Leveraging Temporal Information for 3D Trajectory Estimation of Space Objects

Mohamed Adel Musallam [1] mohamed.ali@uni.lu
Miguel Ortiz del Castillo [1] miguel.ortizdelcastillo@uni.lu
Kassem Al Ismaeil [1] kassem.alismaeil@uni.lu
Marcos Damian Perez [2] secondauthor@i2.org
Djamila Aouada [1] djamila.aouada@uni.lu

[1] SnT, University of Luxembourg
[2] LMO *

Abstract

This work presents a new temporally consistent space object 3D trajectory estimation from a video taken by a single RGB camera. Understanding space objects' trajectories is an important component of Space Situational Awareness, especially for applications such as Active Debris Removal, On-orbit Servicing, and Orbital Maneuvers. Using only the information from a single image perspective gives temporally inconsistent 3D position estimation. Our approach operates in two subsequent stages. The first stage estimates the 2D location of the space object using a convolution neural network. In the next stage, the 2D locations are lifted to 3D space, using temporal convolution neural network that enforces the temporal coherence over the estimated 3D locations. Our results show that leveraging temporal information yields smooth and accurate 3D trajectory estimations for space objects. A dedicated large realistic synthetic dataset, named SPARK-T*, containing 3 spacecrafts, under various sensing conditions, is also proposed and will be publicly shared with the research community.*

1. Introduction

Since the beginning of space exploration, the number of space debris has increased drastically. Debris population comes mainly from the remnants from human-made objects such as dead satellites, used rocket stages, and particles from the collision of other debris [14]. Today, these objects represent a threat as space debris incurs the risk of collision and damage to operational satellites.

To tackle this problem, one of the proposed solutions is Active Debris Removal (ADR). The premise of this method consists of capturing and disposing of large debris ($>$ 10cm). To that end, new technological challenges related to orbital rendezvous in general, and to relative navigation in particular, must be addressed. A reliable navigation system should be developed. It is required to be able to provide accurate relative state estimates of the targeted debris, over a wide range of different distances, from early detection until target capture.

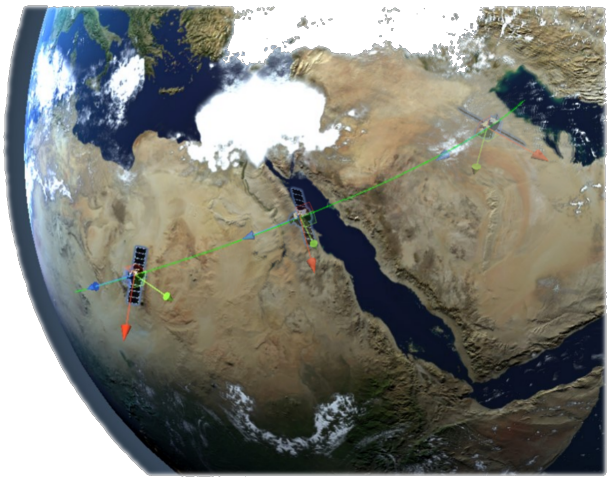

Figure 1. Spacecraft trajectory simulation

Programs such as *CleanSpace* [5], *RemoveDebris* [13], *AnDROiD* [20], and future missions such as *ClearSpace-1* [3] lead the efforts to provide a cleaner space. Depending on the specific mission objectives, debris state estimates can cover either the relative position and velocity (3-DoF relative navigation) of the targeted object, or the relative position, velocity, attitude (6-DoF relative navigation), as well as the target trajectory.

The contribution of this paper is twofold: First, we propose a new spatio-temporal approach for space ob-

*This work was funded by the Luxembourg National Research Fund (FNR), under the project reference BRIDGES2020/IS/14755859/MEET-A/Aouada, and by LMO (https://www.lmo.space).

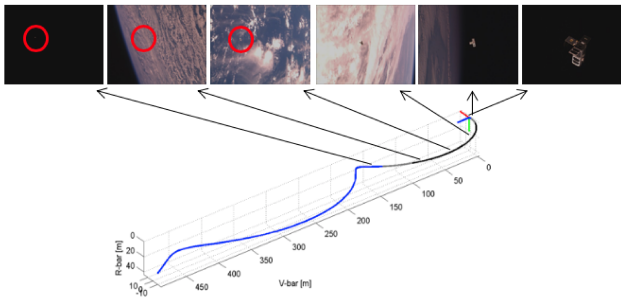

Figure 2. Tracking a spacecraft within vision-based navigation camera field of view over the reference trajectory from RemoveDebris mission[13].

ject 3D trajectory estimation. Second, a large and, to the best of our knowledge, the first photo-realistic synthetic dataset with temporal information for space object 3D trajectory estimation was created and will be publicly shared with the research community. This dataset is named *SPARK-T* [1] , where "T" stands for *trajectories*, and "SPARK" is in reference to the recent challenge on SPAcecraft Recognition leveraging Knowledge of Space Environment (SPARK) [18], for which the same simulator was used.

In this paper, we focus on space object 3D trajectory estimation from videos where we exploit the temporal information. The proposed approach follows a top-down strategy. First, we start by detecting the center of a space object as 2D coordinates for each frame. Then, we lift the detected 2D coordinates to 3D space leveraging the temporal information contained in the observed video sequence. In order to test the proposed approach, a new dedicated dataset has been generated under a photo-realistic space simulation environment, with a large diversity in sensing conditions. Obtained experimental results show stable and accurate space object trajectory estimation. For ADR, such decomposition of the problem reduces the difficulty of the task at hand. It gives the possibility to control which estimation to use based on the orbital situation of the spacecraft.

The rest of the paper is organized as follows: Section 2 describes the trajectory estimation problem. Details about the proposed solution are provided in Section 3. Section 4 presents the generated dataset used for training and testing the problem. Section 5 describes the implementation details, the conducted experiments, and presents the results. Section 6 concludes this work.

2. Problem formulation

In this section, we formulate the considered problem of spacecraft 3D trajectory estimation.

[1]The SPARK-T dataset will be shared here `https://cvi2.uni.lu/`

Let $V_I = \{I_1, \cdots, I_N\}$ be a sequence of RGB images corresponding to the observed spacecraft or debris, where N is the total number of frames, and where the acquisition is done with a known camera whose intrinsic matrix is $K \in \mathbb{R}^{3\times4}$. Subsequently, the goal of this work is to estimate the trajectory of the object of interest in 3D. That is, the objective is to estimate the trajectory $\mathcal{Y} = \{\ell_1, \cdots, \ell_N\}$, where $\ell_i \in \mathbb{R}^3$.

The object may be localized on each image I_i by estimating its pixel coordinates $(u_i, v_i) \in \mathbb{R}^2$. This 2D location corresponds to the projection of the 3D location ℓ_i of the object in the scene onto a 2D image plane using the camera intrinsic parameters K such that

$$\begin{pmatrix} \grave{u}_i \\ \grave{v}_i \\ \grave{w}_i \end{pmatrix} = K(R_i\ell_i + t_i), \quad \text{and} \quad \begin{pmatrix} u_i \\ v_i \end{pmatrix} = \begin{pmatrix} \grave{u}_i/\grave{w}_i \\ \grave{v}_i/\grave{w}_i \end{pmatrix} \quad (1)$$

where R_i and t_i are the unknown space object rotation and translation, respectively, relative to the camera.

The task at hand can be formulated as a two-step problem: (1) Estimation of the object 2D location (u_i, v_i) in the image plane at each frame i for $i = 1, \cdots, N$; (2) Estimation of the corresponding 3D locations $\ell_i = (x_i, y_i, z_i)$ constituting the trajectory $\mathcal{Y}$.

3. Proposed approach

In order to estimate the 3D trajectory $\mathcal{Y}$, we cast the problem as a 2D trajectory estimation followed by lifting to 3D space [15, 22], where the hypothesis is that temporal information may compensate the lack of the third dimension. This is verified in other applications, e.g., 3D human pose estimation, where the low-dimensional 2D location over time is shown to be discriminative enough to estimate the 3D location with high accuracy [22].

In this section, we describe the main components of the proposed two-step space object 3D trajectory estimation.

3.1. 2D Location estimation

In order to estimate an object 2D location (u, v) from an RGB image I, we represent our object of interest as a single point which is a simpler and a more efficient representation. Indeed, while a common approach is to use a regular bounding box, we choose to track a selected 2D point, i.e., the origin, as it is geometrically related to the desired 3D location (x, y, z) through eq. (1).

Inspired by CenterNet [26], we use an encoder-decoder architecture based on U-Net [24] with ResNet18 [7] as the encoder for feature extraction and the Differentiable Spatial to Numerical Transform (DSNT) [19] to regress the 2D location (u, v), as shown in Figure 3.

We choose the encoder part of our architecture to be ResNet18 in order to better preserve finer details of the input image especially in the cases where the object is small or

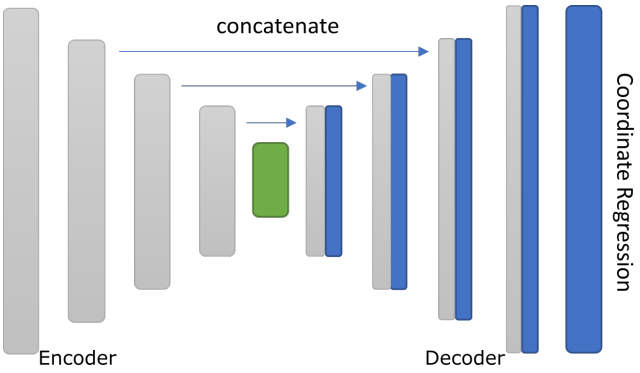

Figure 3. Proposed architecture for 2D point regression: U-Net [24] is used with ResNet18 [7] as encoder, gray blocks represent the encoder output, green block represent the bottleneck and the blue blocks represent the scaled up output of the decoder. Finally DSNT [19] for coordinate regression.

far away from the camera. In addition, skip connections are used from the encoder to the corresponding up-convolution in the decoder, and features are concatenated in each corresponding stage between the encoder and the decoder. The final convolution layers of the decoder perceive the spatial resolution of the input image, and output features are passed to a softmax function which produces a single-channel normalized heatmap where all elements are non-negative and sum to one.
This output is passed to the DSNT layer, which is fully differentiable, and exhibits good spatial generalization unlike heatmap matching, and also outputs direct numerical coordinates (u, v).

Then, for a given video sequence V_I, this 2D localization approach:

$$f : I \in \mathbb{R}^{M \times N} \mapsto (u, v) \in \mathbb{R}^2 \tag{2}$$

is applied frame by frame on V_I resulting in a sequence of estimated 2D locations $\hat{\mathcal{X}} = \{f(I_1), \cdots, f(I_N)\}$. In (2), M is the image dimension, and N is the number of frames.

3.2. 3D Trajectory estimation

Given a sequence $\mathcal{X} \subset \mathbb{R}^2$, the goal is to lift this sequence of 2D locations into the 3D space. To that end, we need to estimate a function $g(\cdot)$, which maps a sequence of 2D points sequence to its corresponding 3D sequence, such that:

$$g : \mathcal{X} \subset \mathbb{R}^2 \mapsto \mathcal{Y} \subset \mathbb{R}^3. \tag{3}$$

Estimating the 3D location from individual frames leads to a temporally incoherent result, where the independent error from each frame leads to unstable 3D position estimation over the video sequence. Thus, in our work, we follow the same approach proposed in [22, 17] for human pose estimation where a fully convolutional architecture is used to perform temporal convolution over 2D skeleton joint positions in order to estimate the 3D skeleton in a video. Therefore, the function $g(\cdot)$ is approximated by a Sequence to Sequence (Seq2Seq) Temporal Convolutional Network (TCN) model as can be seen in Figure 4 using 1D temporal convolution. Consequently, the sequence of 3D locations can be obtained using the combination of the functions $f(\cdot)$ and $g(\cdot)$, such that $\hat{\mathcal{Y}} = g \circ f(V_I)$, where $\circ$ denotes function composition.
We note that TCN is a variation of convolutional neural network for sequence modelling tasks. Compared to traditional Recurrent Neural Networks (RNNs), TCN offers more direct high-bandwidth access to past and future information. This allows TCN to be more efficient to model the temporal information of the input data with fixed size [16]. TCN can be causal; meaning that there is no information "leakage" from future to past, or non-causal where past and future information is considered. The main critical component of the TCN is the dilated convolution [9] layer, which allows to properly treat temporal order and handle long-term dependencies without an explosion in model complexity. For simple convolution, the size of the receptive field of each unit - block of input which can influence its activation - can only grow linearly with the number of layers. In the dilated convolution, the dilation factor d increases exponentially at each layer. Therefore, even though the number of parameters grows only linearly with the number of layers, the effective receptive field of units grows exponentially with the layer depth. The dilated convolution $*_d$ with a dilation factor d of a 1D signal s with a kernel of size k is defined as:

$$(k *_d s)_t = \sum_{\tau=-\infty}^{\infty} k_\tau \cdot s_{t-d\tau}. \tag{4}$$

Convolutional models enable parallelization over both the batch and the time dimension while RNNs cannot be parallelized over time [2]. Moreover, the path of the gradient between output and input has a fixed length regardless of the sequence length, which mitigates the vanishing and exploding gradients. This has a direct impact on the performance of RNNs [2]. Architectures with dilated convolutions have been successfully used for audio generation in Wavnet [21], semantic segmentation [25], machine translation [11], and 3D pose estimation [22]. As stated in [2], TCNs generally outperform most of the commonly used networks such as Long Short-Term Memory (LSTM) [8] or Gated Recurrent Unit (GRU) [4] for different tasks.

4. Data generation

In the space domain, given the difficulty of obtaining large real datasets, synthetic datasets are currently the default approach for developing DL methods for Space Sitiuational Awareness (SSA) and ADR tasks. To the best

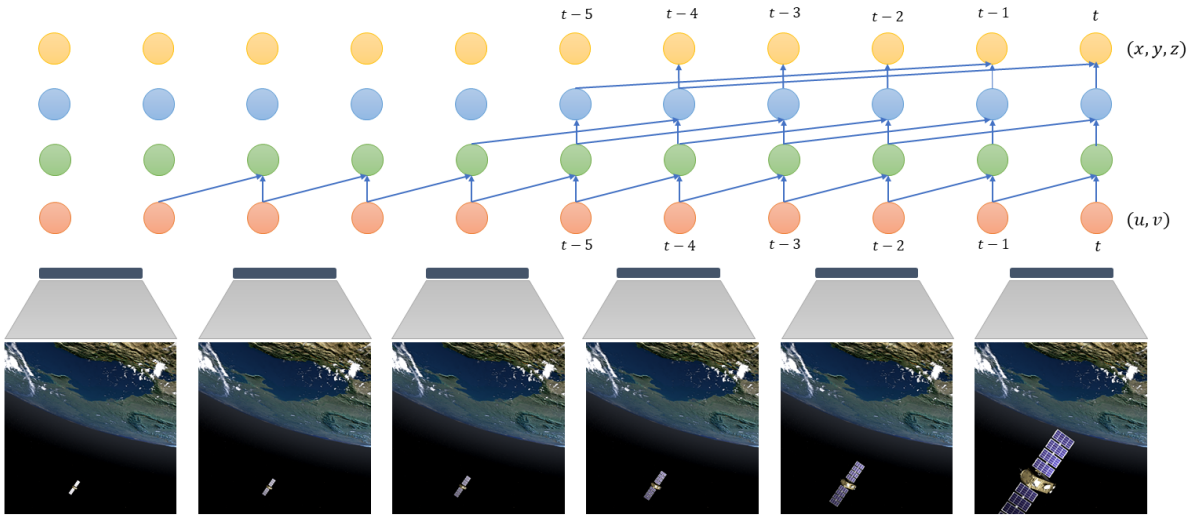

Figure 4. 3D Trajectory estimation model for each frame in time t, we forward the historical 2D coordinate (u, v) from the previous frames and estimated its 3D coordinate (x, y, z) using temporal convolution network (TCN) leading to stable and accurate trajectory estimation.

of our knowledge, existing datasets [23, 12], and more recently [10], do not provide temporal data as they were designed specifically for single image spacecraft pose estimation [6].

To study spacecraft trajectory estimation, we utilized our realistic space simulation environment, providing a large range of diversity in sensing conditions and trajectories.

Figure 5. Samples from our generated SPARK-T dataset. Top row – Jason satellite, middle row – heat shield tile, down row – CubeSat.

We used 3D models of three target spacecrafts: (1) a 3D model of *'Jason'* satellite with dimensions $3.8m \times 10m \times 2m$ with the solar panels deployed; (2) 1RU generic *'CubeSat'* with dimensions $10cm \times 11cm \times 11cm$; and (3) for debris we used a heat shield tile model with dimensions $15cm \times 10cm \times 3cm$. The 3D models were obtained from NASA 3D resources [1].

SPARK-T dataset was generated by placing the target spacecraft in different trajectories within the field of view of a camera mounted on a chaser. Furthermore, the Sun and Earth were rotated around their respective axes. This has ensured a diversity in the generated dataset with high-resolution photorealistic RGB images for different orbital scenarios.

For this work, 50 sequences were generated for each of the three spacecrafts, with 50 frames each, and including their 3D trajectories as ground truth and the corresponding R, t of the spacecraft with respect to the camera reference frame. Finally, all images were resized to 512×512 and processed with a zero-mean Gaussian blurring with variance $\sigma^2 = 1$ and an additive Gaussian white noise with variance $\sigma^2 = 0.001$.

5. Experiments

In this section, we present the experimental setup along with the obtained results. To evaluate the proposed approach, experiments were conducted on our generated spacecraft trajectories dataset presented in Section 4.

5.1. Data preparation

The data were split into 80% (i.e., 120 sequences) for training , and 20% (i.e., 30 sequences) for testing.

For training the 2D location estimation model $f(.)$ presented in Section 3.1, the training data were shuffled in order to eliminate the temporal dependency in the dataset. During training, the input 2D coordinates $p = (u, v)$ were normalized to be in the range [-1, 1], as in [19].

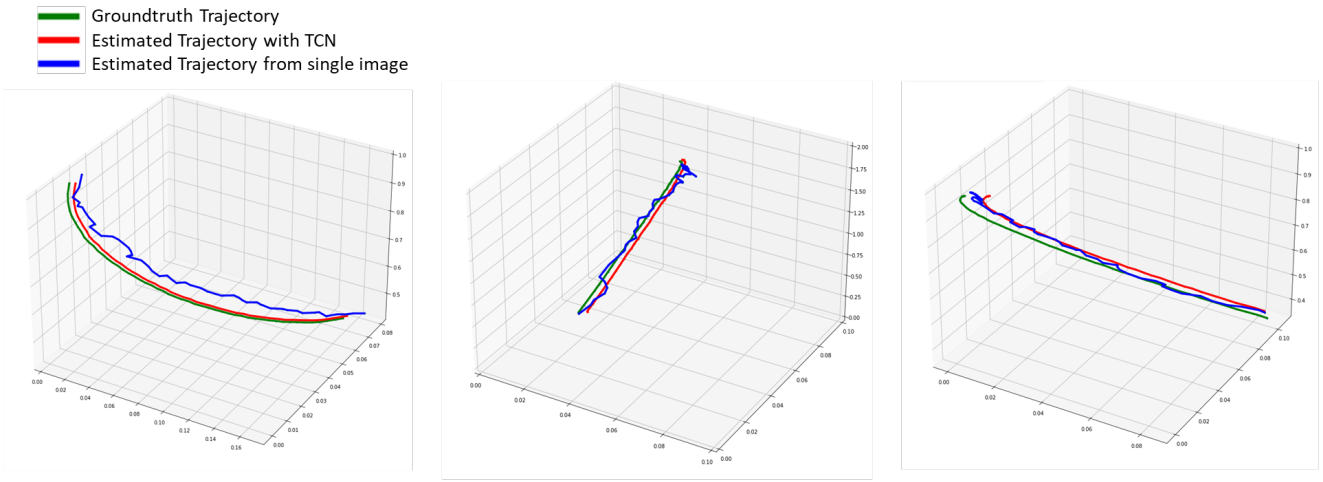

Figure 6. Three examples of groundtruth trajectories (in green) and the estimated 3D trajectories using TCN (in red) and the estimated 3D positions using direct regression (in blue).

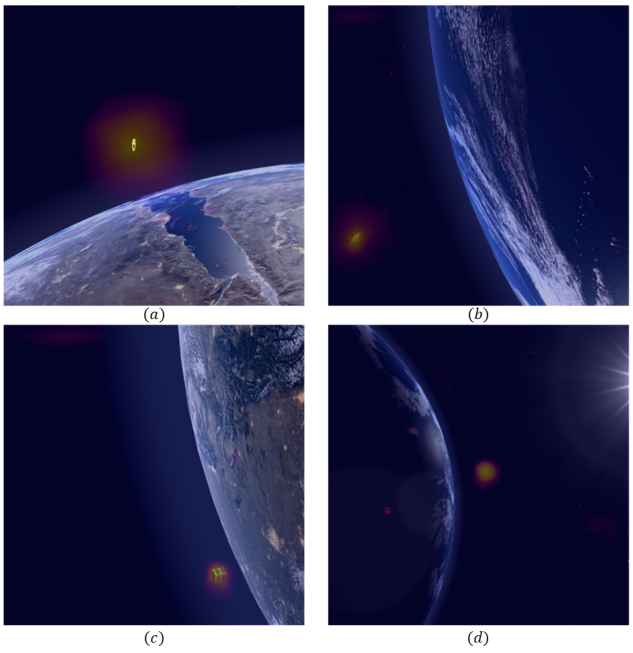

Figure 7. Visualization of the predicted spacecraft 2D location with the heat map overlaid on the input image. The red point ● is the ground truth 2D location, the green point ● is the predicted 2D location, in (a) Jason satellite successfully detected, (b) detected debris, (c) detected CubeSat, in (d) wrongly detected cube sat due to optical sensor sun flare (zooming in might be necessary).

For training the 3D trajectory estimation model $g(.)$, presented in Section 3.2, the model was trained with the sequence of 2D location of the ground truth as an input and 3D trajectories ground truth as an output. The 2D / 3D point sequences were normalized in order to have values in the range [0,1] for training the TCN model.

5.2. Implementation details

In order to detect the 2D coordinates of the space object present in the image, we train our 2D regression model presented in Figure 3 by passing the output of a single-channel normalized heatmap from U-Net to the DSNT layer [19] that outputs numerical coordinates, then we calculate the Euclidean distance[2] between the prediction μ and the ground truth p as

$$\mathcal{L}_{euc}(\mu, p) = \|p - \mu\|_2. \tag{5}$$

The estimated 2D coordinate sequences $\hat{\mathcal{X}}$ are passed through a TCN network in order to obtain the corresponding 3D coordinate sequences. By using a TCN network we preserve the temporal coherence present in the 2D sequences which leads, in turn, to improving the quality of the estimated 3D coordinates. The following parameters were used: kernel size $k = 6$; dilation rate $d \in \{1, 2, 4, 8\}$; Adaptive Moment Estimation (ADAM) optimizer with learning rate of 0.001; and 100 epochs.

To highlight the difference between using TCN for temporal consistency and direct 3D location regression, we trained another model similar to the one presented in Figure 3 (2D points regression) with adding an auxiliary brunch from the bottleneck of the ResNet encoder to directly regress the 3D location $\ell = (x, y, z)$ and jointly train the model to predict p and ℓ.

5.3. Results

We evaluate the obtained results qualitatively and quantitatively at the two levels, namely, (1) 2D location estima-

[2]No unit as 2D locations are normalized.

tion, and (2) 3D trajectory with respect to the camera.
With regards to the 2D location, we have used our proposed model presented in Figure 3. As a result we obtained an error $\mathcal{L}_{euc}$ of 0.48 for training and 0.62 for testing. Overall and in most of the cases, the obtained 2D coordinate detection from a single RGB image has a small error. Investigating the cases with high error, we found that those correspond to images generated under direct sun illumination and subject to lens flare. These challenging conditions contributed the most to wrongly detected 2D coordinates in these images as can be seen in Figure 7 (d). We note, nonetheless, that these frames do not appear continuously in a video. Using multiple frames for estimating the position is therefore a suitable strategy to mitigate errors coming from isolated frames.

In order to estimate the 3D trajectories of the spacecraft, we have lifted the 2D coordinates to 3D space using the proposed TCN based model presented in Section 3.2 and illustrated in Figure 4. Figure 6 shows a visual comparison between the 3D trajectory estimated with the proposed model (red) and the one estimated using direct 3D position regression from single images (blue). We note that our approach provides a smoother and a more temporally coherent trajectory. The overall quantitative result confirms the qualitative observation, with a mean squared error (MSE)[3] of 0.009 for training and 0.012 for testing as compared to 0.084 and 0.174 for training and testing, respectively, in the case of direct 3D position regression. The obtained results confirm a significant improvement as compared to directly estimating the 3D locations from the corresponding RGB images.

6. Conclusion

In this paper, we investigated the problem of spaceobject 3D trajectory estimation using only RGB information. We proposed a two-step approach decomposing the problem into: (1) a per-image 2D spacecraft detection; followed by (2) a per-sequence 3D trajectory estimation. Our experimental results showed that by properly leveraging temporal information, it is possible to simplify the problem and further increase accuracy as compared to a direct 3D position regression. Furthermore, we proposed a large realistic synthetic dataset that provides ground truth trajectories for three spacecrafts, under various sensing conditions. This dataset will be publicly shared with the research community in order to further the research on spacecraft trajectory estimation in the context of ADR.

References

[1] Nasa 3d resources. https://nasa3d.arc.nasa.gov/.

[2] S. Bai, J. Z. Kolter, and V. Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv:1803.01271*, 2018.

[3] Robin Biesbroek, Sarmad Aziz, Andrew Wolahan, Stefano Cipolla, Muriel Richard-Noca, and Luc Piguet. The clearspace-1 mission: Esa and clearspace team up to remove debris.

[4] KyungHyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *CoRR*, abs/1409.1259, 2014.

[5] Bruno Esmiller and Christophe Jacquelard. Cleanspace "small debris removal by laser illumination and complementary technologies". *AIP Conference Proceedings*, 1402:347–353, 11 2011.

[6] Albert Garcia, Mohamed Adel Musallam, Vincent Gaudillière, Enjie Ghorbel, Kassem Al Ismaeil, Marcos Damian Perez, and Djamila Aouada. Lspnet: A 2d localization-oriented spacecraft pose estimation neural network. *CoRR*, abs/2104.09248, 2021.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2016.

[8] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, Nov. 1997.

[9] Matthias Holschneider, Richard Kronland-Martinet, Jean Morlet, and Ph Tchamitchian. A real-time algorithm for signal analysis with the help of the wavelet transform. In *Wavelets*, pages 286–297. Springer, 1990.

[10] Yinlin Hu, Sebastien Speierer, Wenzel Jakob, Pascal Fua, and Mathieu Salzmann. Wide-depth-range 6d object pose estimation in space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15870–15879, June 2021.

[11] Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, and Koray Kavukcuoglu. Neural machine translation in linear time. *arXiv preprint arXiv:1610.10099*, 2016.

[12] M. Kisantal, S. Sharma, T. H. Park, D. Izzo, M. Martens, and S. D'Amico. Satellite pose estimation challenge: Dataset,competition design and results. *IEEE Trans. ON Aerospace AND Electronic Systems*, 2020.

[13] Eric Marchand, François Chaumette, Thomas Chabot, Keyvan Kanani, and Alexandre Pollini. Removedebris vision-based navigation preliminary results. In *IAC 2019-70th International Astronautical Congress*, pages 1–10, 2019.

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

[15] J. Martinez, R. Hossain, J. Romero, and J. J. Little. A simple yet effective baseline for 3d human pose estimation. In *ICCV*, 2017.

[16] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner, 2017.

[17] Mohamed Adel Musallam, Renato Baptista, Kassem Al Ismaeil, and Djamila Aouada. Temporal 3d human pose estimation for action recognition from arbitrary viewpoints.

[3] No unit as coordinates are normalized.

In *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 253–258. IEEE, 2019.

[18] Mohamed Adel Musallam, Kassem Al Ismaeil, Oyebade Oyedotun, Marcos Damian Perez, Michel Poucet, and Djamila Aouada. Spark: Spacecraft recognition leveraging knowledge of space environment, 2021.

[19] Aiden Nibali, Zhen He, Stuart Morgan, and Luke Prendergast. Numerical coordinate regression with convolutional neural networks, 2018.

[20] DE Olmos, TV Peters, J Naudet, CC Chitu, and K Sewerin. Android small active debris removal mission. In *Proceedings of the Fifth CEAS Air and Space conference, Delft, Netherlands*, pages 7–11, 2015.

[21] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[22] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. *arXiv preprint arXiv:1811.11742*, 2018.

[23] P. F Proença and Y. Gao. Deep learning for spacecraft pose estimation from photorealistic rendering. In *2020 IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2020.

[24] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[25] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[26] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points, 2019.