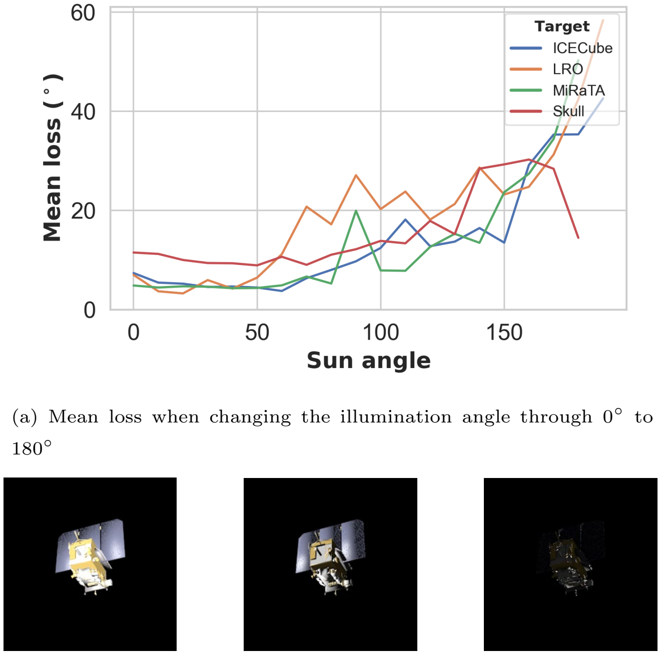


Figure 8: Satellite 3D models. Left to right: ICECube, MiRaTa, LRO, Skull

5.1. Lighting Conditions

We begin the analysis by looking at the response of our model under different lighting conditions, since this is a critical challenge in space missions. Figure 9 contains a comparison of the mean model loss, averaged over the test datasets. In this test case, the illumination direction is rotated between 0° (behind the observer) to 180° (behind the target). The figure also shows the camera view under the change in illumination angle, to illustrate the amount of information present in the image.



(a) Mean loss when changing the illumination angle through 0° to 180°

(b) The view from the chaser satellite where the sun direction is 0°, 90° and 180°, respectively

Figure 9: Model loss response to changing illumination angle, tested on datasets with rotation angles of 20° per step. When the object is poorly illuminated, the images contain less information and the landmark detection is more challenging

In all following analysis, we look at the case where the sun angle is at 90° to the camera view direction, since it is important to maintain good performance under these more challenging illumination conditions.

5.2. Model Accuracy

We now look at the accuracy of the model on a number of test datasets. The rotation axis, lighting conditions and relative position of the satellites are all kept consistent between each dataset. The loss (θ) at each step of an observation period is plotted in Figure 10 against the offset from the initial attitude. Here, the target is the ICECube satellite which is rotating by 20° per step. The Figure is labelled with the view from the chaser at several points during the observation.

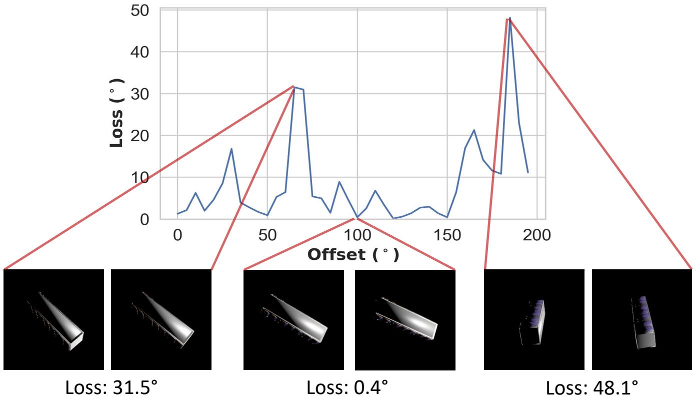


Figure 10: Loss over the observation period against the offset from the initial position, illustrated with examples of the image data at several steps

There is a significant amount of variation in the results over the test dataset. The model is capable of providing accurate estimates where the conditions are good, but is not robust to the more challenging conditions. For example, when we have disappearing edges or significant lighting variations between the two input images, the model can fail resulting in a loss angle which is much larger than the rotation angle of the target over the timestep. These failures have a noticeable impact on the mean loss of the model.

Figure 11 shows the mean losses over entire datasets, between which the angular velocity (and therefore the rotation angle per step) of the target satellite is varied. Since we expect the loss angle to strongly correlate with the rotation angle, we normalise the loss by dividing the former by the latter.

From this Figure, we notice a different response for the different target objects. In all cases, as the rotation angle increases the motion of the landmarks is more pronounced which gives us better information for calculating the rotation. However, on the other hand, as the rotation angle increases we are more likely to

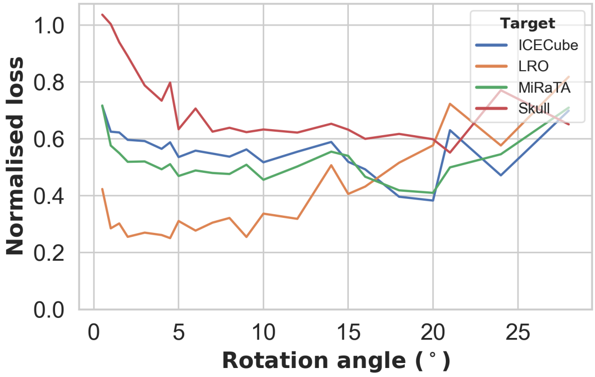


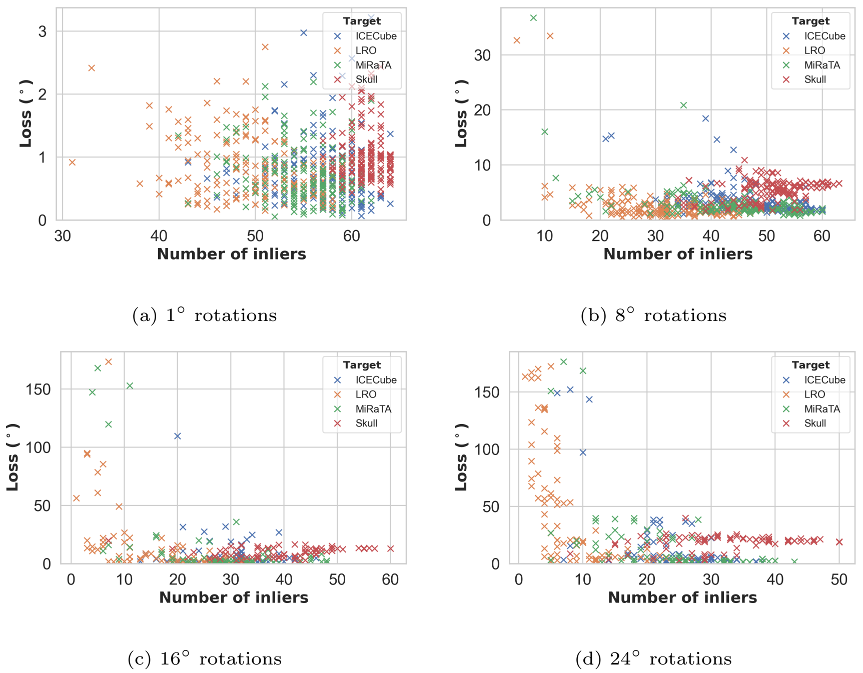
Figure 11: Loss normalised by the rotation angle on all datasets

have self-occlusions, disappearing faces or changes in lighting which the model is not robust to and will lead to failures as addressed previously.

The combination of these two effects depends on the satellite target. For the more detailed LRO target, there is more information in the images to work with but more likelihood of self-occlusions due to the large solar panel, for example. In contrast, the simpler, CubeSat-like, satellites (ICECube and MiRaTa) contain fewer details but are less prone to these issues. Finally, the Skull has a very difficult geometry and few strong features so has lower accuracy throughout.

These observed effects can be seen more clearly in Figure 12. In this figure, the loss angle is plotted against the number of inliers after the RANSAC algorithm at each step. We show the results for four different rotation angles. It is clear from this figure that at very low rotation angles, any small error in the position of landmarks has a large impact on the result so the predictions have a universally high loss. As we increase the angle, the accuracy of the majority of cases increases, but this is counteracted by an increased chance of failure. As has already been noted, the chance of failure is higher on the LRO dataset.

The other result to note here is that the failures occur when the model fails to find a sufficient number of inliers to correctly define the rotation. This suggests



(a) 1° rotations

(b) 8° rotations

(c) 16° rotations

(d) 24° rotations

Figure 12: Step-wise loss against number of inliers. A larger rotation angle results in improved accuracy in the general case but leads to higher likelihoods of a poor match between landmarks

that it is possible to predict a failure by looking at the number of inliers, and rejecting that reading preemptively.

5.3. Network Architecture

As discussed in Section 4, we investigated two approaches for the rotation estimation layer of the model. The root-mean-square deviation method is an algorithmic approach, while the alternative uses a dense fully-connected neural network to make the model completely deep learning-based. Figure 13 analyses the two approaches, labelled as "rmsd" and "dense" respectively.

This Figure shows that the fully deep-learning approach is incapable of learning a useful solution, so it defaults to predicting no rotation. It is possible that the problem is too complicated for a simple neural network architecture. However, we believe that this is more likely caused by the lack of robustness as

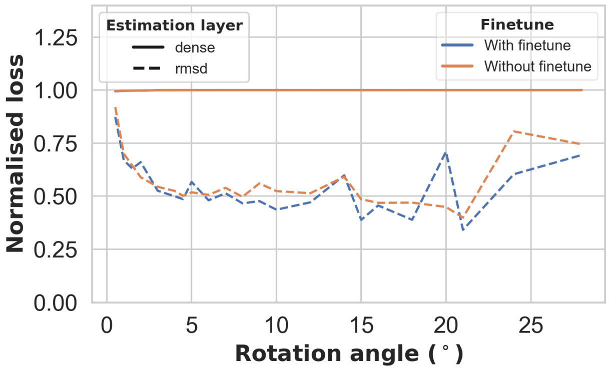


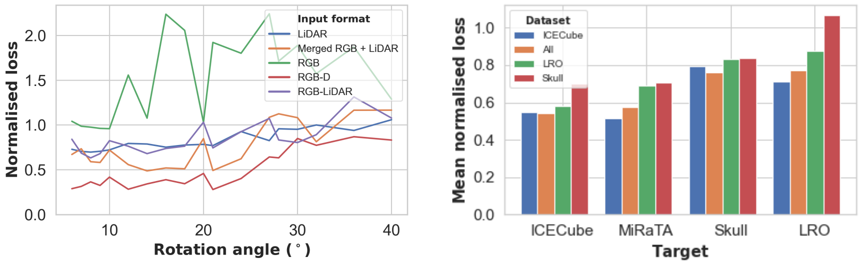
Figure 13: Response of the different estimation layers, on the ICECube dataset

addressed above. When the feature matching layer fails, this often results in large errors, in which cases predicting no rotation is more accurate. This makes training the model more challenging. We also note that finetuning the entire network, as opposed to training the landmark extraction separately to the estimation layer, yields only small improvements, likely for a similar reason.

5.4. Generalisability and Robustness

The same network configuration can be used with various forms of input data. In Figure 14, we investigate the response of the model using data from each of several different visual sensors. These are: a single RGB camera with no depth information; an ideal RGB-D sensor; a LiDAR depth sensor with an angular resolution of one quarter of that of the RGB camera; an RGB camera with LiDAR; and finally, merged RGB and LiDAR data in 6-channel images. In each case, the model is re-trained using data of each type.

Firstly, providing RGB images alone, with no depth information, appears to be insufficient to predict the rotation. An ideal RGB-D sensor performs best, with a slight drop in accuracy when using LiDAR data since there is a loss in information due to the sampling rate. Interestingly, the model with RGB-LiDAR data does not perform noticeably better than with LiDAR data only.



(a) Different input data formats

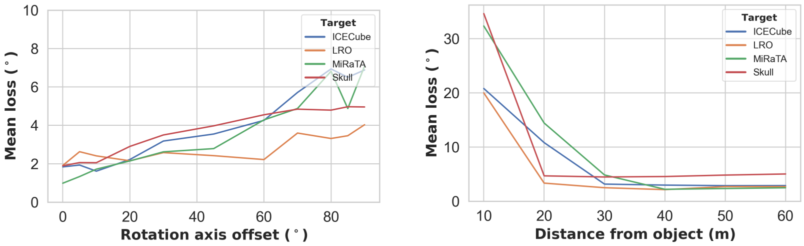
(b) Restricted training datasets

Figure 14: Performance on different datasets, illustrating the possibility of using different sensor types and unseen debris targets

This suggests that either the RGB data is less important than the depth, or the model is not making good use of the RGB data.

The second plot addresses the generalisability of the model to previously unseen targets. Poor generalisability is a known risk when using deep learning approaches; it is important that the model remains accurate when given input data which it has not seen during training. When training on only satellite targets which the model struggles with, we see that our model performs worse on all datasets. On the other hand, if our training data contains targets with strong edges and features from which the network can learn to extract landmarks, we see an improved performance – even on those targets which were not present in the training data. From this, we can conclude that the model has good generalisation properties. This is somewhat expected, since our landmark detection approach looks only for simple image features, due to the low depth of the network.

In the previous analysis, we showed that our model works well under good conditions but is not robust to large changes in lighting conditions or self-occlusions. We further investigate this in Figure 15. In the first of these plots, we look at the axis of rotation of the target in reference to the viewpoint of the chaser. We consider the simplest case to be a rotation about the camera's z axis, in which the motion is contained entirely within the camera's view plane and the depth information is less important. We offset the rotation axis from 0° to 90° from the z axis and observe the response of the loss. The second plot investigates the response to changing the distance between observer and target. Due to the camera's field of view, when the two satellites are within 20 metres, the entire target no longer fits within the image frame. In both cases, we consider a rotation angle of 8° at each step.



(a) Offset of rotation axis from the z axis

(b) Distance of the observer from the target

Figure 15: Analysis of the robustness to different conditions

Again, we observe that our model shows promising results under the simpler conditions, but the robustness requires improvement for the more challenging scenarios. In particular, if the target is not fully contained within the camera frame, this approach does not work. However, the state determination will usually be performed during an observation phase prior to rendezvous, so this should not be an issue.

5.5. Comparison

Finally, we present a comparison of our method with a conventional approach to the same problem, using computer vision techniques for keypoint extraction and matching. Following the keypoint matching, the rest of the algorithm is applied identically to our method. The first method we look at uses ORB features [49] with brute-force matching. The second employs SIFT [50] for keypoint extraction with matching based on the Fast Approximate Nearest Neighbour Search (FLANN) algorithm [51]. Both SIFT and ORB are commonly used in various image matching applications. The brute-force matching approach ensures best matches by looking at all possible matches, whereas FLANN is a more efficient approach but will not always find the best match. In addition, we compare our results with a point cloud based technique, using nearest neighbour matching and the iterative closest point (ICP) algorithm, using 10 iterations per step. The results of the comparison are plotted in Figure 16, on the ICECube test dataset.

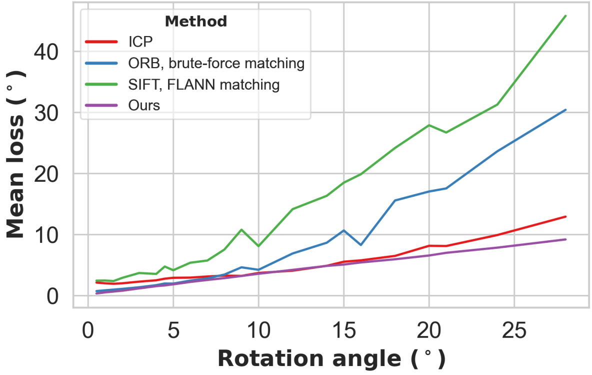


Figure 16: Prediction accuracy compared with conventional computer vision algorithms

Our model appears to have an improved performance in almost all cases when compared with the conventional approaches. The significant improvement over the same approach using conventional feature matching algorithms shows that our neural network-based approach is capable of learning intrinsically more useful image features for this specific problem. The computer vision approaches may have difficulty matching features in this challenging environment, since the matching is based on descriptions of the surrounding pixels and many of the features look very similar. By training on images in this environment, we can learn to better match these features, and to extract features which best describe the rotation. In addition, we have shown that the mean loss for our model is heavily influenced by a few steps with high losses; this is not the case for the conventional approaches, which instead show a more consistent loss which is higher than in our model, in the general case. If these steps can be identified and rejected, by making use of their relationship with the number of inliers, we will expect a further improvement. The fact that our approach achieves similar or better performance than the ICP method, which has been used in real missions, shows that our improved feature matching approach could be a feasible alternative solution to this problem.

Our model has a further advantage over the conventional approaches. While the latter require a greyscale image only, our model can accept any form sensor data, including combinations of depth and visual sensors. The requirement is simply that the model be trained or finetuned using the specific input data.

The timing analysis illustrates an area where our model falls slightly behind the state of the art, in part due to the fact that the RANSAC algorithm is not well optimised for the GPU. We noted a sampling rate of 23Hz for our model, increasing to 30Hz if the RANSAC layer is disabled. In contrast, when using the ORB feature extractor with brute force matching, we can achieve up to 50Hz, though this is also reduced to 25Hz if we include the same RANSAC algorithm. Our implementation of the ICP method is significantly slower, approximately 2-3 Hz – however, it should be noted that the efficiency of the implementation could be much improved, so this is not a fair comparison. Analysis of light curves [52] shows that the minimum observed period of rotation for an object in LEO is approximately 1 second, with most being significantly higher. Even in the most extreme case, a sampling rate of 20Hz would result in a rotation angle of 18° per step, which is within the range of reasonable accuracy for our model. Our model also uses a GPU for the majority of the calculations while the conventional approaches use CPU processing. GPUs offer improved performance and energy efficiency for a lower cost when compared with CPUs, making them an attractive candidate for on-board computation in space [53].

6. Conclusion

In this study, we have presented a deep learning model to determine the change in attitude of an unknown satellite target. The developed model uses machine learning techniques to detect and match landmarks from visual data. Outlier landmarks have been rejected by constructing a fully differentiable and parallelised implementation of the RANSAC algorithm. The model is compared with several conventional approaches, showing similar or better performance throughout. The neural network-based approach to feature extraction is able to learn intrinsically more useful image features than conventional approaches, enabling this method as a potential alternative to the commonly-used iterative closest point algorithm. For simple cases, the model demonstrates a good performance, though it is not yet fully robust to the more challenging conditions which can be present in the space environment. Although the developed model can be prone to large errors where too many landmarks are rejected as outliers, we have shown that these poor estimations are predictable by their relationship with the number of inliers, and so can be rejected preemptively to improve the accuracy. In addition, as the model can generalise well to previously unseen satellite shapes, due to the nature of the feature extraction approach, it does not require information about the target satellite to be known a-priori. We are able to perform calculations at a sufficiently high rate for any observed tumbling rate of debris objects, using GPU instead of CPU processing which is increasingly looking like a more attractive solution for on-board computation.

This work proposes a promising solution to the problem of guidance, navigation and control in active debris removal missions. However, there remain some further avenues to investigate. More work is required to improve the robustness, either by preventing or pre-emptively rejecting those steps with high errors. However, as is always the case in machine learning applications, the robustness can likely be improved simply by increasing the amount and variance of the training data. Also, by looking at only one step we are not making use of the fact that the angular velocity of the target spacecraft should not be changing significantly over small timesteps, and must do so according to its attitude kinematics; thus, by providing rotational information from previous steps, we can increase the amount of information that the model relies on to make a prediction; alternatively, the model could be combined with appropriate filtering techniques, such as a Kalman filter, to improve the accuracy of the attitude reconstruction. Additionally, all analysis of the model is performed using simulated data. In order to determine whether the model will generalise well to real data, we aim to produce more realistic, experimental datasets within a lab, using actual visual and LiDAR sensors.

7. Acknowledgements

This work was supported by the EPSRC Centre for Doctoral Training in Next Generation Computational Modelling Grant No. EP/L015382/1. The authors acknowledge the use of the IRIDIS High Performance Computing Facility, and associated support services at the University of Southampton, in the completion of this work. Hodei Urrutxua also wishes to acknowledge funding from grant ESP2017-87271-P (MINECO/AEI/FEDER, UE), as well as Project Grant F663 - AAGNCS by the “Dirección General de Investigación e Innovación Tecnológica, Consejería de Ciencia, Universidades e Innovación, Comunidad de Madrid” and “Universidad Rey Juan Carlos”.

References

[1] D. J. Kessler, B. G. Cour-Palais, Collision frequency of artificial satellites: The creation of a debris belt, Journal of Geophysical Research: Space Physics 83 (A6) (1978) 2637–2646.

[2] J.-C. Liou, N. L. Johnson, Risks in space from orbiting debris, Science 311 (5759) (2006) 340–341.

[3] R. S. Jakhu, Iridium-Cosmos collision and its implications for space operations, Springer Vienna, Vienna, 2010, pp. 254–275.

[4] C. Pirat, M. Richard-Noca, C. Paccolat, F. Belloni, R. Wiesendanger, D. Courtney, R. Walker, V. Gass, Mission design and gnc for in-orbit demonstration of active debris removal technologies with cubesats, Acta Astronautica 130 (2017) 114 – 127. doi:10.1016/j.actaastro.2016.08.038.

[5] H. Urrutxua, C. Bombardelli, J. M. Hedo, A preliminary design procedure for an ion-beam shepherd mission, Aerospace Science and Technology 88 (2019) 421 – 435. doi:10.1016/j.ast.2019.03.038.

[6] M. M. Castronuovo, Active space debris removal—a preliminary mission analysis and design, Acta Astronautica 69 (9) (2011) 848 – 859. doi:10.1016/j.actaastro.2011.04.017.

[7] H. Hakima, M. R. Emami, Concurrent attitude and orbit control for deorbiter cubesats, Aerospace Science and Technology 97 (2020) 105616. doi:https://doi.org/10.1016/j.ast.2019.105616. URL https://www.sciencedirect.com/science/article/pii/S1270963819321030

[8] X. Wang, L. Shi, J. Katupitiya, A strategy to decelerate and capture a spinning object by a dual-arm space robot, Aerospace Science and Technology 113 (2021) 106682. doi:https://doi.org/10.1016/j.ast.2021.106682. URL https://www.sciencedirect.com/science/article/pii/S1270963821001929

[9] J. L. Forshaw, G. S. Aglietti, N. Navarathinam, H. Kadhem, T. Salmon, A. Pisseloup, E. Joffre, T. Chabot, I. Retat, R. Axthelm, S. Barraclough, A. Ratcliffe, C. Bernal, F. Chaumette, A. Pollini, W. H. Steyn, Removedebris: An in-orbit active debris removal demonstration mission, Acta Astronautica 127 (2016) 448 – 463. doi:10.1016/j.actaastro.2016.06.018.

[10] R. Pinson, R. Howard, A. Heaton, Orbital express advanced video guidance sensor: Ground testing, flight results and comparisons, 2008. doi:10.2514/6.2008-7318.

[11] D. A. Whelan, E. A. Adler, S. B. W. III, G. M. R. Jr., DARPA Orbital Express program: effecting a revolution in space-based systems, in: B. J. Horais, R. J. Twiggs (Eds.), Small Payloads in Space, Vol. 4136, International Society for Optics and Photonics, SPIE, 2000, pp. 48 – 56. doi:10.1117/12.406656.
URL 10.1117/12.406656

[12] N. Rowell, M. N. Dunstan, S. M. Parkes, J. Gil-Fernández, I. Huertas, S. Salehi, Autonomous visual recognition of known surface landmarks for optical navigation around asteroids, The Aeronautical Journal (1968) 119 (1220) (2015) 1193–1222. doi:10.1017/S0001924000011210.

[13] D. S. Lauretta, S. S. Balram-Knutson, E. Beshore, W. V. Boynton, C. Drouet d'Aubigny, D. N. DellaGiustina, H. L. Enos, D. R. Golish, C. W. Hergenrother, E. S. Howell, et al., Osiris-rex: Sample return from asteroid (101955) bennu, Space Science Reviews 212 (1-2) (2017) 925–984. doi:10.1007/s11214-017-0405-1.
URL http://dx.doi.org/10.1007/s11214-017-0405-1

[14] J. Guo, Y. He, X. Qi, G. Wu, Y. Hu, B. Li, J. Zhang, Real-time measurement and estimation of the 3d geometry and motion parameters for spatially unknown moving targets, Aerospace Science and Technology 97 (2020) 105619. doi:https://doi.org/10.1016/j.ast.2019.105619.
URL https://www.sciencedirect.com/science/article/pii/S1270963819323880

[15] V. Pesce, M. F. Haydar, M. Lavagna, M. Lovera, Comparison of filtering techniques for relative attitude estimation of uncooperative space objects, Aerospace Science and Technology 84 (2019) 318–328. doi:https://doi.org/10.1016/j.ast.2018.10.031.
URL https://www.sciencedirect.com/science/article/pii/S127096381830693X

[16] L. Zhang, S. Zhang, H. Yang, H. Cai, S. Qian, Relative attitude and

position estimation for a tumbling spacecraft, Aerospace Science and Technology 42 (2015) 97–105. doi:https://doi.org/10.1016/j.ast.2014.12.025.
URL https://www.sciencedirect.com/science/article/pii/S1270963814002788

[17] F. Cavenago, P. Di Lizia, M. Massari, A. Wittig, On-board spacecraft relative pose estimation with high-order extended kalman filter, Acta Astronautica 158 (2019) 55–67. doi:https://doi.org/10.1016/j.actaastro.2018.11.020.
URL https://www.sciencedirect.com/science/article/pii/S0094576518301516

[18] V. Pesce, S. Silvestrini, M. Lavagna, Radial basis function neural network aided adaptive extended kalman filter for spacecraft relative navigation, Aerospace Science and Technology 96 (2020) 105527. doi:https://doi.org/10.1016/j.ast.2019.105527.
URL https://www.sciencedirect.com/science/article/pii/S1270963818328785

[19] A. Rana, G. Valenzise, F. Dufaux, Evaluation of feature detection in hdr based imaging under changes in illumination conditions, in: 2015 IEEE International Symposium on Multimedia (ISM), 2015, pp. 289–294. doi:10.1109/ISM.2015.58.

[20] J. van de Weijer, T. Gevers, A. D. Bagdanov, Boosting color saliency in image feature detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (1) (2006) 150–156. doi:10.1109/TPAMI.2006.3.

[21] V. Pesce, R. Opromolla, S. Sarno, M. Lavagna, M. Grassi, Autonomous relative navigation around uncooperative spacecraft based on a single camera, Aerospace Science and Technology 84 (2019) 1070–1080. doi:https://doi.org/10.1016/j.ast.2018.11.042.

URL https://www.sciencedirect.com/science/article/pii/S1270963818317346

[22] R. Opromolla, G. Fasano, G. Rufino, M. Grassi, A review of cooperative and uncooperative spacecraft pose determination techniques for close-proximity operations, Progress in Aerospace Sciences 93 (2017) 53–72. doi:https://doi.org/10.1016/j.paerosci.2017.07.001.
URL https://www.sciencedirect.com/science/article/pii/S0376042117300428

[23] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: F. Pereira, C. J. C. Burges, L. Bottou, K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 25, Curran Associates, Inc., 2012, pp. 1097–1105.
URL http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-ne
pdf

[24] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: International Conference on Learning Representations, 2015.

[25] F. Zeng, C. Wang, S. S. Ge, A survey on visual navigation for artificial agents with deep reinforcement learning, IEEE Access 8 (2020) 135426–135442. doi:10.1109/ACCESS.2020.3011438.

[26] S. Milz, G. Arbeiter, C. Witt, B. Abdallah, S. Yogamani, Visual slam for automated driving: Exploring the applications of deep learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2018.

[27] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, A. Farhadi, Target-driven visual navigation in indoor scenes using deep reinforcement learning (2016). arXiv:1609.05143.

[28] S. Sharma, S. D'Amico, Neural network-based pose estimation for noncooperative spacecraft rendezvous, IEEE Transactions on Aerospace and Electronic Systems 56 (2020) 4638–4658.

[29] M. Kisantal, S. Sharma, T. H. Park, D. Izzo, M. Märtens, S. D'Amico, Satellite pose estimation challenge: Dataset, competition design, and results, IEEE Transactions on Aerospace and Electronic Systems 56 (5) (2020) 4083–4098. `doi:10.1109/TAES.2020.2989063`.

[30] S. Sonawani, R. Alimo, R. Detry, D. Jeong, A. Hess, H. B. Amor, Assistive relative pose estimation for on-orbit assembly using convolutional neural networks, arXiv preprint arXiv:2001.10673.

[31] T. Phisannupawong, P. Kamsing, P. Torteeka, S. Channumsin, U. Sawangwit, W. Hematulin, T. Jarawan, T. Somjit, S. Yooyen, D. Delahaye, P. Boonsrimuang, Vision-based spacecraft pose estimation via a deep convolutional neural network for noncooperative docking operations, Aerospace 7 (9). `doi:10.3390/aerospace7090126`.

[32] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, S. Savarese, Densefusion: 6d object pose estimation by iterative dense fusion, 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 3338–3347.

[33] M. H. Kaplan, Modern spacecraft dynamics and control, 1976.

[34] W. H. Clohessy, R. S. Wiltshire, Terminal guidance system for satellite rendezvous, Journal of the Aerospace Sciences 27 (9) (1960) 653–658. `doi:10.2514/8.8704`.

[35] J. Tschauner, P. Hempel, Rendezvous zu einem in elliptischer bahn umlaufenden ziel, Astronautica Acta 11 (2) (1965) 104–+.

[36] D. A. Vallado, W. D. McClain, Fundamentals of Astrodynamics and Applications, 3rd Edition, Vol. 4, Springer, Berlin, Germany, 2007, Ch. 6, pp. 389–416.

[37] L. Markley, J. Crassidis, Fundamentals of Spacecraft Attitude Determination and Control, 2014. doi:10.1007/978-1-4939-0802-8.

[38] M. Blanco-Muriel, D. C. Alarcón-Padilla, T. López-Moratalla, M. Lara-Coira, Computing the solar vector, Solar Energy 70 (5) (2001) 431 – 441. doi:10.1016/S0038-092X(00)00156-0.

[39] M. Sundermeyer, Z.-C. Marton, M. Durner, M. Brucker, R. Triebel, Implicit 3d orientation learning for 6d object detection from rgb images, 2018.

[40] S. Honari, P. Molchanov, S. Tyree, P. Vincent, C. Pal, J. Kautz, Improving landmark localization with semi-supervised learning (2017). arXiv:1709.01591.

[41] J. Ku, A. Harakeh, S. L. Waslander, In defense of classical image processing: Fast depth completion on the cpu, in: 2018 15th Conference on Computer and Robot Vision (CRV), IEEE, 2018, pp. 16–22.

[42] R. Szeliski, Computer Vision: Algorithms and Applications, 1st Edition, Springer-Verlag, Berlin, Heidelberg, 2010.

[43] F. Yu, V. Koltun, Multi-scale context aggregation by dilated convolutions (2015). arXiv:1511.07122.

[44] Y. Zhang, Y. Guo, Y. Jin, Y. Luo, Z. He, H. Lee, Unsupervised discovery of object landmarks as structural representations (2018). arXiv:1804.04412.

[45] E. Coutsias, C. Seok, K. Dill, Using quaternions to calculate rmsd, Journal of computational chemistry 25 (2004) 1849–57. doi:10.1002/jcc.20110.

[46] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–44. doi:10.1038/nature14539.

[47] D. Huynh, Metrics for 3d rotations: Comparison and analysis, Journal of Mathematical Imaging and Vision 35 (2009) 155–164. doi:10.1007/s10851-009-0161-2.

[48] M. A. Fischler, R. C. Bolles, Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, Commun. ACM 24 (6) (1981) 381–395. doi:10.1145/358669.358692.

[49] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, Orb: An efficient alternative to sift or surf, in: 2011 International Conference on Computer Vision, 2011, pp. 2564–2571.

[50] D. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2004) 91–. doi:10.1023/B:VISI.0000029664.99615.94.

[51] M. Muja, D. G. Lowe, Fast approximate nearest neighbors with automatic algorithm configuration, in: In VISAPP International Conference on Computer Vision Theory and Applications, 2009, pp. 331–340.

[52] J. Šilha, J.-N. Pittet, M. Hamara, T. Schildknecht, Apparent rotation properties of space debris extracted from photometric measurements, Advances in Space Research 61 (3) (2018) 844–861. doi:https://doi.org/10.1016/j.asr.2017.10.048.

[53] L. Kosmidis, J. Lachaize, J. Abella, O. Notebaert, F. J. Cazorla, D. Steenari, Gpu4s: Embedded gpus in space, in: 2019 22nd Euromicro Conference on Digital System Design (DSD), 2019, pp. 399–405. doi:10.1109/DSD.2019.00064.