A Monocular Pose Estimation Case Study: The Hayabusa2 Minerva-II2 Deployment

Andrew Price
Tohoku University
Department of Aerospace Engineering
pricea@dc.tohoku.ac.jp

Kazuya Yoshida
Tohoku University
Department of Aerospace Engineering
yoshida.astro@tohoku.ac.jp

Abstract

In an environment of increasing orbital debris and remote operation, visual data acquisition methods are becoming a core competency of the next generation of spacecraft. However, deep space missions often generate limited data and noisy images, necessitating complex data analysis methods. Here, a state-of-the-art convolutional neural network (CNN) pose estimation pipeline is applied to the Hayabusa2 Minerva-II2 rover deployment; a challenging case with noisy images and a symmetric target. To enable training of this CNN, a custom dataset is created. The deployment velocity is estimated as ***0.1908 m/s*** *using a projective geometry approach and* ***0.1934 m/s*** *using a CNN landmark detector approach, as compared to the official JAXA estimation of* ***0.1924 m/s*** *(relative to the spacecraft). Additionally, the attitude estimation results from the real deployment images are shared and the associated tumble estimation is discussed.*

1. Introduction

In the current era, the utility of satellites cannot be overstated; civil welfare, defence agendas, commercial enterprises, and academic pursuits all rely greatly on satellite technologies. Unfortunately, the maintenance of the current level of utilization and the enabling of future innovations are at risk due to the well-admitted problem of increasing orbital debris [10, 14, 28, 30]. Observing the growth in orbital debris described in [30], combined with the recent rapid expansion of the commercial market [29], it is clear that the likelihood of in-orbit collisions will increase dramatically in the coming years.

In the context of this orbital debris problem, guidelines [11], mitigation standards [27], and independent recommendations [25][1] have been written. Additionally, a number of capture demonstration missions have already flown (*e.g.*, ETS-VII [13]) or are scheduled to fly [33]. (*e.g.*, Clearspace-1 mission [16]).

1.1. Satellite Pose Estimation

Regardless of the orbital debris removal mission specifics, the capability to **remotely** obtain target inertial and spatial information remains a core technological competency for all non-cooperative target interactions (*e.g.*, debris capture). Consequently, research into the development of vision-based pose-estimation systems has grown rapidly and drawn heavily upon more mature fields (*e.g.*, automated cars, factory robotics).

One particular pose-estimation benchmark for monocular systems was the Satellite Pose Estimation Challenge [15, 38]. The dataset was produced at Stanford University, using OpenGL synthetic images to supplement real images of the PRISMA mission TANGO spacecraft [34]. The pose estimation challenge resulted in a number of high accuracy pose estimation pipelines [6, 7, 31].

However, deep space missions provide uniquely challenging circumstances. Images are often characterized by high dynamic ranges and non-diffuse lighting, resulting in highly noisy images with minimal surface texture information. Additionally, the challenges of remote operation require a high degree of onboard autonomy and intelligence for performing observation and capture tasks [26].

1.2. Objective

Similar to many institutes, we desire a generic pose estimation algorithm capable of handling varying geometries, varying lighting conditions and that works autonomously. As early work towards this goal, we selected one of the satellite pose estimation challenge pipelines and applied it to a real-world case study: the deployment images of the Minerva-II2 rover (example images shared as figures 1 and 2). The lessons learned provided valuable insight into working with such noisy images.

[1] Murtaza et alia's paper provides an excellent summary of the current state of affairs regarding orbital debris [25].

Although the Minerva-II2 deployment was not an orbital debris case, the circumstances were similar. The images were taken from deep space and depicted a noncooperative free floating object with unknown trajectory and unknown inertial state. The images also contained significant noise artifacts, which are discussed in section 2. Furthermore, the trajectory estimation is of value as it was used to estimate where the rover landed on the Ryugu asteroid [5].

Figure 1. Deploy. im.2

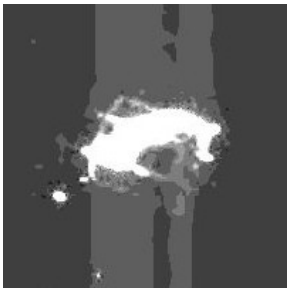

Figure 2. Deploy. im.3

The reader is encouraged to watch the accompanying anonymized paper-summary video. The original contributions of this paper may be summarized as follows:

- The validation of JAXA's rover deployment velocity estimation work [5] using two independent methods.
- The application of a spacecraft pose estimation pipeline to real images (not lab-curated). We provided a roadmap containing our lessons learned while working with the noisy images of a symmetric target (a major challenge for pose estimation pipelines).
- The development and sharing of a new dataset, *Synthetic Minerva-II2* [32], designed to replicate the Minerva-II2 deployment images. The dataset was used for training the pose estimation pipeline. The dataset contains renders of the Minerva-II2 spacecraft with dominant noise artifacts inserted. It is expected that a future general solution to the pose estimation problem will need to be able to work with similarly noisy images.

2. Minerva-II2 Deployment

The MIcro Nano Experimental Robot Vehicle for Asteroid (MINERVA) rovers are small exploration rovers deployed from the Hayabusa2 space probe to explore the surface of the asteroid Ryugu [39]. Unfortunately, the Minerva-II2 rover experienced technical difficulties before deployment and was thus deployed as a visual object to track for microgravity observations. The Minerva-II2 rover was deployed on October 2nd, 2019 [5]. Images of the deployment were captured using the Hayabusa2 Optical Navigation Camera (ONC) Wide Angle number 2. The ONC-W2 camera specifications and calibrated distortion parameters are shared in table 1. The undistorted image, r in pixels from the image center can be expressed as

Parameter	Specification
Sensor Type	CCD
Resolution	1024 x 1024
Sensor Size	$13\mu m$ x $13\mu m$
Focal length	$10.38mm$
FoV	$68.89°$
Distortion ϵ_1	$2.893E^{-7}[pixel^{-2}]$
Distortion ϵ_2	$-1.365E^{-13}[pixel^{-4}]$

Table 1. Hayabusa2 ONC-W2 Camera Specifications [37]

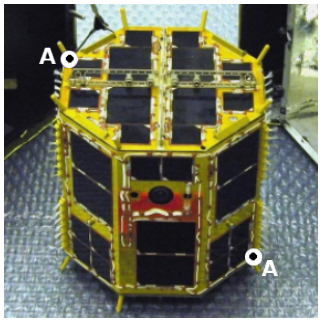

Figure 3. Minerva-II2 rover. Courtesy of JAXA [5]

$$r = r_{distorted} + \epsilon_1 r^3 + \epsilon_2 r^5 \quad (1)$$

to correct for distortion. The ONC-W2 camera captured deployment images every 3 seconds (FPS of $\frac{1}{3}$).

Two example images of the Minerva-II2 deployment are shared as figures 1 and 2. For context, an image of the Minerva-II2 rover is shared in figure 3 in standard diffuse lighting conditions. The Minerva-II2 is an octagonal prism with six of the eight rectangular faces nearly identical; the top and the bottom faces nearly identical; and the remaining 2 rectangular faces —containing the Minerva-II2 cameras— are inverted, but quite similar as well. From the perspective of attitude estimation, the Minerva-II2 presents an extremely challenging target for a monocular system.

The Minerva-II2 surface textures are primarily composed of solar cells, antenna structure, cables, and a reflective yellow surface coating. As demonstrated in deployment image 3 (figure 2), the highly variable reflectivity resulted in high dynamic range pixel intensities and consequently overloaded the image sensor. Reproducing these image artifacts for the synthetic dataset is discussed in section 3. The specific spectral properties of the surface materials, such as reflectance and emissivity, were unknown to us.

The Minerva-II2 deployment mechanism design resulted in a large release velocity uncertainity of 0.054 to 0.254 m/s. Consequently, Oki *et al.* performed a number of

Monte Carlo simulations to select the optimal release attitude for the Hayabusa2 [5]. After release, JAXA maneuvered Hayabusa2 to observe several orbits of the Minerva-II2 rover about Ryugu, using the ONC-T (telescopic) camera. Oki *et al*. then back calculated the Minerva-II2 orbit propagation to estimate the initial deployment velocity and shared the magnitude as 0.1924 m/s relative to the Hayabusa2 (camera fixed frame) [5].

2.1. Projective Geometry Velocity Estimation

Since the geometry of the Minerva-II2 is known, it was possible to estimate the initial release velocity of the Minerva-II2 rover using the ONC-W2 deployment photos (two of which are shared as figures 1 and 2), using projective geometry methods.

The estimation proceeded as:

(1) **Selected two vertices.**

(I) **Method A: Hand selected two vertices.**

(II) **Method B: Assumed a model of a sphere with low oblateness.** As the Minerva-II2's geometry may be described as a regular octangonal prism, two vertices opposite on each $[X, Y, Z]_{bodyFrame}$ will be the same distance apart as two other vertices opposite on $[X, Y, Z]_{bodyFrame}$. For clarity, two such opposite vertices have been labeled as "A" in figure 3. Vertice selection was automated to select the two pixels furthest apart on the "sphere", resulting in the two opposite $[X, Y, Z]_{bodyFrame}$ vertices. Pixel intensity was incorporated into the distance calculation. Outliers were identified manually.

(2) **Projected the model geometry.** Once two vertices of a known distance apart were selected, the distance was compared to the image's projected distance. As the attitude of the Minerva-II2 was unknown for each image, reprojection accuracy was limited.

(3) **Performed least-squares fitting** of the linear velocity across the various deployment images. The least-squares fit was to the linear polynomial

$$\hat{\vec{X}}_k = \vec{X}_0 + \vec{\alpha} t_k \tag{2}$$

where $\hat{\vec{X}}_k = \left[\hat{X}_k, \hat{Y}_k, \hat{Z}_k\right]^T$ is the projective geometry measured target location for image k;
$\vec{X}_0 = \left[X_0, Y_0, Z_0\right]^T$ is the target location at 0 s;
$\vec{\alpha} = \left[V_X, V_Y, V_Z\right]^T$ is the parameter to be fitted;
and t_k is the time at which image k was recorded.
Bisquare weighting on the residuals and the MATLAB fit function [22] were used.

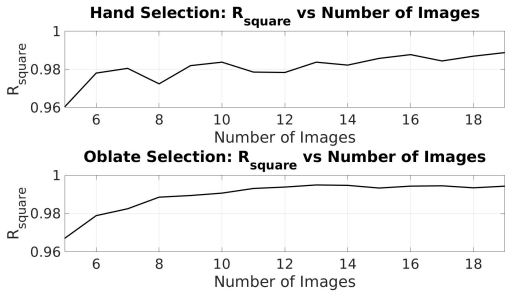

Figure 4. Least-squares R^2 progression. (higher is better)

Parameter	Hand Selection	Oblate Selection
Velocity $\frac{mm}{s}$	$[6.7, 19.8, 188.8]$	$[6.8, 19.8, 189.7]$
Velocity Norm $\frac{m}{s}$	0.1900	0.1908
R^2 (↑ better)	0.9888	0.9943
RMSE (↓ better)	0.3385	0.2477

Table 2. Projective geometry velocity estimation results

Originally, it was assumed that the earlier deployment images (target closer to the camera) would produce higher quality vertice selections and thus data from earlier images should be weighed higher. However, the Coefficient of Determination (R^2) improved with the inclusion of later images (target further from the camera) for both methods. Consequently, a total of 19 images were utilized in the least-squares fit. R^2 versus the number-of-images-used can be observed in figure 4.

The results of the projective geometry approach have been shared in table 2. The low-oblateness model selection exhibited better fit statistics and resulted in a slightly closer estimation to the JAXA velocity norm of 0.1924 m/s.

3. Synthetic Minerva-II2 Dataset

A dataset was required in order to train a pose estimation neural network. Work began by constructing a 3D model of the Minerva-II2 rover in Solidworks. As the Minerva-II2 photos were primarily dominated by image noise artifacts, it was deemed sufficient to use the Solidworks **Photoview 360** ray-tracing rendering software [9] for the intitial render. The dataset was focused on developing a realistic noise reproduction function for the post-**Photoview 360** render.

3.1. Photoview 360 Rendering

Photoview 360 contains a library of different surface materials including fabrics, glasses, metals and ceramics. Appropriate surface materials and their associated spectral properties were selected for each Minerva-II2 surface. Additionally, the Hayabusa2 ONC-W2 camera parameters, shared in table 1, were utilized as the perspective rendering parameters.

To simulate lighting conditions, the sun-camera angle vector was required. Without specifics of the Hayabusa2 compensation maneouver during release [5] and without the relative velocity with respect to Ryugu, it was not possible to estimate the Hayabusa2 attitude with respect to the sun. However, inspired by the light source detection work of Lopez-Moreno *et al*. [18], the sunlight angle was visually estimated and then verified in simulated renderings compared to the real images. A light source angle range of +/- 6° was applied resulting in various light source renderings on the range of approximately $[58, 70]^\circ$ longitude and $[6, 18]^\circ$ latitude. Other potential light sources (*e.g*., surface reflections) were assumed negligable and not simulated.

As the rover's attitude and tumble progression were unknown, it became necessary to render a sampling of all possible orientations. Selecting an orientation is akin to selecting a point on the surface of a unit sphere. In his thesis [34], Sharma describes the optimal orientation rendering sampling solution as "...solving for a minimum-energy configuration for charged particles on a sphere..." also known as the "Thomson problem" aiming to find the minimum energy, E, described as

$$E = \sum_{i=1}^{n-2} \sum_{j=1+1}^{n} \frac{1}{|s_j - s_i|} \tag{3}$$

for each particle (*i.e*., orientation vector) separation $|s_j - s_i|$. In this way, should 100 images with equally spaced orientations be desired, minimizing equation 3 with $n = 100$ will ensure a uniform spread of orientations. Here, the approximate solution developed by Markus Deserno [4] is adopted; please refer to the reference for the algorithm.

Finally, the target distances were sampled from the range $[0.5, 2.5]\ m$ based on the projective geometry velocity estimation and the ONC-W2 FPS of $\frac{1}{3}$, discussed in section 2.

Using the above defined configuration, three datasets were rendered:

- **SetA:** 10,000 renderings of the realistic Minerva-II2 model. This dataset was used in tandem with the real Minerva-II2 deployment images (section 5 case 2).
- **SetB:** 10,000 renderings of a fictitious Minerva-II2 model. The fictitious model was created by strategically removing solar panels to ensure all 10 faces were uniquely identifiable. This dataset was used in tandem with the **Tumble** dataset as part of the proof-of-concept case (section 5 case 1).
- **Tumble:** 300 renderings of the fictitious Minerva-II2 model with the attitudes defined by integrating Euler's rigid body rotation equations. The integration was completed in MATLAB using a self-coded "Runge-Kutta 4$^{\text{th}}$ order" integrator. Based on the Minerva-II2 mass, an arbitrary axi-symmetric inertia matrix of

$$I = \begin{bmatrix} 0.002785 & 0 & 0 \\ 0 & 0.002785 & 0 \\ 0 & 0 & 0.002533 \end{bmatrix} kg \cdot m^2 \tag{4}$$

was utilized for the tumbling integration. An arbitrary initial rotation speed of $\vec{w} = \left[0.2, -0.1, 0.4\right]^T \frac{rad}{s}$ was also used. The integration utilized timesteps of 0.001 s and the system energy was monitored as constant to 12 decimal places, thus ensuring the integration validity. Images were rendered at 1 s intervals.

We share two example renderings as figures 5 and 6.

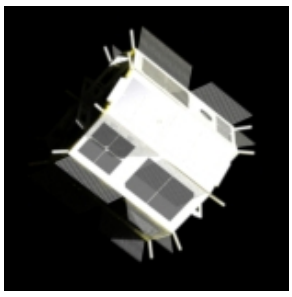

Figure 5. SetA render

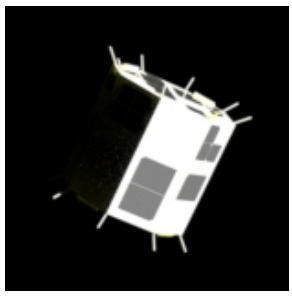

Figure 6. SetB render

3.2. Dataset Noise Filtering

The output of the Photoview 360 ray tracing render software resulted in perspective realistic renderings with correct attitudes. However, the renderings did not exhibit the various sources of noise and image artifacts seen in figures 1 and 2. Consequently, post-processing filtering was completed in MATLAB. Our custom filter function contained a number of steps and will be itemized for brevity. Note that here pixel intensities are referred as 0 for black and 1 for white. MATLAB functions are identified where applicable.

(1) **RGB to Grayscale:** The ONC-W2 camera is a CCD intensity camera. The conversion coefficients were 0.2989Red + 0.5870Green + 0.1140Blue.

(2) **Contrast Adjustment:** To assist in the replication of the high dynamic range scenario, render image intensities < 0.4 were set to 0.

(3) **Gaussian Blur:** To replicate a slight defocus, 2D Gaussian filtering was used; the Gaussian distribution (σ) and filtersize were scaled proportional to target distance. The MATLAB function imgaussfilt() was used.

(4) **Motion Blur:** Motion blurring was applied along a random vector using the MATLAB fspecial('motion') filter kernel.

(5) **Artificial Bloom:** CCD pixels are often arrayed sequentially with anti-blooming drains at the end of a row or column. When the charge capacity of a pixel cell is saturated, it may overflow into adjacent cells [1]. Based on our observations, the ONC-W2 pixels are vertically sequential and thus 1D column Gaussian blurring was applied to intensities > 0.67. σ and filter-size were scaled proportionally to target distance.

(6) **Random Artifact:** Although not readily observed in figures 1 and 2, polygon image artifacts of intensity $\sim$1 dominated several deployment images. Thus a function was coded to produce a random polygon of 10-20 vertices, scaled based on target distance. The function was passed 5 times over an image with a probability of 0.5 to add a polygon.

(7) **Artificial Streaking:** Streaking is another potential artifact caused by CCD pixel saturation [1]. 20% of the time, a vertical line of intensity 0.4 was added to the image at an arbitrary column that intersected with the rover. A form of streaking is observable in figure 2.

(8) **Random Particles:** Also observable in figure 2, are spherical artifacts. The artifacts may be background stars (depending on ONC-W2 automatic contrast adjustment) or particles released during the Minerva-II2 deployment. In anycase, for 50% of the images, $rand[5-50]$ particles were added at random locations.

(9) **Poly Bloom:** Observable in figure 1, some high intensity locations developed sharp looking edges as a saturation effect. This function replicated such behaviour.

(10) **Sensor Saturation:** Observable in figure 2, ONC-W2 automatic contrast adjustment sometimes resulted in higher intensity backgrounds. This function globally increased the image intensity by $+rand[0, 0.1]$.

(11) **Speckle Noise:** The Photoview 360 renders resulted in homogeneous backgrounds (intensity precisely = 0) whereas the deployment images contained slight inconsistencies. The MATLAB function imnoise(image, 'speckle', 0.001) was used here.

(12) **Intensity Reduction:** When the ONC-W2 camera was overloaded, the camera dynamic range became limtied. For 40% of the images, a similar dynamic range reduction was applied.

Continuing our rendering example, post-processing modified outputs are shared as figures 7 and 8.

Although significant effort was expended to reproduce the noise artifacts of the real deployment images, this dataset still posed a large domain adaptation challenge; our synthetic dataset did not have comparable surface texture fidelity as that of Stanford University's SPEED dataset

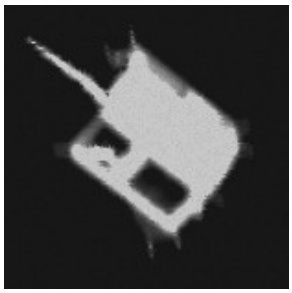

Figure 7. SetA mod.

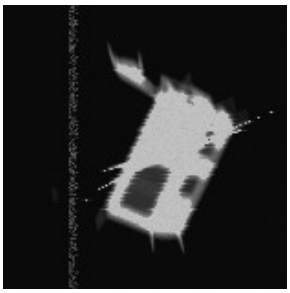

Figure 8. SetB mod.

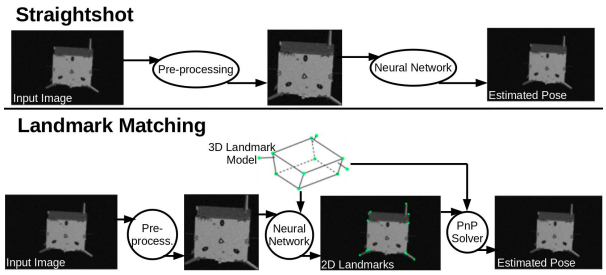

Figure 9. Pose estimation pipelines. Adapted from [15]

[34, 38]. However, by focusing on replicating the dominant noise artifacts, we were successful in developing a dataset suitable for training a neural network to work with the real deployment images. Similarly, for future space mission applications, it is worth considering and focusing on which aspects of an image are the most dominant (*e.g.*, texture, geometry, noise, lighting).

4. Satellite Pose Estimation Methodology

Most Convolutional Neural Network (CNN) architectures can be roughly grouped into two categories: straightshot and landmark matching as shown in figure 9. The primary difference is that the landmark matching approach explicitly utilizes knowledge of the target's geometry.

While there are merits to both approaches, the landmark matching approach exhibited higher accuracy for pose estimation in the satellite pose estimation challenge benchmark [15]. With the Minerva-II2 geometry available, we adopted a landmark matching approach here. Additionally, as we did not have hardware or real-time operation constraints, a larger CNN pipeline could be used. Consequently, we adopted the satellite pose estimation pipeline developed by Dr. Chen [7], which uses the HRNet backbone [36].

4.1. Landmark Regression CNNs

HRNet was initially developed for the Microsoft COCO dataset [17]; both top-down (object $\rightarrow$ landmarks) and bottom-up (landmark $\rightarrow$ object grouping) versions exist [2]. For our case, a bottom-up network could be selected with

the last fully connected layers removed to maintain the 2D landmark feature maps.

A common characteristic of CNN-architectures used in feature point detection is the reduction, and later recovery, of feature map **resolution** (variants of ResNet [12] and feature pyramids [40]). The HRNet team have continually demonstrated the value in maintaining higher resolution feature maps throughout the CNN [8].

One common domain adaptation approach is to train the network on a large general dataset and then freeze the early layer weights [23]. Next, the final layers are trained on the target-specific dataset. We initially experimented by importing weights from HRNet trained on the COCO dataset [2], but initial findings indicated that training from scratch yielded similar results.

5. Pose Estimation Case Studies

We applied our pose estimation pipeline to two case studies: 1) a fictitious Minerva-II2 model with a simulated tumble velocity and 2) the real Minerva-II2 deployment photos. The first case was completed as a proof-of-concept.

5.1. Case 1: Tumble Simulation

We trained on the **SetB** fictitious model dataset and then tested on the **Tumble** dataset. It may be noted that these datasets contained a non-symmetric target and thus the landmark estimator could preserve correspondences.

5.1.1 Case 1: Training

We trained for 200 epochs from scratch with the Adam optimizer. The output of our model was a tensor of 16 heatmaps, one for each annotated landmark. We reduced the heatmap resolution to 512 x 512 (previously 768 x 768 [7]) and trained the model minimizing the **sum** squared error loss as

$$L(x, y) = \sum_{i=1}^{N} (h_i - h_i^*)^2 \tag{5}$$

between the predicted heatmaps h_i and the ground truth heatmaps h_i^*. The ground truth heatmap, h_i^* was generated as a Gaussian normal with a standard deviation of σ. We found that varying the standard deviation throughout the learning process first helped the network identify the landmarks and later improved the accuracy. A σ of 10 at epoch 1 was adopted and decreased piecewise to a σ of 2.

5.1.2 Case 1: Pose Estimation

The initial pose was first estimated using the MATLAB P3P function $estimateWorldCameraPose()$ [21]. Next, the MATLAB $bundleAdjustment()$ function [20] was run.

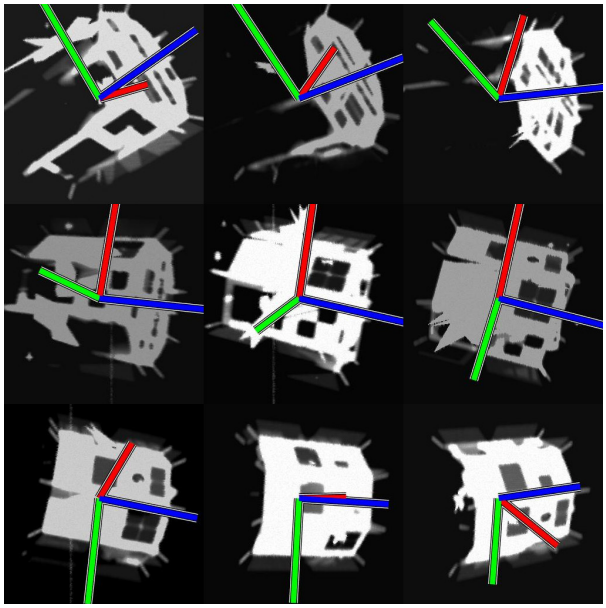

Figure 10. Tumble dataset pose estimation

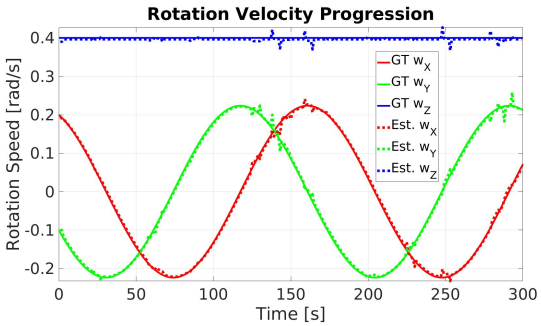

Figure 11. Body rotation vector progression. Ground truth (GT) compared to estimate (Est.)

With the estimated poses (examples shared in figure 10), the body rotation vector could be calculated. The quaternion differential equation was numerically differentiated using the MATLAB $angvel()$ function [19]; the results are shared in figure 11.

Thus we confirmed the pose estimation pipeline methodology was valid for estimating the tumble velocity of a non-symmetric target.

5.2. Case 2: Real Deployment Images

For the real images, we trained on the **SetA** dataset, we then tested on JAXA's real Minerva-II2 deployment images.

5.2.1 Case 2: Addressing the Minerva-II2 Symmetry

Due to the symmetry of the Minerva-II2 rover, HRNet was unable to identify independent correspondences. Conse-

quently, the unorthodox step of dropping the correspondence identification was adopted. Instead of outputing 16 heatmaps (one for each correspondence landmark), a single heatmap was output. The ground truth heatmap was created by superimposing 16 heatmaps (a gaussian normal with standard deviation σ for each correspondence landmark).

As an additional post-processing step, a $findpeaks$ function identified the highest scoring local peaks to assign as landmark locations. A minimum distance of $\frac{\sigma}{2}$ was enforced between peaks to ensure a single global maximum landmark was not assigned multiple times. It is worth noting that at great distances, 2D landmarks will be closely spaced; thus the performance of this $findpeaks$ function dropped off significantly after 10 metres.

5.2.2 Case 2: Training

The network performance on the target (real) dataset decreased for extended training sessions (the network was overfitting to the training dataset). We thus trained 8 models for different durations to compare ($[1, 3, 5, 10, 15, 25, 50]_{epochs}$). Other training parameters were kept the same as for case 1.

With additional training, a tradeoff was observed. Low epoch models provided more landmark estimations while high epoch models provided fewer outliers. Outliers were identified by network confidence and by recursive error reduction of subsequent pose estimation functions (*e.g.*, P3P). An example of the number of identified points for each model in the first 20 images during a training cycle is provided in figure 12.

It appeared that the low epoch models relied more on high contrast and obtuse angle geometry, while the high epoch models began to rely on more localized features. An example (deployment image 7) of these different landmark detections is shared as figure 13 for model 1 (1 epoch) and as figure 14 for model 8 (50 epochs). Model 1 provided 11 landmarks, 2 of which appeared to be outliers. Model 8 provided 5 landmarks, none of which appeared to be outliers. Additionally note that neither model provided landmark locations of nonvisible vertices. This is possibly due to the $findpeaks$ function step. The $findpeaks$ function prioritized higher weighted local peaks (peaks with higher confidence).

5.2.3 Case 2: Pose Estimation Results

CNN output/landmark correspondences were first initiated manually and initial poses were validated with iterative uses of $estimateWorldCameraPose()$ [21]; with more images, perhaps a RANSAC algorithm would be a robust way to autonomously reassign correspondences (we worked with only 10-12 images). In this way, the pose attitude was restricted to 16 possible orientations (a regular octogonal

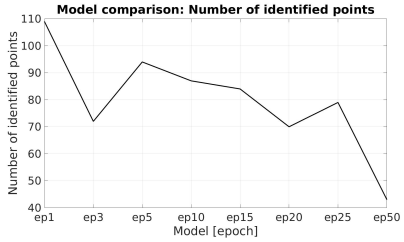

Figure 12. Number of identified landmarks vs training duration

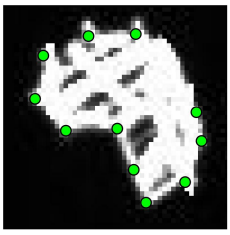

Figure 13. Real image 7; model 1 landmarks

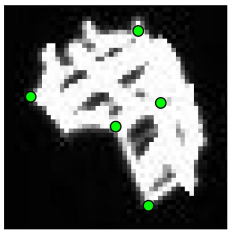

Figure 14. Real image 7; model 8 landmarks

Parameter	CNN Landmark	Oblate Selection
Velocity $\frac{mm}{s}$	$[4.0, 30.5, 191.0]$	$[4.2, 19.7, 189.7]$
Velocity Norm $\frac{m}{s}$	0.1934	0.1908
R^2 ($\uparrow$ better)	0.9813	0.9943
RMSE ($\downarrow$ better)	0.3333	0.2477

Table 3. Velocity estimation results

prism can be oriented 16 ways to produce the same image projection).

As the number of images with estimated poses was rather limited (10-12 images), the $bundleAdjustment()$ [20] Jacobian to be optimized was rather sparse. Additional poses from the SetA estimatitons were incorporated into the bundle adjustment to improve the algorithm's repeatability.

For the linear velocity estimation, similar to the projective geometry method (equation 2), a least-squares approach was adopted [22]. Conversely to that of the projective geometry method, the velocity estimation confidence (R^2) decreased with the inclusion of later images (the network performance decreased as the target distance increased).

The velocity estimation results are shared in table 3. The low-oblateness projective geometry velocity estimation is included for comparison. Again, the JAXA velocity norm estimation was 0.1924 m/s.

Finally, we discuss the attitude estimation. Again, due to the dropped and reassigned correspondences, 16 possible attitudes exist for each image. However, we were able to estimate the axis of symmetry and the associated 16 possible attitudes in the majority of the early deployment images. Example results are shared in figure 15.

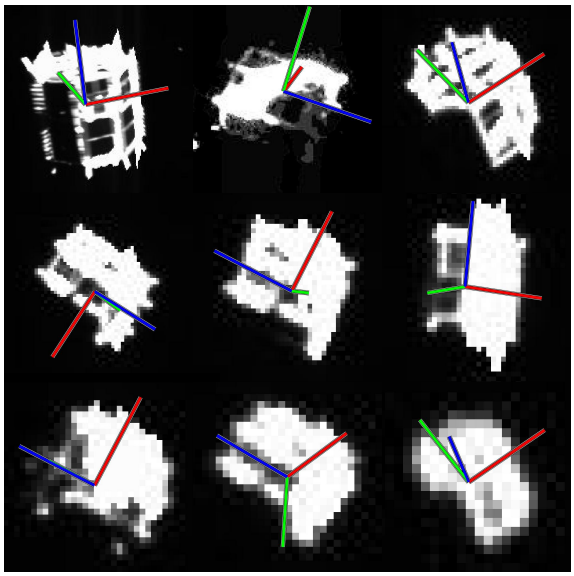

Figure 15. Attitude estimation results

There were insufficient images to constrain $[w_X, w_Y, w_Z]$ and the uknown 9 parameter inertia matrix; Euler's equations of motion could not be estimated using the previous least-squares approach or MATLAB $angvel()$. We are currently investigating an iterative harmonic approach wherein we integrate Euler's equations of motion with different parameters to find a best fit.

6. Discussion

Solving for the target tumbling motion (pose derivative) begins to encroach upon the solution space of Simultaneous Localization And Mapping (SLAM) methods. A monocular ORB-SLAM approach [3, 24] was intitially investigated but quickly discarded. The keypoint detection functions were not sufficient to reliably locate landmarks on the noisy Minerva-II2 images.

Many SLAM algorithms benefit from high framerates for loop-closure queries [24] and/or extrapolate between timesteps using physics based algorithms [35]. However, many spacecraft pose estimation applications (such as the Minerva-II2) are characterized by low framerates and thus information must be supplemented using the physics-based solutions (*e.g.*, an Extended Kalman Filter (EKF) tuned to Euler's rigid body equations of motion).

It is also interesting to note the performance of the HR-Net on landmark detection in section 5.2.2; the HRNet delivered workable results with minimal training. Consequently, as a follow up to this work, we intend to focus on the development of a smaller network suitable for spacecraft hardware. Implementing a similar performing algorithm on spacecraft sized hardware is a non-trivial task. However, with the future generation spacecraft's increased degree of required autonomy [26], we expect smaller network performance to be of immense value.

From this work, we expect valuable next steps for the general spacecraft pose estimation system will be to 1) Perform a similar case study with a much smaller CNN backbone; 2) Assess methods of miniaturizing networks for real-time hardware; 3) Incorporate an overhead pose tracking algorithm (*e.g.* EKF); and 4) Expand the study to more spacecraft, more geometries and more types of image noise.

An additional challenge will be how to autonomously select landmarks. Many texture based keypoint detectors were not designed for the saturated dynamic ranges and noise associated with deep-space imagery.

7. Conclusion

In this paper we investigated the deployment of the Minerva-II2 rover from the Hayabusa2 space probe as a precurser study to the development of a generic satellite pose estimation system. We developed and shared a new dataset, *Synthetic Minerva-II2* [32]. The dataset was specifically crafted to emulate the noise artifacts of the real Minerva-II2 deployment photos, which are representative of real challenges for deep space missions. It is argued here that similar noise artifacts should be included in the development of future pose estimation pipelines or we risk such algorithms only being usable in ideal conditions.

We presented a simple projective geometry approach to estimate the position of the rover then moved onto a higher complexity CNN solution to estimate both the position and attitude of the rover. The velocity was estimated in both cases and validated the independent JAXA orbital back-propagation estimation. It is clear that monocular pose estimation accuracy has advanced significantly and can be utilized as an additional investigation tool for deep space missions.

The challenges of the Minerva-II2 deployment images were discussed. The Minerva-II2 represents a highly symmetric body, with limited high-noise deployment images and no means of independent data acquisition or validation. Working with such limited data is likely to continue to be the norm for deep space missions. This paper provides an example roadmap for future deep-space trajectory and pose estimation problems.

Acknowledgements

All real images captured by the Hayabusa 2 ONC camera system are credited to JAXA, Chiba Institute of Technology, University of Tokyo, Kochi University, Rikkyo University, Nagoya University, University of Aizu, and AIST.

References

[1] Concepts in digital imaging technology: CCD saturation and blooming. https://hamamatsu.magnet.fsu.edu/articles/ccdsatandblooming.html. Accessed 31 August 2020. 5

[2] HRNet. https://github.com/HRNet. 5, 6

[3] ORB-SLAM2. https://github.com/raulmur/ORB_SLAM2. Accessed 15 January 2020. 8

[4] How to generate equidistributed points on the surface of a sphere. https://www.cmu.edu/biolphys/deserno/pdf/sphere_equi.pdf, September 2004. Accessed 30 July 2020. 4

[5] Yusuke Oki andd Kent Yoshikawa, Hiroshi Takeuchi, SHota Kikuchi, Hitoshi Ikeda, Daniel Scheeres, Junichiro Kawaguchi, Yuto Takei, Yuya Mimasu, Naoka Ogawa, Go Ono, Fuyuto Terui, Manabu Yamada, Toru Kouyama, Shingo Kameda, Kazuya Yoshida, Kenji Nagaoka, Tetsuo Yoshimitsu, Takanao Saiki, and Yuichi Tsuda. Orbit insertion strategy of Hayabusa2's rover with the large release uncertainty around asteroid Ryugu. *Asterodynamics*, 4:309–329, November 2020. 2, 3, 4

[6] Pedro F. Proença and Yang Gao. Deep learning for spacecraft pose estimation from photorealistic rendering. *arXiv*, (arXiv:1907.04298v2), August 2019. 1

[7] Bo Chen, Jiewei Cao, Álvaro Parra, and Tat-Jun Chin. Satellite pose estimation with deep landmark regression and nonlinear pose refinement. *arXiv*, (arXiv:1908.11542v1), August 2019. 1, 5, 6

[8] Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S. Huang, and Lei Zhang. Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In *CVPR*, 2020. 6

[9] Develop3D. Solidworks graphics optimization guide. https://www.solidworks.com/sw/docs/SolidWorks_Graphic_Optimization_Guide.pdf, 2013. Accessed 18 February 2021. 3

[10] ESA. The current state of space debris. https://www.esa.int/Safety_Security/Space_Debris/The_current_state_of_space_debris, 2020. Accessed 11 February 2021. 1

[11] United Nations Office for Outer Space Affairs. Space debris mitigation guidelines of the committee on the peaceful uses of outer space. Technical Report V.09-88517, Vienna, Austria, January 2010. 1

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[13] JAXA. About engineering test satellite vii "kiku-vii" (ets-vii). https://global.jaxa.jp/projects/sat/ets7/index.html, 1997. Accessed 11 February 2021. 1

[14] JAXA. Global efforts to deal wih the problem of space debris. https://global.jaxa.jp/article/2017/special/debris/nishida.html, 2017. Accessed 11 February 2021. 1

[15] Mate Kisantal, Sumant Sharma, Tae Ha Park, Dario Izzo, Marcus Märtens, and Simone D'Amico. Satellite pose estimation challenge: Dataset, competition design and results. *IEEE Transactions on Aerospace and Electronic Systems*, (arXiv:1911.02050v2), April 2020. 1, 5

[16] Herbert J. Kramer. Clearspace-1 debris removal mission. https://directory.eoportal.org/web/eoportal/satellite-missions/c-missions/clearspace-1#references, 2020. Accessed 11 February 2021. 1

[17] Tsung-Yi Lin, Michael Maire, Serge Belongie, ubomir Boudev, Ross Girshick, James Hayes, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context. February 2015. Accessed 7 July 2020. 5

[18] Jorge Lopez-Moreno, Sunil Hadap, Erik Reinhard, and Diego Gutierrez. *Light source detection in photographs*, pages 161–167. Eurographics Digital Library, San Sebastián, September 2009. 4

[19] MATLAB. angvel. https://www.mathworks.com/help/robotics/ref/quaternion.angvel.html. 6

[20] MATLAB. bundleadjustment. https://www.mathworks.com/help/vision/ref/bundleadjustment.html. 6, 7

[21] MATLAB. estimateworldcamerapose. https://www.mathworks.com/help/vision/ref/estimateworldcamerapose.html. 6, 7

[22] MATLAB. fit. https://www.mathworks.com/help/curvefit/fit.html#bto2vuv-3. 3, 7

[23] Weihong Deng Mei Wang. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, October 2018. 6

[24] Raúl Mur-Artal, J.M.M. Montiel, and Juan D. Tardós. Orb-slam: A versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015. 8

[25] Abid Murtaza, Syed Jahanzeb Hussain Pirzada, Tongge Xu, and Liu JianWei. Orbital debris threat for space sustainability and way forward (review article). *IEEE Access Multidisciplinary Open Access Journal*, (2020.2979505), April 2020. 1

[26] Angadh Nanhangud, Peter C. Blacker, Saptarshi Bandyopadhyay, and Yang Gao. Robotics and AI-enabled on-orbit operations with future generation of small satellites. *IEEE*, 106(10.1109/JPROC.2018.2794829):429–439, February 2018. 1, 8

[27] NASA. Handbook for limiting orbital debris. https://standards.nasa.gov/standard/osma/nasa-hdbk-871914, April 2018. 1

[28] NASA. Space debris. https://www.nasa.gov/centers/hq/library/find/bibliographies/space_debris, 2019. Accessed 11 February 2021. 1

[29] ESA Space Debris Office. ESA's annual space environment report. Technical Report GEN-DB-LOG-00288-OPS-SD, D-64293 Darmstadt Germany, September 2020. 1

[30] Orbital Debris Program Office. Legend: 3d/od evolutionary model. https://orbitaldebris.jsc.nasa.gov/modeling/legend.html, 2020. Accessed 11 February 2021. 1

[31] Tae Ha Park, Sumant Sharma, and Simone D'Amico. Towards robust learning-based pose estimation of noncooperative spacecraft. *AAS/AIAA Astrodynamics Specialist Conference*, August 2019. 1

[32] Andrew Price and Kazuya Yoshida. Synthetic Minerva-II2. https://github.com/Price-SRL-Tohoku/Synthetic-Minerva-II2, April 2021. 2, 8

[33] Mark. C. Priyant and Kamath Surekha. Review of active space debris removal systems. *Space Policy*, 47:194–206, February 2019. 1

[34] Sumant Sharma. *Pose Estimation of Uncooperative Spacecraft Using Monocular Vision and Deep Learning*. PhD thesis, Stanford University, Stanford, California, USA, August 2019. 1, 4, 5

[35] Joan Solà. Simultaneous localization and mapping with the extended kalman filter. http://www.iri.upc.edu/people/jsola/JoanSola/objectes/curs_SLAM/SLAM2D/SLAM%20course.pdf, October 2014. Accessed 15 January 2020. 8

[36] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019. 5

[37] H. Suzuki, M. Yamada, T. Kouyama, E. Tatsumi, S. Kameda, R. Honda, H. Sawada, N. Ogawa, T. Morota, C. Honda, N. Sakatani, M. Hayakawa, Y. Yokota, Y. Yamamoto, and S. Sugita. Initial inflight calibration for Hayabusa2 optical navigation camera (onc) for science observations of asteroid Ryugu. *Icarus*, 300:341–359, September 2017. 2

[38] ESA Advanced Concepts Team and Stanford SLAB. Kelvins pose estimation challenge. https://kelvins.esa.int/satellite-pose-estimation-challenge/home/, July 2019. Accessed 18 February 2021. 1, 5

[39] Yuichi Tsuda, Makoto Yoshikawa, Masanao Abe, Hiroyuki Minamino, and Satoru Nakazawa. System design of the hayabusa 2 - asteroid sample return mission to 1999 ju3. *Acta Astronautica*, 91:356–362, November 2013. 2

[40] Wei Yang, Shuang Li, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Learning feature pyramids for human pose estimation. In *ICCV*, 2017. 6