SATSPLATYOLO: 3D GAUSSIAN SPLATTING-BASED VIRTUAL OBJECT DETECTION ENSEMBLES FOR SATELLITE FEATURE RECOGNITION

V. Minh Nguyen, Emma Sandidge, Trupti Mahendrakar, and Ryan T. White*; NEural TransmissionS (NETS) Lab, Florida Institute of Technology, Melbourne, FL, *rwhite@fit.edu

Abstract. *On-orbit servicing, spacecraft inspection, and active debris removal missions require precise operations near non-cooperative space objects, posing risks for manned missions and challenges for ground control. We propose an autonomous method for mapping and detecting satellite components, using 3D Gaussian splatting for learning a 3D satellite representation, rendering virtual views, and ensembling a YOLOv5 object detector over those views. Our low-compute pipeline can be run onboard, enabling real-time characterization to facilitate autonomous rendezvous and proximity operations.*

Introduction. The rising threat of space debris necessitates autonomous rendezvous and proximity operations (RPO) for on-orbit servicing (OOS) and active debris removal (ADR). These operations aim to manage or remove large defunct satellites, which form the majority of space debris. The complexity and delays in ground-based control demand full autonomy for these missions. This article outlines an approach to characterize and detect satellite features to enable autonomous RPO. The method: (1) construct a 3D satellite scene using 3D Gaussian Splatting based on the images captured along an inspection orbit, (2) generate synthetic images from various perspectives, and (3) ensemble YOLOv5 across these images for precise satellite feature recognition. The system is designed for low-power spaceflight hardware and includes innovations like systematic 3D rendering, a rendering-based detection ensemble, and realistic hardware-in-the-loop experiments.

Methods. This section provides further details on the methods used and developed herein.

Datasets. We use two datasets. The Web Satellite Dataset (WSD) for training YOLOv5 where solar panels, antennas, thruster and body are annotated. The hardware-in-the-loop (HIL) dataset contains images of a satellite mock-up captured on the ORION testbed[1] under realistic lighting and motion conditions. The satellite mock-up yaws 360° to simulate a station keeping maneuver. The chaser captures images positioned 5 ft away at 5° increments. Chroma key compositing with a green screen is used to remove background artifacts.

3D Gaussian Splatting (3DGS). 3DGS[3] learns a 3D representation of a scene using 3D Gaussian points using several images from different viewing angles. It initializes an initial 3D point-cloud representation of a scene based on given input images using Structure-from-Motion. 3DGS treats each point as a 3D Gaussian distribution with a mean equal to the point's position and a covariance matrix that determines the shape and orientation of the distribution. Each Gaussian density is projected onto the desired view frame to create "splats" of visual primitives like colors and textures. The contributions of the splats are accumulated to render the final image. The shape and size parameters of the Gaussians are optimized by constrained stochastic gradient descent and the number of points is manipulated by an adaptive density control technique periodically during training.

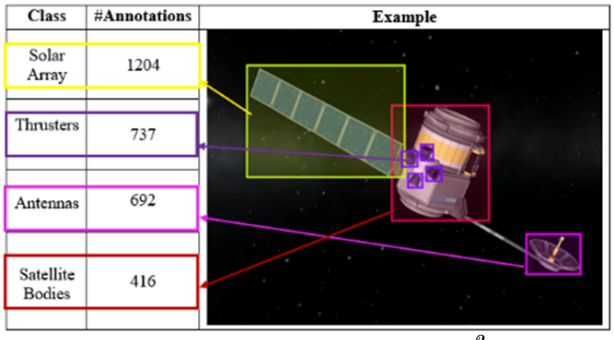

Figure 1. WSD Annotations[2]

Camera Generation. Given n initial camera poses represented by their transformation matrices, we will generate novel camera viewpoints around a synthetic rendezvous path. We estimate a reference point $\mathbf{p} \in \mathbb{R}^3$ that all initial cameras are roughly pointed towards (the camera attention center) as a nearest point problem formulated as a least-squares problem:[4]

$$\min_{\mathbf{p}} \|\mathbf{A}\mathbf{x} - \mathbf{C}\|_2^2, \tag{1}$$

where $\mathbf{C} \in \mathbb{R}^{3n\times 1}$ is a column vector that concatenates all camera positions $[\mathbf{c}_1^\top, \mathbf{c}_2^\top, \ldots, \mathbf{c}_n^\top]^\top$, $\mathbf{x} = [\mathbf{p}^\top, a_1, a_2, \ldots, a_n]^\top$ is the solution vector containing $\mathbf{p}$, and $\mathbf{A} \in \mathbb{R}^{3n\times(n+3)}$ is a specially constructed block matrix encoding the forward vectors of each camera.

We generate novel poses per initial camera via several methods: (1) circular of radius r in the plane orthogonal to $\mathbf{p} - \mathbf{c}_i$, centered at the original $\mathbf{c}_i$, sampled randomly or uniformly, and (2) spherical on a sphere of radius r centered at the original $\mathbf{c}_i$, sampled randomly or semi-uniformly using a Fibonacci lattice.[5] See Figure 2.

YOLOv5 Ensembling. YOLOv5[6] is a single-stage object detector selected as it shows the most success at unknown satellite component detection,[2] though it is imperfect. Further, we choose the "small" architecture, as it can run at sufficient framerates on current spaceflight hardware. We train YOLOv5 on the WSD.[2] We test on a real-life mock-up of a satellite in two cases: (1) known satellite where the object detector has seen some views of the mock-up during training, and (2) unknown satellite where the object detector has never seen the mock-up during training.

Table 1. SatSplatYOLO Results for Camera Sampling Techniques

		Known Satellite			Unknown Satellite		
Cameras	Radius	Precision	Recall	mAP@0.5	Precision	Recall	mAP@0.5
Circular (uniform)	0.5	0.728	0.704	0.701	0.455	**0.400**	0.421
	1.0	0.633	0.596	0.603	0.435	0.339	0.386
Spherical (Fibonacci)	0.5	0.711	0.688	0.684	0.475	**0.400**	0.433
	1.0	0.740	0.604	0.652	**0.590**	0.384	**0.485**
Ground Truth	N/A	**0.770**	**0.728**	**0.758**	0.544	0.384	0.432

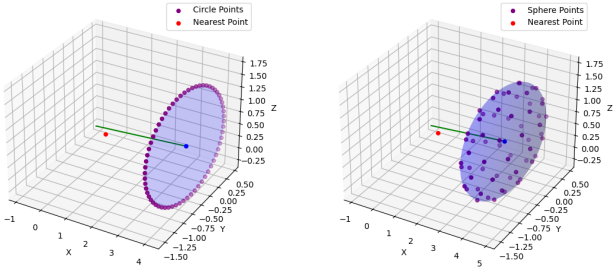

Figure 2. Cameras perturbed about from the original camera position circularly and spherically

We propose a novel ensembling technique to improve performance. For each ground truth camera view, we generate 64 renders on which YOLOv5 detects satellite components and we ensemble the detections to enhance the original detections. The synthetic detections experience an consistent translation offset due to the approximation of the nearest point algorithm (Figure 6). We perform a translation correction by mapping the mean bounding box from the detection groups to high-confidence (> 0.95) YOLOv5 predictions on the original images. We then group shifted detections the renders with high intersection over union (IoU), assign group class prediction based on purity (the fraction of classifications agreeing with the majority), discard small and low-purity groups, and merge redundant groups based on IoU.

If a detection group coincides in location with an original detection, the confidence is max(conf, purity), bounding box is updated to a weighted average (by confidence) of the original and group mean bounding boxes, and assign the group-predicted class. If a detection group does not coincide with the location of an original detection, we make a new prediction to correct original model's likely false negative. We further remove ground truth predictions that do not coincide with a detection group, suppressing false positives.

Results. 3DGS is trained to produce a 3D representation of the scene by the approach of Nguyen et al.[7] The resulting quality of that work is replicated, achieving 0.9213 SSIM, 25.52 PSNR, and 0.0796 LPIPS.

SatSplatYOLO results are shown in Table 1 for some experimental cases. The left half of the table is for a known satellite and the right half is for an unknown satellite. We try two camera generation techniques, each with two radius values. We document the precision, recall, mAP@0.5 for each model. SatSplatYOLO improves results for unknown satellites significantly, but performance on known satellites is better without SatSplatYOLO.

Conclusion. The proposed a method SatSplatYOLO detects components of satellites robustly by learning a 3D representation of the model based on image data captured through one inspection orbit, rendering novel views of the satellite, and ensembling YOLOv5 over them. The computational costs are sufficiently low to feasibly run on near-term spaceflight hardware.

Further study of the positioning of the synthetic cameras, the number of renders that are required, and details of the inspection orbit could improve the accuracy and run times of the method. Implementation on edge hardware and software optimizations could further decrease computational complexity.

This method also provides a first step to reliable pose estimation of unknown non-cooperative satellites. Estimating the pose of the spacecraft this way provides more confidence than most methods in the literature as it provides a holistic understanding of the scenario.

References.

[1] M. Wilde, B. Kaplinger, T. Go, H. Gutierrez, and D. Kirk, "ORION: A simulation environment for spacecraft formation flight, capture, and orbital robotics," in *2016 IEEE Aerospace Conference*, pp. 1–14, Mar. 2016.

[2] T. Mahendrakar, R. T. White, M. Tiwari, and M. Wilde, "Unknown non-cooperative spacecraft characterization with lightweight convolutional neural networks," *AIAA JAIS*, vol. 21, no. 5, pp. 455–460, 2024.

[3] B. Kerbl, G. Kopanas, T. Leimkuehler, and G. Drettakis, "3D Gaussian Splatting for Real-Time Radiance Field Rendering," *ACM Transactions on Graphics*, vol. 42, pp. 139:1–139:14, July 2023.

[4] L. Han and J. C. Bancroft, "Nearest approaches to multiple lines in n-dimensional space," *CREWES Research Report*, vol. 22, pp. 32.1–32.17, 2010.

[5] Á. González, "Measurement of areas on a sphere using fibonacci and latitude–longitude lattices," *Mathematical Geosciences*, vol. 42, pp. 49–64, Jan 2010.

[6] G. Jocher, "YOLOv5 by Ultralytics," May 2020.

[7] V. M. Nguyen, E. Sandidge, T. Mahendrakar, and R. T. White, "Characterizing satellite geometry via accelerated 3d gaussian splatting," *Aerospace*, vol. 11, no. 3, 2024.

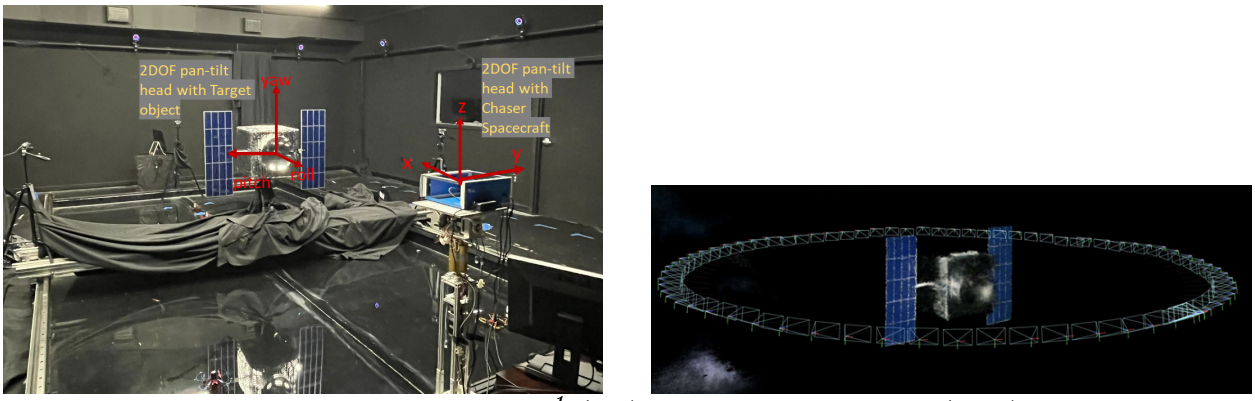

Figure 3. ORION Testbed[1] (left) and image capture path (right)

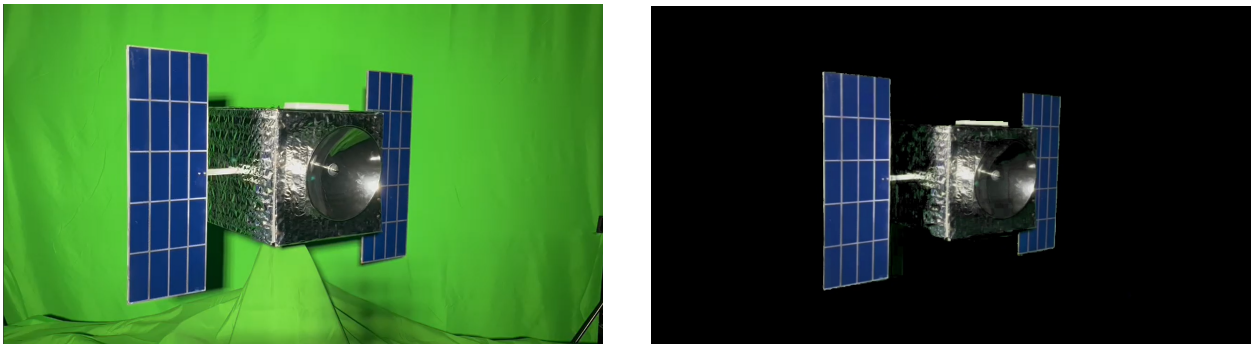

Figure 4. Green screen-based chroma keying to remove the background

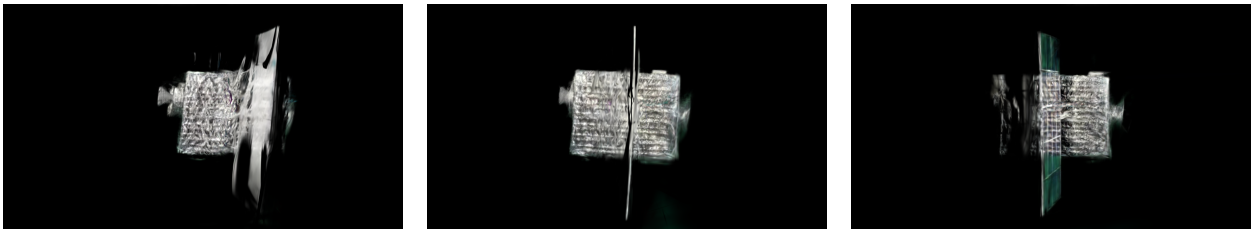

Figure 5. High quality render of true camera (center) and two generated cameras

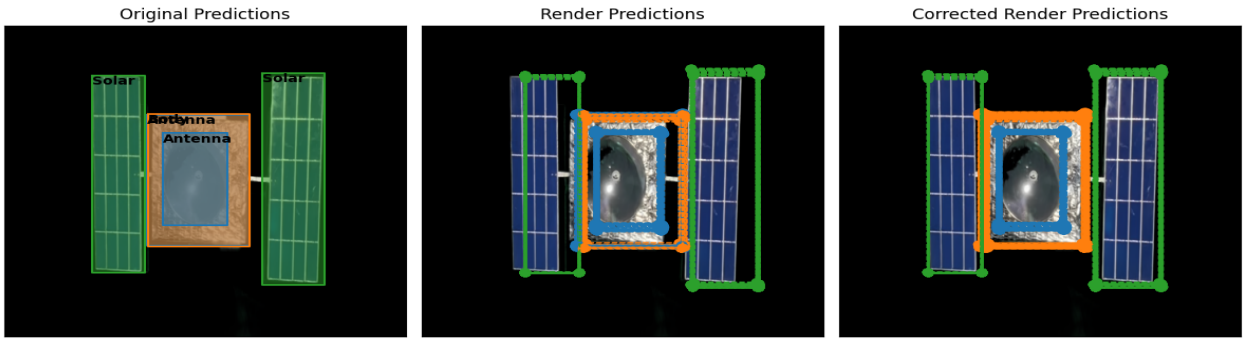

Figure 6. YOLOv5 inferences on the original image (left: note a mistaken antenna prediction coupled with the correct body prediction), the 3DGS renders (center: note the translation errors), and final SatSplatYOLO predictions (right).