Satellite Pose Estimation with Deep Landmark Regression and Nonlinear Pose Refinement

Bo Chen, Jiewei Cao, Álvaro Parra, Tat-Jun Chin
School of Computer Science, The University of Adelaide
Adelaide, South Australia, 5005 Australia
{bo.chen, jiewei.cao, alvaro.parrabustos, tat-jun.chin}@adelaide.edu.au

Abstract

We propose an approach to estimate the 6DOF pose of a satellite, relative to a canonical pose, from a single image. Such a problem is crucial in many space proximity operations, such as docking, debris removal, and inter-spacecraft communications. Our approach combines machine learning and geometric optimisation, by predicting the coordinates of a set of landmarks in the input image, associating the landmarks to their corresponding 3D points on an a priori reconstructed 3D model, then solving for the object pose using non-linear optimisation. Our approach is not only novel for this specific pose estimation task, which helps to further open up a relatively new domain for machine learning and computer vision, but it also demonstrates superior accuracy and won the first place in the recent Kelvins Pose Estimation Challenge organised by the European Space Agency (ESA).

1. Introduction

Estimating the 6DOF pose of space-borne objects (e.g., satellites, spacecraft, orbital debris) is a crucial step in many space operations such as docking, non-cooperative proximity tasks (e.g., debris removal), and inter-spacecraft communications (e.g., establishing quantum links). Existing solutions are mainly based on active sensor-based systems, e.g., the TriDAR system which uses LiDAR [12, 28]. Recently, monocular pose estimation techniques for space applications are drawing significant attention due to their lower power consumption and relatively simple requirements [11, 31, 30, 9].

Due to the importance of the problem, the Advanced Concepts Team (ACT) at ESA recently held a benchmark competition called Kelvins Pose Estimation Challenge (KPEC) [3]; given images that depict a known satellite under different unknown poses (see Figure 1), estimate the pose of the satellite in each image. To develop their algorithms, the challenge participants are given a set of training images containing the target satellite with ground truth poses; Section 1.1 provides more details of the dataset.

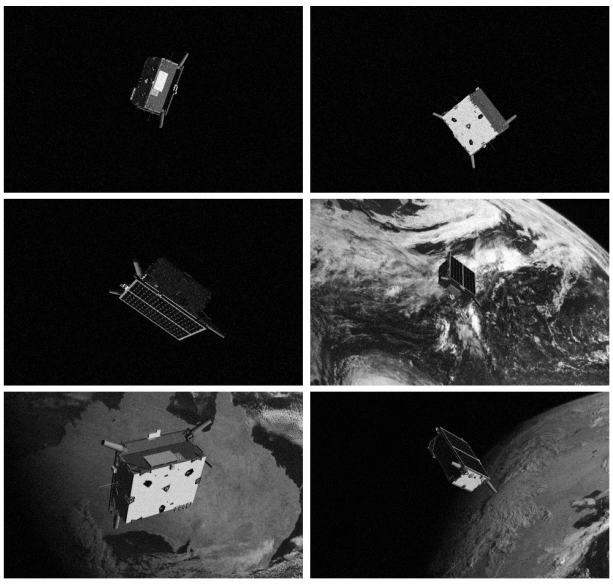


Figure 1: Sample images of the Tango satellite from SPEED [30]. Note the significant variations in object size, object orientation, background and lighting condition.

The scenario considered in KPEC is a special case of monocular vision-based object pose estimation [14, 34]. This is because the target object (the "Tango" satellite) is known beforehand, and there is no need to generalize the pose estimator to unseen-before instances of the object class (e.g., other satellites). However, the background environment can still vary, as exemplified in Figure 1. Contrast the KPEC scenario to the generic pose estimation setting [14, 34], where the provenance of the target object is unknown *a priori* and generalising to unseen-before instances is necessary (e.g., a car pose estimator must work on all kinds of cars).

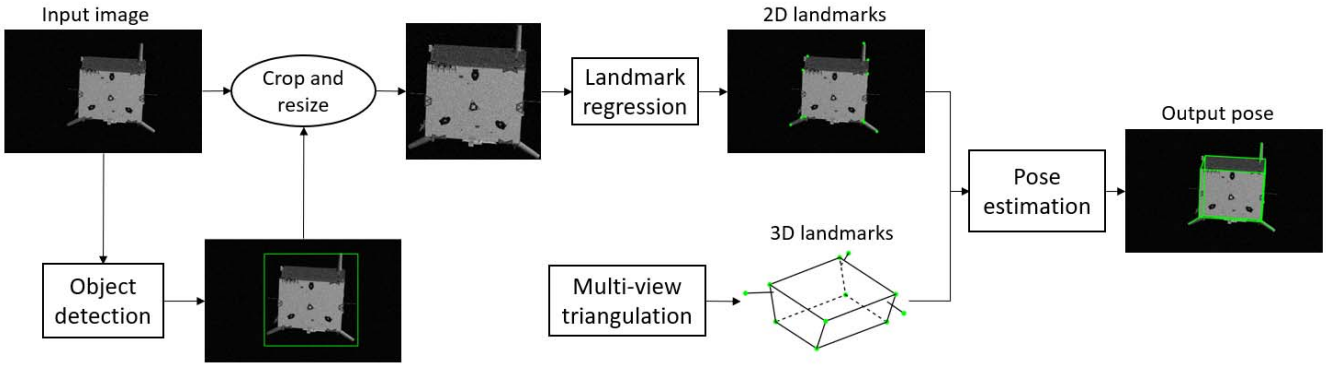


Figure 2: Overall pipeline of our satellite pose estimator.

Under the KPEC setting, we developed a monocular pose estimation technique for space-borne objects such as satellites. Inspired by works that combine the strength of deep neural networks and geometric optimisation [26, 25, 35], our approach contains three main components:

1. using the training images, reconstruct a 3D model of the satellite by multi-view triangulation;
2. train a deep network to predict the position of predefined landmark points in the input image;
3. solve for the pose of the object in the image using the 2D-3D correspondences of the predicted landmarks via robust geometric optimisation.

A high level pipeline of our framework is illustrated in Figure 2. Our code can be accessed in [4].

As suggested above, our method fully takes advantages of all available data and assumptions of the problem. This plays a significant role in producing highly-accurate 6DOF pose estimation for the KPEC. Specifically, our method commits an average cross validation (CV) error of 0.7277 degrees for orientation and 0.0359 metres for translation on the KPEC training set. We achieved an overall score of 0.0094 on the test set which ranked us the first place in KPEC. The rest of the paper first reivews related works and then describes our method and results in detail.

1.1. Dataset

The KPEC was designed around the Spacecraft PosE Estimation Dataset (SPEED) [30], which consists of high-fidelity grayscale images of the Tango satellite; see Figure 1. There are 12,000 training images with ground truth 6DOF poses (position and orientation) and 2,998 testing images without ground truth. Each image is of size 1920×1200 pixels. Half of the available images have no background (i.e., the background is the space void) while the other half contain the Earth as the background. Mirroring the setting during proximity operations, the size, orientation and lighting condition of the satellite in the images vary significantly, e.g., the number of object pixels vary between 1k and 500k; see Figure 3 for an example. For more details of the dataset, see [30].

2. Related works

Monocular vision-based pose estimation has a large body of literature. We review the major classes of previous work, before surveying the specific case of spacecraft pose estimation.

2.1. Monocular pose estimation

Keypoint methods Traditional pose estimation techniques usually use hand-crafted keypoint detectors and descriptors, e.g., SIFT [21, 20], SURF [6], MSER [22] and BRIEF [8]. The key step is to produce a set of 2D-2D or 2D-3D keypoint correspondences, then estimate the pose using non-linear optimisation from the correspondence set. The keypoints are detected automatically and described using heuristic measures of geometric and photometric invariance. However, while the keypoint methods are robust to a certain extent, they typically fail where there is large variations in pose and lighting conditions. Nonetheless, the earlier research has given birth to effective and well-

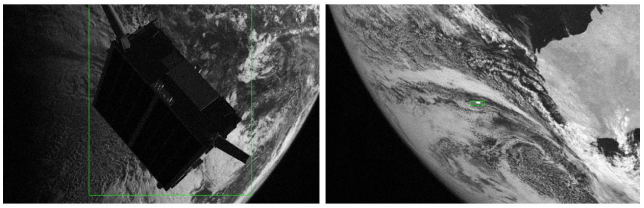


Figure 3: Large variation in object size in the images.

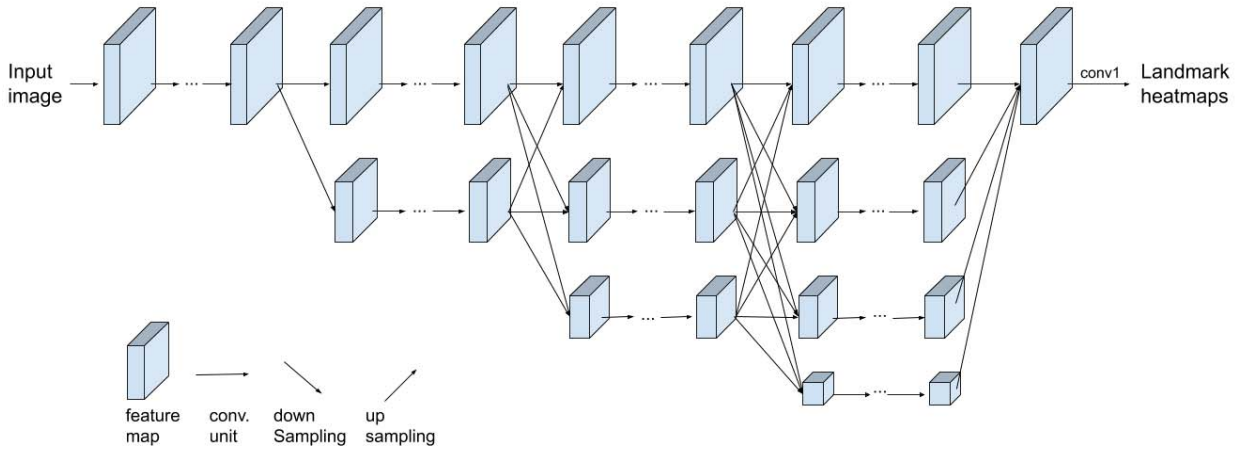


Figure 4: Illustration of the HRNet architecture in our landmark regression model.

understood geometric algorithms (e.g., PnP solvers) that are able to estimate the pose accurately and robustly, *given* a reasonable correspondence set; we exploit these techniques in our pipeline.

End-to-end learning The success of deep learning in image classification and object detection has motivated a large number of works on end-to-end learning for pose estimation [34, 7, 15, 16, 17, 23]. Generally speaking, these methods exploit the convolutional neural network (CNN) architecture to learn a complex non-linear function that maps an input image to an output pose. While such end-to-end methods have demonstrated some success, they have not achieved similar accuracy as geometry-based solutions (e.g., those that optimise pose from a correspondence set). Moreover, recent work [29] suggests that "*absolute pose regression approaches are more closely related to approximate pose estimation via image retrieval*", thus they may not generalise well in practice.

Feature learning methods Instead of handcrafting descriptors to be robust against varying kinds of distortion so that the distances between them can be used reliably to indicate keypoint matching, some methods resort to machine learning to identify keypoints detected from different views, such as Fern [24]. It uses a Naive Bayes classifier to recognize keypoints based on a binary descriptor similar to BRIEF [8], which is produced by pixel intensity comparisons.

While the keypoint matching problem can be solved using machine learning, deep CNN-based feature learning methods typically fix the 2D-3D keypoint associations and learn to predict the image locations of each corresponding 3D keypoint such as [26, 25, 35]. They mainly differ in model architecture and the choice of keypoints. For instance, [25] uses semantic keypoints while [35] chooses the vertices of the 3D bounding box of an object. In our spaceborne scenario, objects are typically not occluded and have relatively rich texture. As a result, we opt for object surface keypoints in order to better relate them to strong visual features.

Another common characteristic of aforementioned CNN-based methods is that, in spite of their various designs of architecture, they all gradually transform the feature maps of the input image from high-resolution representations to low-resolution representations, and recover them to high-resolution representations again at a later stage. Recent research has shown the importance of maintaining a high-resolution representation during the whole process in various tasks including object detection and human pose estimation [32, 33]. Specifically, the High-Resolution Net (HRNet) [32] which maintains a high-resolution representation while exchanging information across the parallel multi-resolution subnetworks throughout the whole process, as illustrated in Figure 4, produces heatmaps of landmarks with superior spatial precision. To achieve state-of-the-art accuracy in satellite pose estimation, in our framework we use the HRNet for predicting the locations of 2D landmarks in each image.

2.2. Spacecraft pose estimation

Monocular spacecraft pose estimation techniques usually adopt a model-based approach. For example, [11, 31] first preprocess the images and use feature detectors to identify prominent features such as line segments and basic geometric shapes. Search algorithms are then used to find the right matches between the detected features and the 3D structure. Lastly poses are computed using PnP solvers such as EPnP [19] and are further refined using optimisation techniques. As summarised in Section 1, our approach

also generates 2D-3D correspondences; however, we use a trained deep network to regress the coordinates of 2D landmarks.

The Spacecraft Pose Network (SPN) [30] is the seminal work on the SPEED. SPN uses a hybrid of classification and regression neural networks for the pose estimation problem. To perform classification, SPN discretises the 3D rotation group SO(3) into m uniformly distributed base rotations. SPN first predicts the bounding box of the satellite in the image with an object detection sub-network. Then, a classification sub-network retrieves the n most relevant base rotations from the feature map of the detected object. This regression sub-network learns a set of weights and outputs the predicted rotation as a weighted average of the n base rotations. Lastly, SPN solves the relative translation of the satellite utilising constraints from the predicted bounding box and rotation.

For a more comprehensive survey of spacecraft pose estimation, we refer the reader to [9].

3. Methodology

Figure 2 describes the overall pipeline of our methodology, which consists of several main modules: using a small subset of manually chosen training images (9 images were chosen), we first reconstruct a 3D structure of the satellite with a number of manually chosen landmarks (11 was chosen in our implementation) via multi-view triangulation (recall that the training images were supplied with ground truth poses). An object detection network is then used to predict the 2D bounding box of the satellite in the input image. The bounded subimage is then subjected to a landmark regression network to predict the 11 landmark image positions. Finally, we solve for the poses using the predicted 2D-3D correspondences. Details of the main steps are described in the rest of this section. Our code is available in [4].

3.1. Multi-view triangulation

We represent the structure of the object with a small number N of 3D landmarks $\{\mathbf{x}_i\}_{i=1}^N$ such that they correspond to strong visual features in the images. For the satellite, we select its eight corners plus the centres of the ends of its three antennas, which make a total of $N = 11$ landmarks. We use multi-view triangulation to reconstruct the 3D structure. To generate the input for triangulation (i.e., 2D-3D correspondences), we manually match every 3D point with 2D corresponding points over a few hand-picked close-up images from the training set. Let $\mathbf{z}_{i,j}$ denote the 2D coordinates of the i-th landmark obtained from the j-th image, the 3D landmarks $\{\mathbf{x}_i\}$ are reconstructed by solving the following objective[1]:

$$\min_{\{\mathbf{x}_i\}_{i=1}^N} \sum_{i,j} ||\mathbf{z}_{i,j} - \pi_{\mathbf{T}_j^*}(\mathbf{x}_i)||_2^2 \,, \tag{1}$$

where $\mathbf{T}_j^*$ is the ground truth pose of image j and $\pi_{\mathbf{T}}$ is the projective transformation of a structural point into the image plane with pose $\mathbf{T}$ and known camera intrinsics. Figure 5 shows the 11 selected 3D landmarks and the reconstructed model as a wireframe.

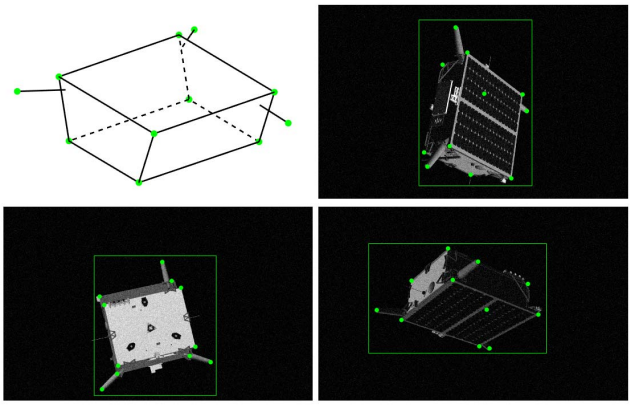


Figure 5: The reconstructed 3D model with 11 landmarks and 3 examples of the bounding boxes determined by the projected 2D landmarks.

3.2. Object detection

Our pipeline starts by obtaining a bounding box of the object in the image. The aforementioned set of structural landmarks $\{\mathbf{x}_i\}$ facilitates object detection since the convex hull of their 2D matches $\{\mathbf{z}_i\}$ covers almost the whole object in any image. Hence a simple but effective method to obtain the ground truth bounding box is to slightly relax the (axis-aligned) minimum rectangle that encloses all $\mathbf{z}_i$, as shown in Figure 5. We use this method for the training images for which we obtain the ground truth 2D landmarks $\{\mathbf{z}_i^*\}$ by projecting $\{\mathbf{x}_i\}$ to the image plane with the ground truth camera pose $\mathbf{T}^*$, *i.e.*,

$$\mathbf{z}_i^* = \pi_{\mathbf{T}^*}(\mathbf{x}_i),\ i = 1, ..., N \,. \tag{2}$$

For the testing images, we train an object detection model to predict the bounding boxes. We use an off-the-shelf object detection model described in [33], which applies an HRNet as backbone in the Faster-RCNN [27] framework. The HRNet backbone is initialised with a pre-trained model HRNet-W18-C[2] [33]. We train the detection model on the MMDetection platform [10] and follow the training settings as in [33].

[1] We used the routine `triangulateMultiview` in MATLAB.
[2] The pretrained model was downloaded from [2].

3.3. Landmark regression

Each training image is coupled with a bounding box and a set of ground truth 2D landmarks $\{\mathbf{z}_i^*\}$ as described in Section 3.2. We use these labels to supervise the training of a regression model to predict the 2D landmarks in the testing images. Additionally, to handle images that only capture partial object, we label the visibility v_i of each 2D landmark $\mathbf{z}_i^*$ of each image in the training set where

$$v_i = \begin{cases} 1 & \text{if } \mathbf{z}_i^* \text{ is inside image frame,} \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

We used the HRNet as described in [32] to regress the 2D landmark locations. Specifically we used pose-hrnet-w32 [1] for our architecture (Figure 4), which has 32 channels in the highest resolution feature maps. The output of the model is a tensor of 11 heatmaps; one for each 3D landmark. Because of this model-designed one-to-one association between 3D landmarks and heatmaps, the model solely has to learn the image location of each 3D landmark but not the heatmap-3D landmark associations.

To increase the prediction accuracy as well as robustness against the Earth background, we crop each image with their bounding boxes and resize them to fit the input window of the regression model. We conduct this process in both the training and the testing phase. For the later, we predict the bounding boxes of the testing images with the object detection model. Because HRNet maintains a high-resolution representation, it is able to produce high-resolution heatmaps with superior spatial accuracy. To leverage this characteristic of HRNet, we increased the size of the input window as well as the size of the output heatmaps to 768×768 from the default 256×256.

We train the model from scratch by minimising the following loss:

$$\ell = \frac{1}{N} \sum_{i=1}^{N} v_i (h(\mathbf{z}_i) - h(\mathbf{z}_i^*))^2 \,, \tag{4}$$

i.e., the mean squared errors between the predicted heatmaps $h(\mathbf{z}_i)$ and ground truth heatmaps $h(\mathbf{z}_i^*)$ of the visible landmarks in each image. The notation $h(\cdot)$ denotes a heatmap representation of a 2D point. We generate the ground truth heatmaps as 2D normal distributions with means equal to the ground truth locations of each landmark, and standard deviations of 1-pixel. The loss function ℓ is defined based on a single image. In a mini batch, ℓ is simply averaged. The model is trained for 180 epochs with the Adam optimizer [18]. Other training setup is adopted from [32].

3.4. Pose estimation

The final step in our pipeline is to estimate the pose $\mathbf{T} \in \mathrm{SE}(3)$ for a test image given the predicted 2D-3D correspondences $\{(\mathbf{z}_i, \mathbf{x}_i)\}$ as described in Section 3.3. We estimate $\mathbf{T}$ by solving the robust non-linear least-squares problem

$$\min_{\mathbf{T}} \sum_i L_\delta(r_i(\mathbf{T})) \tag{5}$$

with residuals

$$r_i(\mathbf{T}) = \|\mathbf{z}_i - \pi_{\mathbf{T}}(\mathbf{x}_i)\|_2 \,, \tag{6}$$

and subject to cheirality constraints. $L_\delta : \mathbb{R} \to [0, \infty)$ is the Huber loss

$$L_\delta(r) = \begin{cases} \dfrac{r^2}{2} & \text{if } |r| \le \delta \\ \delta|r| - \dfrac{\delta^2}{2} & \text{otherwise.} \end{cases} \tag{7}$$

We use Levenberg-Marquardt (LM) to solve Eq. (5); we called LMPE to our C++ implementation with the Ceres Solver [5]. We can run LMPE after setting δ and choosing an initial linearisation point $\mathbf{T}_0$; however, picking a value for δ, and potential outlying correspondences could impact on producing an accurate estimation. Instead, we propose a Simulated Annealing scheme (SA-LMPE) as depicted in Algorithm 1 to progressively adjust δ and remove potential outlying correspondences. A correspondence $(\mathbf{z}_i, \mathbf{x}_i)$ is regarded as an outlier if

$$r_i(\mathbf{T}^*) > \epsilon \tag{8}$$

for a threshold ϵ, and the ground truth pose $\mathbf{T}^*$. In practice, we use the residual with respect to the current pose $r_i(\mathbf{T}_{t+1})$ to indicate potential outliers for removal.

Algorithm 1 SA-LMPE.

Require: 2D-3D matches $H_0 := \{(\mathbf{z}_i, \mathbf{x}_i)\}$, initial pose $\mathbf{T}_0$, initial values for δ and ϵ, cooling parameters $\delta_{\min}, \epsilon_{\min} > 0$, $0 < \lambda_\delta, \lambda_\epsilon \le 1$, and number of iterations $t_{\max}$.

1: $t \leftarrow 0$.
2: **while** $t < t_{\max}$ **do**
3: $\quad \mathbf{T}_{t+1} \leftarrow$ LMPE$(H_t, \mathbf{T}_t, \delta)$.
4: $\quad H_{t+1} \leftarrow \{(\mathbf{z}_i, \mathbf{x}_i) \in H_t \,|\, r_i(\mathbf{T}_{t+1}) \le \epsilon\}$.
5: $\quad \delta \leftarrow \max(\delta_{\min}, \lambda_\delta \delta)$.
6: $\quad \epsilon \leftarrow \max(\epsilon_{\min}, \lambda_\epsilon \epsilon)$.
7: $\quad t \leftarrow t + 1$.
8: **end while**
9: **return** T_t.

There is a virtuous circle in our annealing process: an accurate pose will help on carefully removing potential outliers (Line 4), while lesser outlying corrupted data will produce a more accurate estimation (Line 3). Thus, initial δ and ϵ values progressively "cool down" (Step 5 and Step 6),

Metric	SPN [30] (on test set)			Ours (on training set CV)		
Mean IOU	0.8582			0.9534		
Median IOU	0.8908			0.9634		
Mean E_R (degree)	8.4254			0.7277		
Median E_R (degree)	7.0689			0.5214		
Mean E_T (metre)	N/A			0.0359		
Median E_T (metre)	N/A			0.0147		
Mean $\lvert \boldsymbol{t}^* - \boldsymbol{t} \rvert$ (metre)	[0.0550,	0.0460,	0.7800]	[0.0040,	0.0040,	0.0346]
Median $\lvert \boldsymbol{t}^* - \boldsymbol{t} \rvert$ (metre)	[0.0240,	0.0210,	0.4960]	[0.0031,	0.0030,	0.0134]

Table 1: Performance comparison between the SPN and the proposed method.

until reaching minimum predefined values ($\delta_{\text{min}}, \epsilon_{\text{min}}$) or a maximum number of iterations t_{max}.

For the SPEED images, we obtained the initial pose $\mathbf{T}_0$ in Algorithm 1 by using a RANSAC fashion PnP solver[3] (with kernel P3P [13] and minimal four-points sets) on the predicted correspondences.

4. Evaluation

In this section we report the evaluation metrics and experimental results of our methodology.

4.1. Metrics

We evaluate the estimated pose of each image using a rotation error E_R and a translation error E_T. Let q^* and q denote the rotation quaternion ground truth of an image and its estimation. Analogously, let $\boldsymbol{t}^*$ and $\boldsymbol{t}$ denote the ground truth and estimated translation vectors of an image. We then define E_R and E_T as

$$E_R = 2\cos^{-1}\left(|z|\right) , \tag{9}$$

where z is the real part of the Hamilton product between q^* and the conjugate of q, *i.e.*, $z + \boldsymbol{c} = q^* \,\text{conj}(q)$, where $\boldsymbol{c}$ is the vector part of the Hamilton product and

$$E_T = ||\boldsymbol{t}^* - \boldsymbol{t}||_2 . \tag{10}$$

We report our object detection results via the Intersection Over Union (IOU) score based on CV. For each image, its IOU score is the intersection area divided by the union area of the predicted and the ground truth bounding boxes.

We compare against KPEC's participants trough the scores defined in the KPEC: namely the rotation score S_R, the translation score S_T, and the overall score S. S_R is the same as E_R but in radians,

$$S_T = \frac{||\boldsymbol{t}^* - \boldsymbol{t}||_2}{||\boldsymbol{t}^*||_2} , \tag{11}$$

and

$$S = S_R + S_T . \tag{12}$$

[3]We used the routine `estimateWorldCameraPose` in MATLAB.

4.2. Experiments

Since the KPEC withheld the ground truth poses of the test set, we cannot conduct analysis based on the test set other than providing the overall score. Instead, we analysed our method using 6-fold CV over the training set. Specifically, we split the 12,000 training images into 6 groups, and then for each group, we train an object detection (Section 3.2) model and a landmark regression (Section 3.3) model with the images in the remaining 5 groups. We test each model with their respective designated group, i.e., the complement of the 5 groups we train the model with. Thus each model is equipped with a disjoint test group so that in total, they cover all 12,000 images in the training dataset.

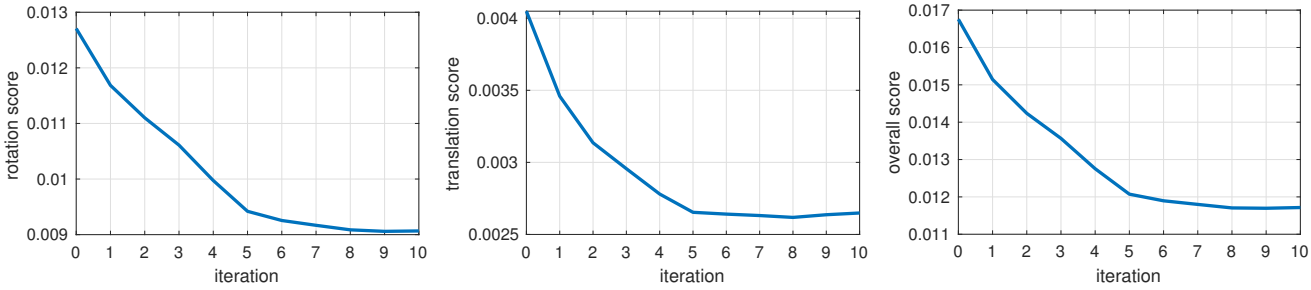
Following the above CV procedure, we estimate the pose of every training image. In effect, we predict the image coordinates of every 3D landmark to obtain 2D-3D correspondences from which we obtained an initial pose using RANSAC with a PnP kernel, which we finally refine with Algorithm 1. We make clear that we invoke Algorithm 1 with all predicted 2D-3D correspondences and not with the consensus set after RANSAC.

For the test set, we exploit the advantage of ensemble methods since we have 6 trained models resulted from the 6-fold CV. We average the 6 heatmaps predicted by the 6 trained models for each landmark and each test image before we obtain the final 2D landmark coordinates. The rest of the precedure is the same as described in Section 3.4.

4.3. Results

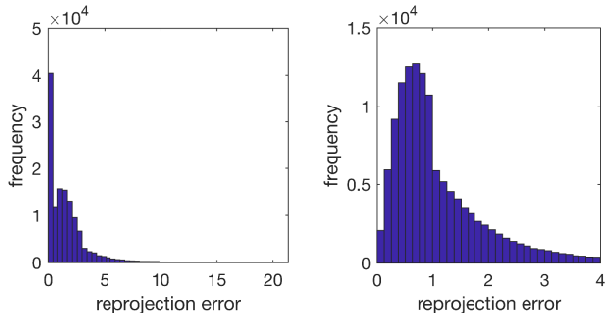
We first compare against SPN [30]; Table 1 report the performance results. Our proposed method achieves superior performances in both object detection and pose estimation. Both our rotational and translational errors are at least one degree of magnitude smaller than SPN.

In terms of the KPEC scores, our average overall score



(a) S_R (b) S_T (c) S

Figure 6: Score evolution for SA-LMPE over all images.



(a) RANSAC (b) SA-LMPE

Figure 7: Histograms of reprojection errors of all landmarks from all images with respect to (a) the initial poses obtained by RANSAC, and (b) the final poses after the SA-LMPE refinement. For better visualisation of the RANSAC histogram, we truncated its long tail by removing the 1% largest errors.

Rank	Participant Name	Score
1	UniAdelaide	0.0094
2	EPFL_cvlab	0.0215
3	pedro_fairspace	0.0571
4	stanford_slab	0.0626
5	Team_Platypus	0.0703
6	motokimura1	0.0758
7	Magpies	0.1393
8	GabrielA	0.2423
9	stainsby	0.3711
10	VSI_Feeney	0.4658

Table 2: Top 10 scores of KPEC.

of the training set based on a 6-fold CV is 0.0117. To investigate the effect of the pose refinement, Figure 6 displays the evolution of average scores during the refinement process while Figure 7 provides a comparison of reprojection residuals before and after the refinement. Based on the error distribution of the initial poses in Figure 7(a), we set $\delta = 5$ and $\epsilon = 50$ to initialise SA-LMPE. For the cooling parameters we take $\delta_{\text{min}} = 1$, $\epsilon_{\text{min}} = 4$, and $\lambda_\delta = \lambda_\epsilon = 0.7$. We set the maximal number of iterations $t_{\text{max}} = 10$. SA-LMPE removed 8495 potential outliers in total which is equivalent to approximately 0.7 outliers per image. As shown in Figure 6, the pose refinement improves the average overall score S from 0.0167 to 0.0117.

Our overall score of the test set is 0.0094 which is slightly better than the training set CV 0.0117, thanks to the benefits from the ensemble of 6 models. Table 2 provides the top 10 scores in KPEC. We provide Figure 8 and 9 for visual inspection of object detection, landmark regression and pose estimation results on a sample of the test set. Note that we did not cherry-pick the images from testing results -they were selected at random. Visual inspection indicates high accuracy of our approach even with images that have very small object size.

5. Conclusion

We propose a monocular pose estimation framework for space-borne objects such as satellite. Our framework exploits the strength of deep neural networks in feature learning and geometric optimisation in robust fitting. In particular, the high-resolution representation of images used in HRNet enables accurate predictions of 2D landmarks while the SA-LMPE algorithm allows further removal of inaccurate predictions and refinement of poses.

Our approach won the first place in the the KPEC. Our CV-based evaluation also indicates our method significantly outperforms previous work on the SPEED benchmark.

Acknowledgement

This work was jointly supported by ARC project LP160100495 and the Australian Institute for Machine Learning.

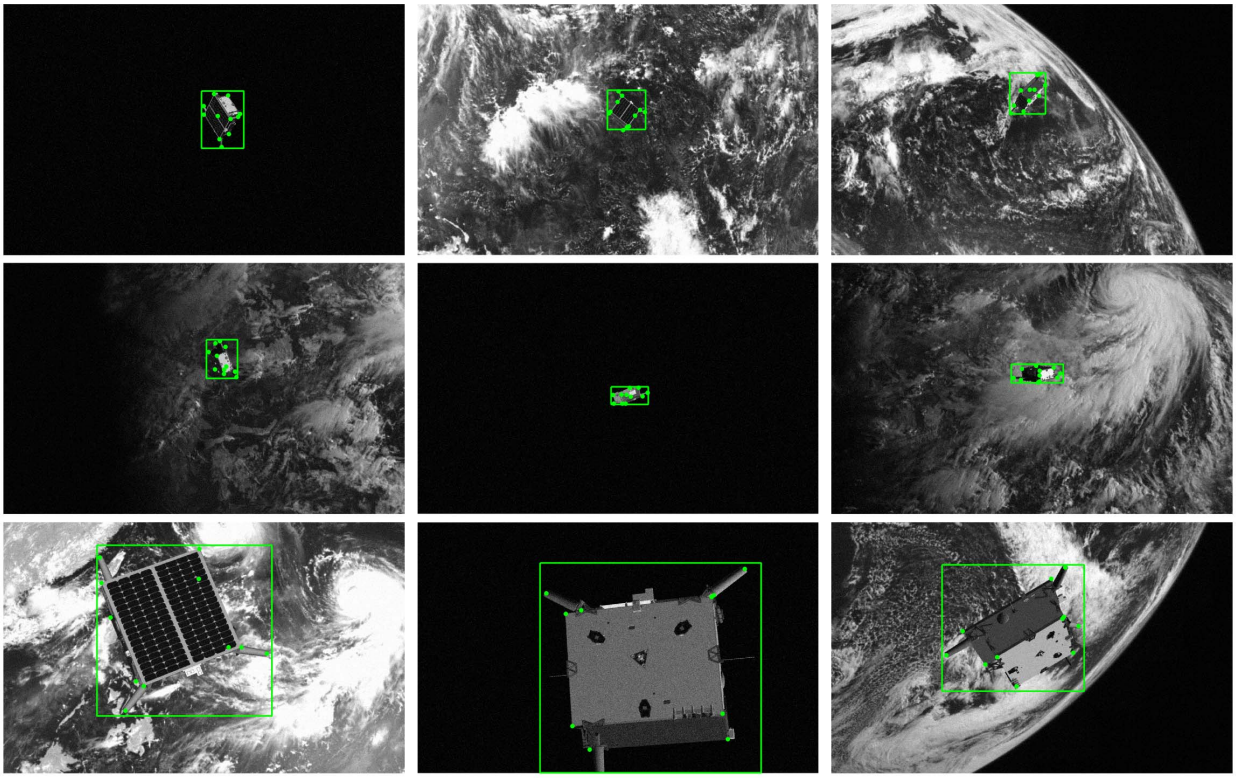


Figure 8: A montage of random test images with the predicted bounding boxes of the satellite and the estimated 2D landmarks.

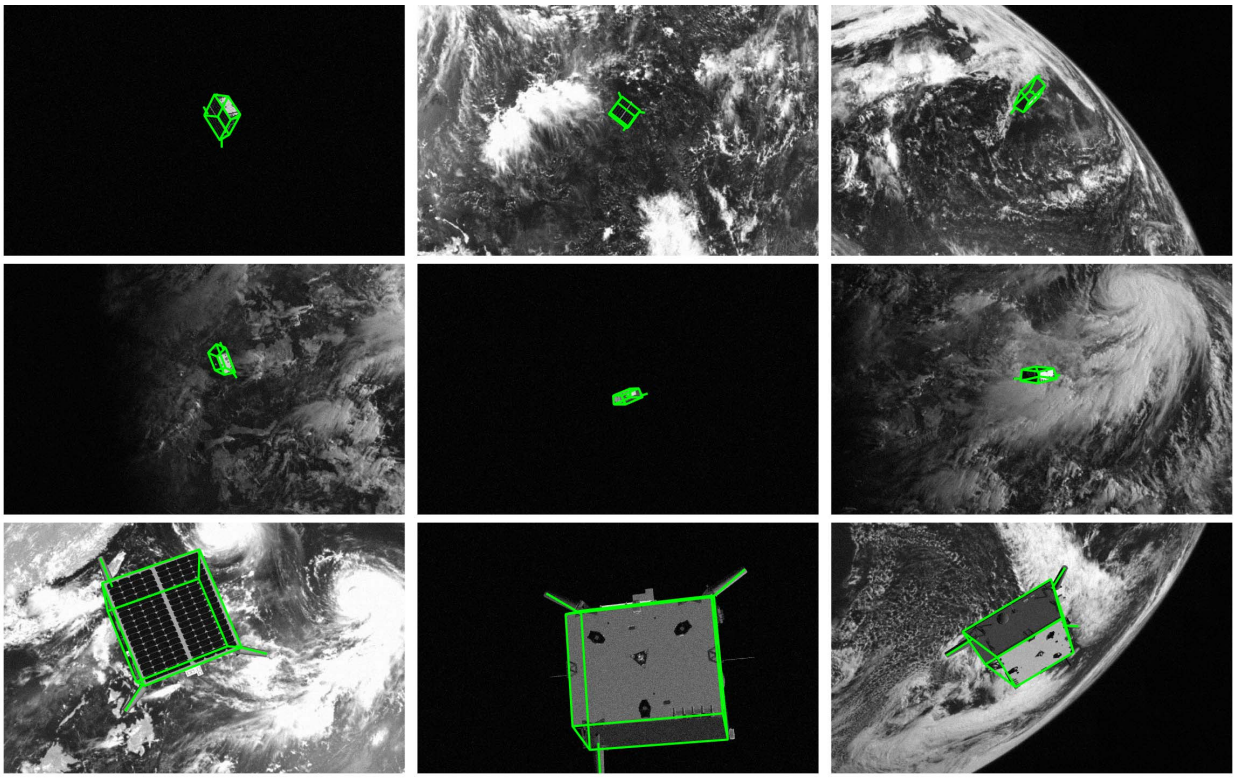


Figure 9: A montage of the same test images in Figure 8 with the predicted poses shown as green wireframes.

References

[1] Deep high-resolution representation learning for human pose estimation. https://github.com/leoxiaobin/deep-high-resolution-net.pytorch. 5

[2] High-resolution networks (HRNets) for image classification. https://github.com/HRNet/HRNet-Image-Classification. 4

[3] Kelvins pose estimation challenge. https://kelvins.esa.int/satellite-pose-estimation-challenge/home/. 1

[4] Satellite pose estimation. https://github.com/BoChenYS/satellite-pose-estimation. 2, 4

[5] S. Agarwal, K. Mierle, and Others. Ceres solver. http://ceres-solver.org. 5

[6] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *ECCV*, 2006. 2

[7] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz. Geometry-aware learning of maps for camera localization. In *CVPR*, 2018. 3

[8] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. In *ECCV*, 2010. 2, 3

[9] L. P. Cassinis, R. Fonod, and E. Gill. Review of the robustness and applicability of monocular pose estimation systems for relative navigation with an uncooperative spacecraft. *Progress in Aerospace Sciences*, 2019. 1, 4

[10] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, et al. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 4

[11] S. D'Amico, M. Benn, and J. L. Jørgensen. Pose estimation of an uncooperative spacecraft from actual space imagery. *International Journal of Space Science and Engineering*, 2(2):171–189, 2014. 1, 3

[12] C. English, S. Zhu, C. Smith, S. Ruel, and I. Christie. Tridar: A hybrid sensor for exploiting the complimentary nature of triangulation and lidar technologies. In *Proceedings of the 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, 2005. 1

[13] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng. Complete solution classification for the perspective-three-point problem. *TPAMI*, 25(8):930–943, 2003. 6

[14] T. Hodaň, F. Michel, E. Brachmann, W. Kehl, A. Glent Buch, D. Kraft, B. Drost, J. Vidal, S. Ihrke, X. Zabulis, C. Sahin, F. Manhardt, F. Tombari, T.-K. Kim, J. Matas, and C. Rother. BOP: Benchmark for 6D object pose estimation. In *ECCV*, 2018. 1

[15] A. Kendall and R. Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *ICRA*, 2016. 3

[16] A. Kendall and R. Cipolla. Geometric loss functions for camera pose regression with deep learning. In *CVPR*, 2017. 3

[17] A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *ICCV*, 2015. 3

[18] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[19] V. Lepetit, F. Moreno-Noguer, and P. Fua. Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision*, 81(2):155, 2009. 3

[20] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 2

[21] D. G. Lowe et al. Object recognition from local scale-invariant features. In *ICCV*, 1999. 2

[22] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 22(10):761–767, 2004. 2

[23] I. Melekhov, J. Ylioinas, J. Kannala, and E. Rahtu. Image-based localization using hourglass networks. In *ICCV*, 2017. 3

[24] M. Ozuysal, M. Calonder, V. Lepetit, and P. Fua. Fast keypoint recognition using random ferns. *IEEE transactions on pattern analysis and machine intelligence*, 32(3):448–461, 2009. 3

[25] G. Pavlakos, X. Zhou, A. Chan, K. G. Derpanis, and K. Daniilidis. 6-dof object pose from semantic keypoints. In *ICRA*, 2017. 2, 3

[26] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao. Pvnet: Pixel-wise voting network for 6dof pose estimation. In *CVPR*, 2019. 2, 3

[27] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 4

[28] S. Ruel, T. Luu, and A. Berube. Space shuttle testing of the tridar 3d rendezvous and docking sensor. *Journal of Field Robotics*, 29(4):535–553, 2012. 1

[29] T. Sattler, Q. Zhou, M. Pollefeys, and L. Leal-Taixe. Understanding the limitations of cnn-based absolute camera pose regression. In *CVPR*, 2019. 3

[30] S. Sharma and S. D'Amico. Pose estimation for non-cooperative rendezvous using neural networks. In *AAS/AIAA Astrodynamics Specialist Conference*, 2019. 1, 2, 4, 6

[31] S. Sharma, J. Ventura, and S. DAmico. Robust model-based monocular pose initialization for noncooperative spacecraft rendezvous. *Journal of Spacecraft and Rockets*, 55(6):1414–1429, 2018. 1, 3

[32] K. Sun, B. Xiao, D. Liu, and J. Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019. 3, 5

[33] K. Sun, Y. Zhao, B. Jiang, T. Cheng, B. Xiao, D. Liu, Y. Mu, X. Wang, W. Liu, and J. Wang. High-resolution representations for labeling pixels and regions. *CoRR*, abs/1904.04514, 2019. 3, 4

[34] M. Sundermeyer, Z.-C. Marton, M. Durner, M. Brucker, and R. Triebel. Implicit 3d orientation learning for 6d object detection from rgb images. In *ECCV*, 2018. 1, 3

[35] B. Tekin, S. N. Sinha, and P. Fua. Real-time seamless single shot 6d object pose prediction. In *CVPR*, 2018. 2, 3