POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

A Point-based rendering approach for on-board instance segmentation of non-cooperative resident space objects

Laurea Magistrale in Space Engineering - Ingegneria Spaziale

Author: Mario Corradetti

Advisor: Prof. Pierluigi Di Lizia

Co-advisor: Niccolò Faraco

Academic year: 2021-2022

1. Introduction

Since the beginning of the space era Earth orbit is exponentially getting cluttered with human-made objects also known as Resident Space Objects (RSOs). According to the ESA's Space Environment Report the number of objects in orbit exceeds 32000 and the majority of these are placed in LEO and GEO orbits. The increasing number of launches and in-orbit permanence of space debris leads to a significant conjunction risk. If no arrangements are made for the disposal of the RSOs, the number of collision will rise. Long term, this could lead to "Kessler Syndorme" [2]. To forestall such an accumulation buildup, space agencies have begun to develop all kinds of solutions mitigate the problem. All the techniques developed to remove debris or mitigate their generation required the knowledge of the attitude of the target. An inspection mission of the target is performed in order to have information on the attitude of the RSOs. The selection of the inspection orbit is not a trivial task, and, due to the non-cooperative and uncontrolled nature of the RSOs, continuous correction of the trajectory are required to have a detailed description of the scene. Common techniques based on ground communication can not grant real-time decision making which is a fundamental aspect for the inspection. Thus, performing on-board decision could lead to an optimal solution to improve the perfromance of this kind of mission. In this work, the images taken from the cameras equipped by the chaser spacecraft are fed inside an on-board computer that uses machine learning models (ML) to perform features recognition of the RSOs. The generation of the images for the training is entrusted to an improved version of the software JINS [4]. The new version of the software allows to have the possibility to implement the attitude of the target and the chaser, and to reproduce the scenes more faithfully. The machine learning model analyzed in the work is called PointRend. Several scenario have been analyzed to assess the performance of the model and the results are compared to the one obtained using Mask R-CNN, the state-of-the-art model.

2. PointRend

Convolutional Neural Network (CNN) are the main techniques used for image segmentation. The output resolution of a CNN is a trade-off between computational cost and the amount of detail captured. The state-of-the-art model

Mask R-CNN has a resolution of 28 × 28. With this level of resolution, a regular grid will oversample smooth area while undersampling object boundaries. PointRend [3] overcomes this problem by adopting a different working principle with respect the standard head of Mask R-CNN. The main idea of the method is to view image segmentation as a rendering problem. The approach starts with a coarse prediction, then it is gradually upsampled by means of bilinear interpolation, thus refining the prediction only for a subset of points. The core of the model is the selection of points in the image: these points should be denser near high-frequency region, such as object boundaries. The way points are selected differs depending on whether one is in the inference or training phase.

During the inference the selection is inspired by the adaptive subdivision, a technique used in computer graphics. The points are located in regions where there is a high chance that the value is significantly different from its neighbors, leading to a good detection of the edges. For each region of interest, the output mask is iteratively generated in a coarse-to-fine fashion. In each iteration, PointRend performs an upsample of its previously predicted segmentation using bilinear interpolation. Then the first N points, between the most uncertain ones, are selected on the low resolution grid. The point-wise feature representation, constructed combining the fine-grained and coarse prediction features, is computed for each point. Once the feature representations are obtained a Multilayer Perceptron is applied to perform point-wise segmentation predictions.

For the training phase a non-iterative approach based on random sampling is used. N points are selected on region with high uncertainty but preserving a uniform coverage. The selection of the points is based on three principles: Overgeneration, Importance sampling and Coverage. While Mask R-CNN produces mask with resolution 28 × 28, PointRend predicts mask of resolution 224 × 224 with the same computational cost.

3. JINSv2 : a synthetic images generator

Machine Learning models need lots of data,or images as in this case, to be trained. Unfortunately, the number of satellite images is very limited, so generating a dataset for training is difficult, if not almost impossible. To overcome this type of problems, Jins Is Not a Simulator (JINS) software was developed by Faraco et al.[4]. The software allows to automatically generate annotated images by using a computer graphics software, called Blender. The software requires 3D models of satellites and the Earth to accurately reconstruct an inspection scenario. Fundamental advantage in using Blender is to be able to automate the scene generation process through Python scripts. JINS produces not only the scene but also binary masks of each component of the model, know as ground truth. These are needed for the generation of the annotations.

The most obvious changes in JINSv2 from its previous version is scene generation. In JINS there was no real scene construction: the satellite model was placed in the center of the Blender scene, the cameras were randomly generated, and the Earth in the background was generated by overlaying the satellite scene with one containing only the Earth. In the new version of JINS, to be as faithful as possible to an inspection mission, a single Blender scene containing the target, the chaser and the Earth has been used. The most difficult aspects of generating a scene containing these objects is the difference in order of magnitude between their dimension. This can be easily overcome thanks to Blender properties that allow to set the scale for the objects.

The initial configuration adopted for scene generation used the Earth Centered Inertial reference system, placing the Earth at the center and the satellites on user-selected orbits. This configuration causes problems with the mesh of the small objects placed far from the origin. The problem is solved by constructing the scene in the Local Vertical Local Horizontal frame with the target centered in the origin. This reference frame matches the one of the Clohessy-Wiltshire equations [1] that has been used by JINS to propagate the orbit of the chaser around the target. In the new version of JINS it is also possible to implement the attitude of the chaser and target to achieve a result even closer to reality. As a last update on the new version, a user interface with selection menu was implemented to make it easier for the user to generate the dataset.

Once the images and ground truths have been rendered, it is necessary to generate the annotations in order for the images to be used by PointRend for training. There are several ways to annotate a dataset, the one adopted in this work is the COCO format, which stands for Common Object in Context. The annotation are generated by a dedicated function called *satellitetoCOCO.py* and saved inside a *.json* format. The file contains all the information related to the images and the parts. The most important information are the coordinates of the bounding box and the list of points that describes the perimeter of the component.

4. Results

Unlike other machine learning models for which many online variants are available, only the original version developed by the Facebook AI Research is available for PointRend. The model has been trained on Google Colab with a GPU TESLA T4 16 GB and using as backbone the R50-FPN-3x. Hyperparameters like the learning rate or the batch size have been selected and kept fixed for all the work in order to have comparable results. Mask R-CNN was trained in parallel to PointRend in order to compare their accuracy. The metrics used to assess the performance of the models is the Average Precision (AP).

4.1. Training and testing on a single model

The first test performed on PointRend was to train and validate it only on one satellite model. The training dataset contains 360 images and the component considered are *antenna*, *body* and *solarPanel*. The time required for the training is about 40 minutes, the results are illustrated in Table 1.

AP values detection

AP	antenna	body	solarPanel
86.21	81.48	89.134	88.02

AP values segmentation

AP	antenna	body	solarPanel
76.50	72.21	78.85	78.42

Table 1: AP values PointRend.

The tables illustrates the overall AP and the APs for each class. An additional test was performed on this dataset. It was intended to check if the segmentation of the components may increase in the case where the Earth in the background was also segmented. The results of the test are shown in Table 2.

AP values detection

AP	antenna	body	solarPanel
86.580	78.63	83.10	84.95

AP values segmentation

AP	antenna	body	solarPanel
80.26	69.60	75.86	75.747

Table 2: AP values PointRend with *Earth* class.

The same test, with also the Earth segmented, has been performed using Mask R-CNN and the results are reported in Table 3.

AP values detection

AP	antenna	body	solarPanel
78.31	62.18	72.64	78.79

AP values segmentation

AP	antenna	body	solarPanel
72.00	59.04	62.96	68.407

Table 3: AP values Mask R-CNN with *Earth* class.

As confirmed by the theory the PointRend achieves an higher accuracy with respect to Mask R-CNN, not only in the segmentation but also in the detection of the components. This type of test is not only used to assess the performance of the model but could be useful in the case where during an inspection mission the geometry of the RSO is known and its model can be used for the generation of the scenes used in the training phase. Furthermore, the improvement in using PointRend over Mask R-CNN is significant.

In reality the geometry of the RSO to be observed is not always available. To verify the performance of PointRend in this condition, it has been trained on a dataset containing scenes of 5 satellite's models and tested on a sixth model that the algorithm has never seen. Different testing dataset are used in order to cover as many scenarios as possible and to test the robustness of the selected model.

4.2. Training with no Earth in background

The model was trained on images containing no Earth in the background. This was done to verify the performance of the model in the case where no disturbance element is present in the scene. Both PointRend and Mask R-CNN are trained on a dataset containing 750 images. The models have been tested not only on scene without the Earth in background but also on scene with the presence of the Earth since during a real mission the chaser will have periods when the Earth is visible. The results when models are tested with the Earth in the background are shown in Table 4.

PointRend AP segmentation

AP	antenna	body	solarPanel
23.23	20.10	23.52	49.30

Mask R-CNN AP segmentation

AP	antenna	body	solarPanel
24.19	16.02	29.57	51.19

Table 4: AP values of models trained without Earth in the background.

The higher AP values have been obtained with the Mask R-CNN model. This result may be caused by the intrinsic nature of PointRend, where more attention is paid to the edges of the object. The significant increase of details in the scene with respect to the ones used in the training phase leads the models to show uncertainties along the edges of the objects, and thus to perform incorrect predictions.

4.3. Training with Earth in background

When the Earth is not considered in the background PointRend becomes sensitive to the details in the scene causing a decrease in its performance. A new dataset of 750 scenes with the Earth in background has been generated. The training took about 53 minutes in contrast to the 40 minutes required for the previous test. The models have been tested on scene with the Earth in the background, the results are reported in Table 5.

PointRend AP segmentation

AP	antenna	body	solarPanel
63.23	37.81	50.32	66.91

Mask R-CNN AP segmentation

AP	antenna	body	solarPanel
60.20	37.04	47.87	67.07

Table 5: AP values of models trained with Earth in background.

The AP values have significantly increased with respect to the previous training. Now the models are trained on images full of details. Thus, there are no more accuracy problems as in the previous case.

Machine learning models need a large number of images for training to achieve optimal results. To further increase the number of images in the dataset without resorting to generating additional images, data augmentation techniques were implemented. There are different ways to artificially increase the number of images. The one implemented in the work are *rotation* and *flip*. This is done by a dedicated function called *Rotation.py* which takes images as input and perform three operations: flip and rotation of $\pm 90°$.

The new dataset contains 9000 images resulting in a significant increase in training time : $1h$ $43min$ for PointRend and $1h$ $30min$ for Mask R-CNN. The new trained models have been tested always on scene with the Earth in the background. The results are illustrated in Table 6.

PointRend AP segmentation

AP	antenna	body	solarPanel
71.28	56.53	67.20	71.11

Mask R-CNN AP segmentation

AP	antenna	body	solarPanel
64.28	40.51	53.90	69.72

Table 6: AP values of models trained with data augmentation.

Due to data augmentation the AP values for both PointRend and Mask R-CNN have increased. The improvement is most obvious for PointRend, which has gained 8 AP points. To show the difference in accuracy between the two methods, the inference was performed on a particular configuration of the testing satellite where both solar panels are placed in profile. The inferences are shown in Fig. 1.

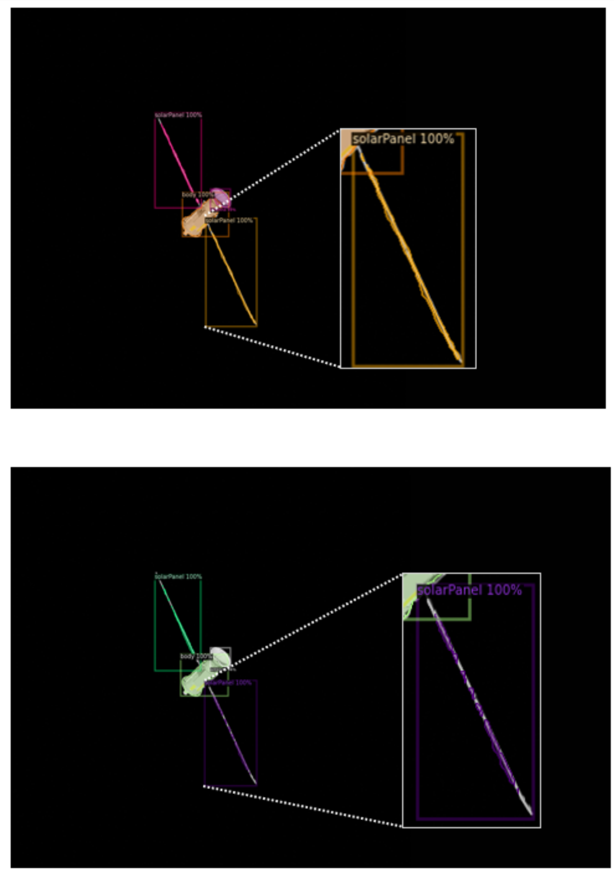

Figure 1: PointRend inference (top) and Mask R-CNN inference (bottom).

As confirmed by theory PointRend outputs masks with higher resolution than those provided by Mask R-CNN, increasing the accuracy especially along object's edges.

4.4. Training with noisy images

During a real mission, the images are affected by disturbances such as electrical noise and optical aberrations. A copy of the augmented dataset has been created and noise has been considered, in particular Poisson noise.

The models trained without noise have been tested on the noisy images to asses their performance. The results are shown in Table 7.

PointRend AP segmentation

AP	antenna	body	solarPanel
59.28	24.516	35.907	56.72

Mask R-CNN AP segmentation

AP	antenna	body	solarPanel
50.30	22.67	33.06	47.55

Table 7: AP values of models tested on noisy images.

The accuracy of both models has decreased but PointRend remains the most accurate one. Another training has been performed on the augmented noisy dataset. The models trained on noisy images present AP values 10 points higher with respect to the previous models when tested on scene with noise. The presence of noisy images inside the training dataset is necessary to make the model robust against camera noise.

4.5. Training with grayscale images

Many of the camera-equipped satellites acquire grayscale images. One of the main reasons is because they require little memory to save and therefore are easier to process or send to a ground station. The models have been tested on the augmented dataset converted into grayscale. The results are shown in Table 8.

PointRend AP segmentation

AP	antenna	body	solarPanel
50.15	19.621	36.38	62.66

Mask R-CNN AP segmentation

AP	antenna	body	solarPanel
56.07	24.20	37.50	65.13

Table 8: AP values of models tested on noisy images.

In the case of noisy images Mask R-CNN is more accurate than PointRend. Since both models are designed to take RGB or RGBA images as input, the lack of colors in the training dataset leads to reduced performance when tested on colored images. The use of grayscale images can be adopted as a form of data augmentation to train the model that will be used in the real inspection mission, or to generate a dataset if it is certain that the chaser only acquires images in grayscale.

4.6. Training with different learning rates

A different type of training for PointRend, has been performed. The variable changed is no more the training dataset but one of the most important hyperparameters: the learning rate. It is intended to see how the accuracy of PointRend changes as the value of the learning rate varies. Three out of the five learning rates selected are close to one used previously, while the other two are selected one order of magnitude higher or smaller. Using different values of learning rate also the training times change. The higher learning rates require shorter training times while for the low values of learning rate the training times increase. The different learning rate produces different result based on the scene used for the inference. When tested on images without noise in the background the learning rates with higher values achieve better accuracy than the ones with lower values. The higher learning rates succeed in apprehending the general features of the scene faster than a lower learning rate which learns slower by focusing more on details. This is confirmed by the fact that when the models are tested on noisy images the highest accuracy is achieved by the lowest learning rate. The value of the learning rate should be taken into account for training the model in the case of a real mission. A more widely used approach used by the latest developed machine learning models during training is to adopt a variable learning rate. Initially the value used is higher so that the model can better detect generic features, while toward the last iterations the value of the learning rate is decreased so that details are better recognized.

5. Conclusions

The PointRend model grants higher accuracy than the state-of-the-art model Mask R-CNN, with the same inference speed and required computational power. The fundamental part for obtaining good results is the choice of the training dataset and the setting of the hyperparameters. One of the major contribution for the increase in accuracy is given by the data augmentation techniques. The ease with which it was possible to generate so many datasets for training is thanks to the new version of the JINS software. The software now has several functionality such as propagating the relative orbit that one desire to represent, the satellite model can be directly chosen by the user via a menu and no longer by running several scripts, and most importantly it is possible to implement both chaser and target attitude.
The use of machine learning in orbit is receiving more and more attention. In order for the topic covered in this work to keep up with the continuing evolution of the computer vision world, several ideas for future development have been proposed. To achieve the same accuracy level of the last developed model, the backbone of the model could be changed with a vision transformer such as the SWIN transformer. Adopt a model that works with a lower inference time in order to achieve greater temporal consistency. Another idea consist in implementing the model on a Raspberry Pi board to asses its performance on real images.

References

[1] W. H. Clohessy and R. S. Wiltshire. Terminal guidance system for satellite rendezvous. *Journal of the Aerospace Sciences*, 27(9):653–658, 1960.

[2] ESA. Esa's annual space environment report. Technical report, ESA, 2022.

[3] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. *ArXiv:1912.08193*, 2019.

[4] Faraco Niccolò. Instance segmentation for features recognition on non-cooperative resident space objects. Master's thesis, Politecnico di Milano, 2020.