Improved 3D Gaussian Splatting of Unknown Spacecraft Structure Using Space Environment Illumination Knowledge

Tae Ha Park[1] Simone D'Amico[2]

***Abstract*— This work presents a novel pipeline to recover the 3D structure of an unknown target spacecraft from a sequence of images captured during Rendezvous and Proximity Operations (RPO) in space. The target's geometry and appearance are represented as a 3D Gaussian Splatting (3DGS) model. However, learning 3DGS requires static scenes, an assumption in contrast to dynamic lighting conditions encountered in spaceborne imagery. The trained 3DGS model can also be used for camera pose estimation through photometric optimization. Therefore, in addition to recovering a geometrically accurate 3DGS model, the photometric accuracy of the rendered images is imperative to downstream pose estimation tasks during the RPO process. This work proposes to incorporate the prior knowledge of the Sun's position, estimated and maintained by the servicer spacecraft, into the training pipeline for improved photometric quality of 3DGS rasterization. Experimental studies demonstrate the effectiveness of the proposed solution, as 3DGS models trained on a sequence of images learn to adapt to rapidly changing illumination conditions in space and reflect global shadowing and self-occlusion.**

I. Introduction

Autonomous Guidance, Navigation and Control (GN&C) with respect to a non-cooperative spaceborne target is a core technology indispensable to various Rendezvous and Proximity Operation (RPO) missions, such as on-orbit servicing [1], [2] and active debris removal [3]. In particular, monocular vision-based approaches have received widespread attention from the community due to their reliance on a low Size, Weight, Power and Cost (SWaP-C) optical sensor suitable for on-board satellite avionics. In this case, the servicer spacecraft must be able to estimate in real-time the position and orientation (i.e., *pose*) of the target object with respect to the servicer given a sequence of images captured during the RPO process.

A predominant body of existing literature on the topic of vision-based spacecraft pose estimation assumes known targets whose 3D structure and mass properties are available a priori to the operators [4]–[8]. In this setting, many works adopted Machine Learning (ML) and Neural Networks (NN) to perform pose estimation from a single image [9]–[20] or continuous tracking with a navigation filter in the loop given a sequence of images during RPO [21]–[24]. Due to the inaccessibility of space, these methods primarily leverage computer-rendered synthetic images for training, available via various open-source datasets such as SPEED [15], SPEED+ [25], URSO [17] and SPARK [26]. Moreover, in order to prevent overfitting to the synthetic imagery and bridge the performance gap when tested on previously unknown spaceborne imagery, the robustness and generalizability of the CNN models are validated on-ground using hardware-in-the-loop testbeds capable of emulating space-like illumination conditions [25], [27]–[29].

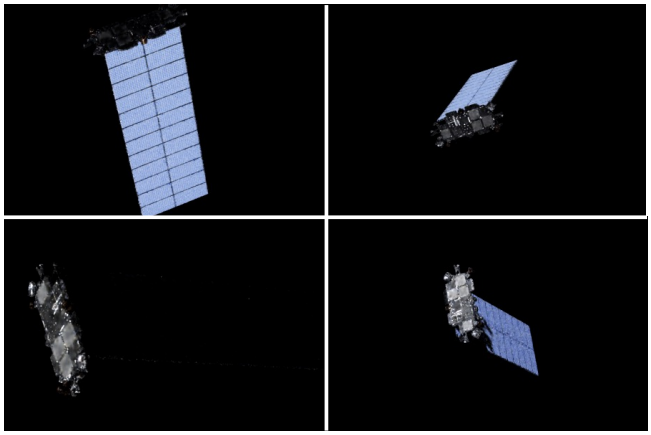

Fig. 1: The rendered images of a previously unknown satellite generated by a 3DGS model trained on a sequence of images captured during a simulated RPO scenario. Compared with vanilla 3DGS, the proposed training pipeline explicitly incorporates knowledge of the Sun vector maintained by the onboard AOCS module through shadow splatting, significantly improving the photometric accuracy of the rendered images.

In order to support a full range of future RPO missions including debris removal, the assumption of known targets must be relaxed. This means that the servicer spacecraft must be capable of not only predicting the target's motion but also reasoning about the target's 3D geometry and inertial properties simultaneously. If these are all done in an online sequence, the problem is akin to the well-known Simultaneous Localization and Mapping (SLAM) [30] in modern robotics.

Current state-of-the-art approaches to visual SLAM adopt radiance field models such as Neural Radiance Fields (NeRF) [31] and 3D Gaussian Splatting (3DGS) [32] as a unified representation of an unknown scene. Radiance field models enable high-fidelity novel view synthesis by learning to map the camera's position and viewing direction to view-dependent color and density of the scene via differentiable volumetric rendering. While the original designs of NeRF and 3DGS required known pose labels acquired from the initial Structure-from-Motion (SfM) stage [33], [34], recent

T. H. Park is supported by the Technological Innovation R&D Program (RS-2024-00513881) funded by the Ministry of SMEs and Startups, Republic of Korea.

[1]Nara Space Technology Inc., Seoul, Republic of Korea. Email: thpark@naraspace.com

[2]Dept. of Aeronautics & Astronautics, Stanford University, Stanford, CA 94305, USA. Email: damicos@stanford.edu

works have shown that these models can be incorporated into the SLAM framework and learned without prior knowledge of the ground-truth camera poses [35]–[38]. Specifically, the poses can be recovered by setting them as independent variables in the optimization of the photometric loss for rendered images while fixing the 3DGS parameters. Naturally, the photometric accuracy of differentiable rendering is crucial to the recovery of a scene or object in a visual SLAM scenario.

However, original NeRF and 3DGS pipelines assume controlled static scenes with consistent illumination conditions, as the scene colors are represented as spherical harmonic coefficients dependent only on the camera's viewing directions. Such a setting is not suitable for a space environment, where directional light of the Sun can cause significant and rapidly varying occlusion on the target RSO as its orbital position and attitude change over time. A conventional approach to learning NeRF or 3DGS in this so-called "in-the-wild" setting is to assign a learnable appearance embedding vector to each training image, effectively capturing the transient components of the scene (e.g., weather, time, etc.) into the low-dimensional latent space [39], [40]. However, there are two issues with the direct application of in-the-wild methods to RPO scenarios. First, the correct appearance embedding for the new images must be solved at test-time via multiple rounds of computationally expensive differentiable rasterization. Second, the images acquired during RPO are likely to be quickly discarded due to limited on-board computational power, which prevents the proper training of per-image appearance embeddings and the downstream 3DGS pipeline.

In light of these challenges, this paper proposes a novel 3DGS pipeline that can recover the geometry and appearance of an unknown target RSO despite rapidly changing illumination conditions in space. Specifically, as opposed to conventional in-the-wild methods which require test-time optimization of per-image appearance embeddings, this work proposes to leverage the prior information that every self-navigating satellite possesses—the Sun vector. Considering that the Sun is the predominant source of illumination in space and that the spacecraft maintains an accurate estimate of its direction for its own Attitude and Orbit Control System (AOCS), incorporating its knowledge into the in-the-wild 3DGS pipeline results in an improved performance without having to optimize the embedding. Specifically, inspired by recent literature on relightable Gaussian Splatting [41], the visibility of each Gaussian due to direct lighting is explicitly obtained by splatting the Gaussians towards the known lighting direction. The resulting visibility/shadow information is provided as an additional input to the pipeline, accelerating the photometrically accurate recovery of the observed images. As previewed in Fig. 1, experiments on high-fidelity synthetic images rendered during representative RPO scenarios suggest that the proposed pipeline can recover the geometrically and photometrically accurate 3DGS models that capture global illumination effects and self-occlusion.

II. Related Work

Spaceborne Visual SLAM. Early studies for vision-based navigation about an unknown target RSO leveraged existing feature detection algorithms and SLAM frameworks. For example, Tweddle [42] utilized SURF features [43] and iSAM [44] to track the motion of a SPHERES satellite within the controlled environment inside the International Space Station (ISS). Dor & Tsiotras [45] applied ORB-SLAM [30] to real images of the Hubble Space Telescope (HST) captured during the NASA STS-125 Servicing Mission 4, but they had to recover its scale using the ground-truth model of HST.

More recent works investigate the applicability of ML to this problem. While not exactly a SLAM method, Park & D'Amico [46] proposed an *offline* training of a Convolutional Neural Network (CNN) to reconstruct an unknown spacecraft structure as a set of superquadric primitives [47] from single images. The model is trained on the SPE3R dataset comprising high-fidelity synthetic images of 64 different spacecraft models. Naturally, the resulting model overfits to SPE3R and lacks zero-shot capability when subject to images of a completely new spacecraft. Moreover, the predicted orientation often exhibits large errors due to the symmetric nature of man-made spacecraft. Follow-up works [48], [49] attempted to address the aforementioned shortcomings with architectural innovation and accounting of pose ambiguities. Specifically, Bates & D'Amico [49] proposed an ambiguity-free solution by redefining the target's body axes to be parallel to the camera axes for each training instance, effectively coupling the 3D shape and orientation estimations. This approach resulted in an improved 3D reconstruction quality on unseen spacecraft models without explicit consideration of their geometric symmetry.

On the other hand, a number of works attempted to leverage radiance field models for vision-based spaceborne navigation about an unknown RSO. Caruso et al. [50] and Mergy et al. [51] implemented various NeRF models [52], [53] for 3D reconstruction of an unknown spacecraft from a set of synthetic and hardware images. However, they still rely on COLMAP [33], [34] to recover the initial pose labels. Recently, Barad et al. [36] proposed an object-centric SLAM pipeline based on 3DGS representation of the target, whereby a small window of keyframes is maintained to train both 3DGS parameters and camera poses in an interchanging fashion. Tested on sequences of synthetic images rendered during the simulations of a spiral trajectory about 10 different spacecraft models, the experiment showed promising results but failed to account for photometric variation due to dynamic illumination conditions.

Radiance Field Models in Dynamic Scenes. As noted earlier, vanilla radiance field models assume static scenes and thus have difficulty by design accounting for different dynamic elements such as lighting conditions, weather, and transient objects such as moving pedestrians. NeRF-W [39] is a pioneering approach that abstracts away per-image photometric and environmental variations and transient components into separate learnable low-dimensional latent spaces. The core principle of disentangling static and dynamic components is adopted by many NeRF- and 3DGS-based works that followed [40], [54]–[58]. However, as

explained earlier, conventional in-the-wild methods are not compatible with RPO scenarios in space due to the need for finding the optimal appearance latent vector for each evaluation image.

III. Methodology

The proposed pipeline aims to perform *online* training of a 3DGS model that can render geometrically and photometrically accurate images of the target RSO given known directions to the Sun as estimated by the servicer spacecraft. As noted in Sec. I, the photometric accuracy of 3DGS models is extremely important since the unknown camera poses can be recovered by optimization using the photometric loss function on the rendered images obtained via differentiable rendering. Therefore, in order to limit the scope of this paper to the photometric accuracy of 3DGS rasterization in space-borne RPO scenarios, a number of favorable assumptions are made throughout this paper, including known pose labels and availability of binary masks to remove the background. Moreover, instead of relying on COLMAP [33], [34], the initial Gaussian positions are uniformly sampled from the surfaces of the ground-truth RSO 3D models. The ways to relieve these ideal assumptions are detailed later in Sec. VI.

Preliminary: 3D Gaussian Splatting. 3D Gaussian Splatting (3DGS) [32] represents the target or scene as a sparse set of anisotropic 3D Gaussians, where the i^{th} Gaussian $\mathcal{G}_i$ is defined by a 3D covariance matrix $\boldsymbol{\Sigma}_i \in \mathbb{S}^3_+$ centered at mean $\boldsymbol{\mu}_i \in \mathbb{R}^3$:

$$\mathcal{G}_i(\boldsymbol{x}) = \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)\right] \tag{1}$$

Note that its mean and covariance respectively represent the position and 3D ellipsoidal shape of $\mathcal{G}_i$ in the RSO's body frame. They are additionally equipped with optical properties including view-dependent RGB color $\boldsymbol{c}_i$ represented as Spherical Harmonic (SH) coefficients and opacity $o_i \in [0,1]$.

The aforementioned geometric and optical parameters of 3D Gaussians are learned using a differentiable rasterizer and gradient-based optimization. During forward propagation, the 3D Gaussians are "splatted" onto 2D image space via projective transformation. In order to make the process differentiable, the splatted 2D covariance $\boldsymbol{\Sigma}^{\text{S}} \in \mathbb{S}^2_+$ is obtained by ignoring the third row and column after an affine approximation of the projective transformation [59] given as

$$\boldsymbol{\Sigma}^{\text{S}} = \left(\boldsymbol{J}\boldsymbol{W}\boldsymbol{\Sigma}\boldsymbol{W}^\top\boldsymbol{J}^\top\right)_{1:2,1:2} \tag{2}$$

where $\boldsymbol{J}$ is the Jacobian of the projective transformation and $\boldsymbol{W}$ is the viewing matrix. This approximation allows for numerical back-propagation of the photometric loss functions to the Gaussian parameters. Note that directly optimizing the covariance matrix $\boldsymbol{\Sigma}$ does not necessarily preserve its positive semi-definiteness. Therefore, $\boldsymbol{\Sigma}$ is further decomposed as $\boldsymbol{\Sigma} = \boldsymbol{R}\boldsymbol{S}\boldsymbol{S}^\top\boldsymbol{R}^\top$ with an identity scale matrix $\boldsymbol{S}$ and a rotation matrix $\boldsymbol{R}$, which is in turn represented as a unit-norm quaternion vector.

Once the Gaussians are splatted onto the 2D image space, the color $C(\boldsymbol{x})$ for each pixel $\boldsymbol{x}$ is rendered by first sorting the splatted Gaussians along the depth and then performing the volumetric α-blending of $\mathcal{N}$ ordered Gaussians as

$$C(\boldsymbol{x}) = \sum_{i\in\mathcal{N}} \boldsymbol{c}_i \alpha_i \prod_{j=1}^{i-1}(1-\alpha_j), \quad \alpha_i = o_i \mathcal{G}_i^{\text{S}}(\boldsymbol{x}) \tag{3}$$

where $\mathcal{G}_i^{\text{S}}(\boldsymbol{x})$ denotes the splatted i^{th} Gaussian evaluated at the pixel $\boldsymbol{x}$.

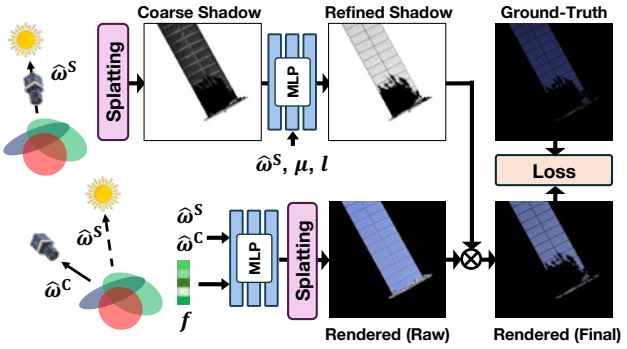

Fig. 2: Proposed 3DGS pipeline with shadow splatting using the known Sun vector.

Leveraging the Sun Vector. As noted in Sec. I, conventional in-the-wild methods are incompatible with spaceborne RPO scenarios due to constrained on-board computational resources which limit test-time optimization via differentiable rasterization of per-image appearance embeddings. Therefore, this work proposes to leverage the known Sun vector maintained by the servicer's AOCS.

The overall pipeline is illustrated in Fig. 2. The core of the pipeline is the rasterization of appearance-conditioned Gaussians. Specifically, instead of directly learning the view-dependent color as done in the original 3DGS, each i^{th} Gaussian is equipped with a trainable feature vector $\boldsymbol{f}_i \in \mathbb{R}^{72}$ that captures varying colors of Gaussians under different environments. In conventional in-the-wild methods such as WildGaussians [40], each j^{th} image is assigned a learnable embedding vector $\boldsymbol{e}_j$, so that the view-dependent color (i.e., SH coefficients) of that Gaussian is obtained by forward propagation of an appearance MLP $\boldsymbol{\Phi}$ (3×256-D hidden layers) given as $\boldsymbol{c}_i = \boldsymbol{\Phi}(\boldsymbol{f}_i, \boldsymbol{e}_j)$.

The proposed pipeline instead replaces the per-image embedding $\boldsymbol{e}_j$ with the Sun vector $\hat{\boldsymbol{\omega}}_j^{\text{S}}$ and the camera viewing vector $\hat{\boldsymbol{\omega}}_{ij}^{\text{C}}$ from the i^{th} Gaussian. Following the practice of NeRF [31], these unit vectors are first passed through the positional encoding function $\gamma_L(\cdot)$ defined as

$$\gamma_L(p) = [\ \sin(2^0\pi p)\ \ \cos(2^0\pi p)\ \ \sin(2^1\pi p)\ \ \cos(2^1\pi p)\ \cdots \\ \sin(2^{L-1}\pi p)\ \ \cos(2^{L-1}\pi p)\ ] \tag{4}$$

Then, the final view-dependent color of the i^{th} Gaussian given known Sun and view vectors are obtained via

$$\boldsymbol{c}_i = \boldsymbol{\Phi}\left(\boldsymbol{f}_i, \gamma(\hat{\boldsymbol{\omega}}_j^{\text{S}}), \gamma(\hat{\boldsymbol{\omega}}_{ij}^{\text{C}})\right) \tag{5}$$

and used in the 3DGS rasterization process.

The final rendered image $\hat{\boldsymbol{C}}_j$ and the observed ground-truth image $\boldsymbol{C}_j$ are used to compute the classical loss

function for 3DGS

$$\mathcal{L}_{\text{3DGS}} = (1 - \lambda_{\text{SSIM}})\mathcal{L}_1 + \lambda_{\text{SSIM}}\mathcal{L}_{\text{SSIM}} \tag{6}$$

where $\mathcal{L}_1$ is the mean pixel-wise ℓ_1-distance, $\mathcal{L}_{\text{SSIM}}$ computes the Structural Similarity Index Measure (SSIM) [60], and $\lambda_{\text{SSIM}} = 0.2$ is the weight hyperparameter. Inspired by GS-SLAM [38], an isotropic scale regularization is also considered, defined as

$$\mathcal{L}_{\text{iso}} = \sum_{i=1}^{|\mathcal{G}|} \|\boldsymbol{s}_i - \tilde{s}_i \cdot \mathbf{1}\|_1 \tag{7}$$

where $\boldsymbol{s}_i$ is the scale parameters of the i^{th} 3D Gaussian, and $\tilde{s}_i$ is its mean value. $\mathcal{L}_{\text{iso}}$ encourages sphericality and discourages elongated or otherwise irregular Gaussian shapes. While the original intention was to facilitate continuous tracking in SLAM, it has an added benefit when coupled with the shadow splatting—by discouraging lopsided geometries, the rendered images result in crisper and sharper shadows as expected in spaceborne imagery.

The final training loss is the combination of Eqs. 6 and 7

$$\mathcal{L} = \mathcal{L}_{\text{3DGS}} + \lambda_{\text{iso}}\mathcal{L}_{\text{iso}} \tag{8}$$

Shadow Splatting. In space, directional sunlight often casts extreme shadows on RSO due to self-occlusion. Such a phenomenon can be difficult for an MLP to learn without any regularization. Therefore, the proposed pipeline adopts the shadow splatting mechanism of GS^3 [41] in order to inform each Gaussian whether it is lit or shadowed given the Sun's relative position. This is cleverly done by reusing the existing 3DGS rasterization framework. As visualized in Fig. 2, the Gaussians are first splatted towards the light source for the j^{th} image, which is equivalent to placing the camera center at the source with its boresight aligned with $-\hat{\boldsymbol{\omega}}_j^{\text{S}}$. Considering the distance to the Sun, the camera is instead "placed" at the original distance but along the Sun vector. Once splatted, the cumulative opacity of the Gaussians sorted along the ray is used to determine the visibility (i.e., shadow) of each i^{th} Gaussian (V_i). The shadow value of each Gaussian is further refined with a small MLP $\boldsymbol{\Psi}$ (3 $\times$ 32-D hidden layers) via

$$V_i' = \boldsymbol{\Psi}(V_i, \gamma(\hat{\boldsymbol{\omega}}_j^{\text{S}}) \mid \gamma(\boldsymbol{\mu}_i), \boldsymbol{l}_i) \tag{9}$$

where $\boldsymbol{\mu}_i$ is the i^{th} Gaussian center location in the object frame and $\boldsymbol{l}_i \in \mathbb{R}^6$ is a trainable shadow latent vector.

The refined shadow values ($\boldsymbol{V}'$) are finally splatted onto a shadow image and pixel-wise multiplied to the rendered image. The resulting image is used to compute the loss function in Eq. 6.

Keyframe Management Strategy. Given the limited on-board processing capability and staying in tune with previous works [36], [38], a small window of maximum $\mathcal{W}$ keyframes (i.e., images) is maintained. Existing SLAM methods select keyframes that avoid unnecessary redundancy while achieving significant baselines between subsequent keyframes with high covisibility of the shared observations [30]. However, considering the scope of this paper and the assumption of known pose labels, the proposed 3DGS pipeline adopts a simple heuristic analogous to that of DSO [61] for keyframe management when training on a sequence of images collected during RPO. Let the window consist of keyframes $\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_n$ ordered in a reverse chronological order.

1) The next frame $\mathcal{I}$ is added if its camera view vector $\hat{\boldsymbol{\omega}}^{\text{C}}$ is more than $\theta_{\text{VIEW}} = 10^\circ$ off from that of the last frame $\mathcal{I}_1$, i.e.,
$$\hat{\boldsymbol{\omega}}^{\text{C}} \cdot \hat{\boldsymbol{\omega}}_1^{\text{C}} < \cos\theta_{\text{VIEW}} \tag{10}$$
2) The last two frames $\mathcal{I}_1, \mathcal{I}_2$ are always kept.
3) If $n > \mathcal{W}$, then the oldest keyframe ($\mathcal{I}_i$) that has the smallest angular distance to its nearest neighbor is removed in order to promote a more "spread-out" set of keyframes in terms of the camera view vector, i.e., the one that maximizes the following distance score:
$$d(\boldsymbol{\mathcal{I}}_i) = \max_{j \in [3,n] \setminus \{i\}} \hat{\boldsymbol{\omega}}_i^{\text{C}} \cdot \hat{\boldsymbol{\omega}}_j^{\text{C}} \tag{11}$$

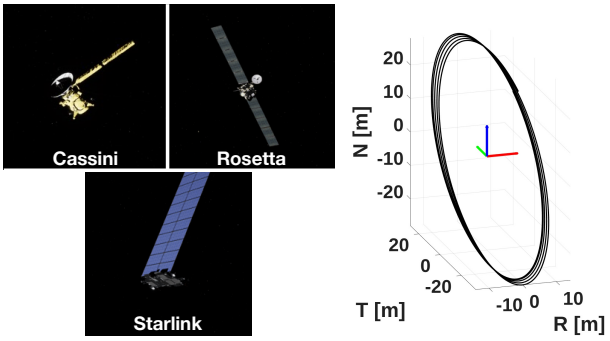

Fig. 3: (*Left*) Visualization of the adopted satellite models. (*Right*) The servicer's orbit relative to the target in the RTN frame.

IV. Experiments

Datasets. In order to train and validate the proposed pipeline, high-fidelity synthetic images are rendered using an Unreal Engine (UE)-based simulator [46] for three different satellite models with varying geometric features as shown in Fig. 3. Specifically, Cassini and Rosetta 3D models are obtained from the ESA Science Satellite Fleet [62], while the Starlink 3D model is CC-licensed and available online [63]. All images are rendered using the camera intrinsics adopted in the SPEED [15] dataset. Since SPEED camera intrinsics result in very high resolution images (1920 $\times$ 1200), all images are downsampled by the factor of 2 throughout the experiments.

For each satellite, a new dataset is created which comprises images captured during a simulated RPO trajectory in Low Earth Orbit (LEO) in which the servicer is in a passively safe e/i-vector-separated relative orbit [64] about the target RSO. The servicer's camera is always pointed at the target, and the images are captured every 5 seconds for 5 orbits, which amounts to 5,764 images per simulation. In order to obtain diverse views of the target more frequently, the target is simulated to tumble about its $\hat{x}$-axis at 2°/sec unless noted otherwise. The servicer's relative orbit is visualized in Fig. 3.

Finally, in order to evaluate the generalizability of the trained 3DGS models, they are evaluated on 100 images of

TABLE I: Quantitative performances of different configurations trained on sequential datasets. Arrows indicate the direction towards better performance. Bold faces denote the best performances. $|\mathcal{G}|$ denotes the number of Gaussians.

Config.	Cassini				Rosetta				Starlink			
	SSIM↑	PSNR↑	LPIPS↓	$\lvert\mathcal{G}\rvert$	SSIM↑	PSNR↑	LPIPS↓	$\lvert\mathcal{G}\rvert$	SSIM↑	PSNR↑	LPIPS↓	$\lvert\mathcal{G}\rvert$
(a) 3DGS [32]	0.9871	26.51	0.0171	42.7K	0.9844	29.61	0.0187	48.7K	0.9539	25.05	0.0440	74.4K
(b) + Illum.	0.9894	30.63	0.0152	51.6K	0.9889	33.40	0.0131	53.9K	0.9657	27.58	0.0299	88.5K
(c) + Shadow Splatting	0.9914	32.86	**0.0131**	45.9K	0.9897	34.53	0.0123	54.4K	**0.9705**	29.06	**0.0251**	89.7K
(d) + Isotropic Loss	**0.9916**	**33.40**	0.0136	49.7K	**0.9905**	**35.27**	**0.0109**	61.8K	0.9683	**29.89**	0.0289	99.6K

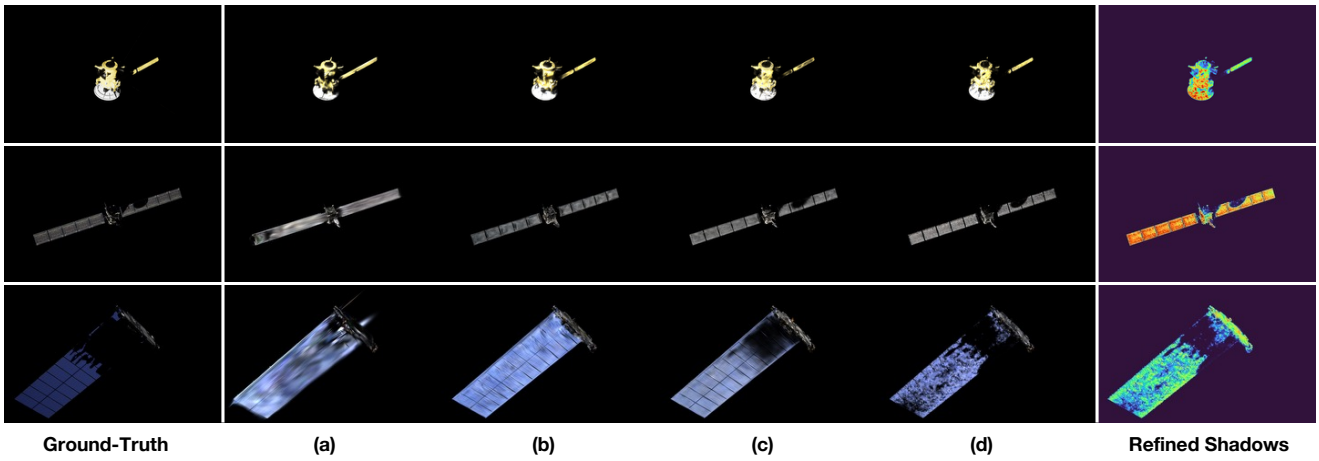

Fig. 4: Qualitative results of different configurations trained on sequential datasets.

each satellite rendered with random illumination conditions and camera and target poses.

Implementation Details. The training begins once the keyframe window is full. Specifically, a round of optimization is performed on a maximum $\mathcal{W} = 10$ keyframes (i.e., one round is equivalent to 10 global iterations) unless noted otherwise. The optimization is triggered only when the composition of the keyframe window changes. The Gaussians are added and pruned according to the original 3DGS implementation once every 10 rounds. Since the number of training rounds could vary in RPO settings, the learning rates are repeatedly decayed within each training round, resulting in a repeated sawtooth-like pattern throughout the training. The underlying implementation and all other training parameters follow the original 3DGS repository[1]. Recall that the Gaussian positions are initialized by uniformly sampling 10,000 points from the 3D model surfaces.

Four different architectural configurations are compared during the experiments. (a) The original 3DGS [32] learns per-Gaussian color directly from the differentiable rasterization and serves as the baseline. (b) Per-Gaussian appearance features ($\boldsymbol{f}$) and the known Sun and camera view vectors ($\hat{\boldsymbol{\omega}}^{\mathrm{S}}$, $\hat{\boldsymbol{\omega}}^{\mathrm{C}}$) are added and optimized via an appearance MLP (Φ). (c) The shadow splatting pipeline is added, and the resulting refined shadow image ($\boldsymbol{V}'$) is multiplied to the final rendered image from the main Gaussian splatting pipeline. (d) Finally, the isotropic scale regularization term (Eq. 7) is added to the final loss function.

[1] https://github.com/graphdeco-inria/gaussian-splatting

Metrics. This paper adopts three standard quantitative metrics commonly used in radiance fields literature: SSIM [60], Peak Signal-to-Noise Ratio (PSNR), and Learned Perceptual Image Patch Similarity (LPIPS) [65]. Since the initial Gaussians are sampled from the ground-truth 3D model, the geometric accuracy of reconstructed 3DGS models is not compared with the ground-truth models using 3D metrics such as Chamfer distance [46].

V. Results

The qualitative and quantitative performances of four different model configurations introduced in Sec. IV are respectively shown in Fig. 4 and Table I. It is immediately clear that the original 3DGS struggles with learning from dynamic illumination conditions in the synthetic images. Adding an appearance MLP and providing the Sun vector significantly improves performance across all metrics. However, Figure 4 shows that the reconstructed model fails to capture the global illumination and self-occlusion even with perfect knowledge of the Sun's position. Once the shadow splatting is incorporated into the pipeline, the performance once again visibly improves across all metrics, and the reconstructed model is capable of capturing the shadows caused by self-occlusion, albeit blurry and approximate. Finally, Figure 4 demonstrates that including the isotropic scale regularization ($\mathcal{L}_{\mathrm{iso}}$) term into the final loss function enables rendering of sharp shadows, but the overall reconstruction quality, especially that of the Starlink satellite with a massive solar panel, degrades a little as evidenced by drops in SSIM and LPIPS metrics as reported in Table I.

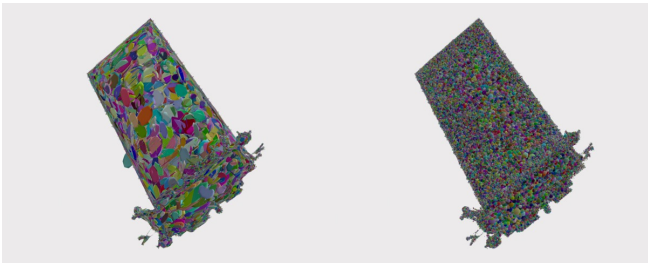

Fig. 5: Visualization of raw Gaussians after training without (*left*) and with (*right*) the $\mathcal{L}_{\text{iso}}$ term.

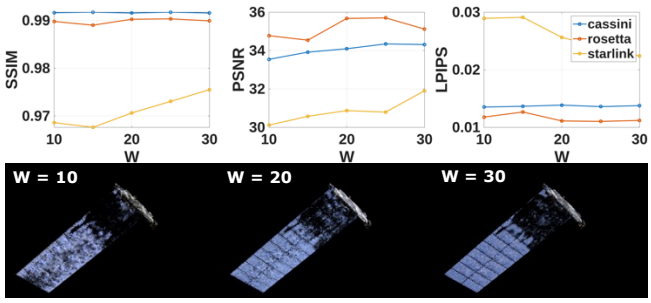

Fig. 6: Performance as the keyframe window size ($\mathcal{W}$) varies

The effect of $\mathcal{L}_{\text{iso}}$ is illustrated in Fig. 5 where the 3D Gaussians are visualized without 2D splatting but instead with random colors. It can be seen that training with the $\mathcal{L}_{\text{iso}}$ term causes the learned Gaussians to become much more spherical and granular, discouraging large and elongated shapes often encountered in the model trained without the regularization. Smaller Gaussians enable capturing sharper shadow contours since the effect of direct sunlight becomes more localized. However, it also has the side-effect of an increased number of Gaussians as shown in Table I.

Finally, a small ablation study is conducted by varying the length of the keyframe window ($\mathcal{W}$) using the full model configuration. Figure 6 reports that increasing $\mathcal{W}$ improves performance across all metrics, and the improvement is most distinct for the Starlink satellite. For instance, when the window length is as small as $\mathcal{W} = 10$, the reconstructed quality of the Starlink's solar panel is poor with many Gaussians unable to reconstruct proper colors. This is an expected behavior since raising $\mathcal{W}$ not only increases the overall number of training steps but also improves the diversity of samples that are observed at each training round. However, there is a trade-off in varying the window size, i.e., the number of images and associated metadata to maintain, and the on-board computational and memory capacity of satellites.

In fact, Fig. 7 presents a preliminary analysis of the training speed and GPU memory usage during a single training step. Each training step comprises both forward and backward propagation through the full rasterization pipeline shown in Fig. 2 including shadow splatting for a single 960 $\times$ 600 image input. The reported GPU memory denotes the

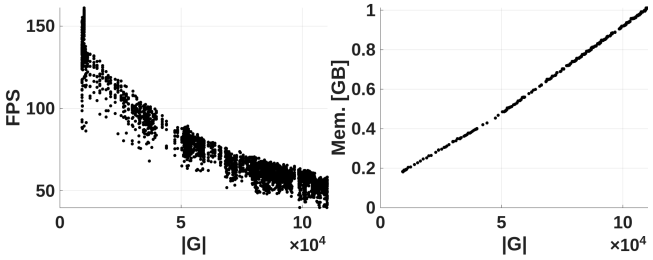

Fig. 7: Training speed and incurred memory vs. number of Gaussians ($|\mathcal{G}|$) measured on an NVIDIA GeForce RTX 4090 GPU.

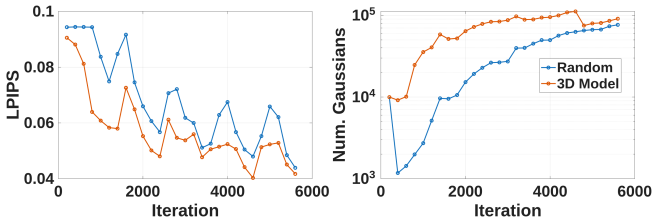

Fig. 8: Comparison of the LPIPS metric and the number of Gaussians when the model is initialized as 10,000 points sampled randomly and from the ground-truth 3D model.

amount allocated by the aforementioned training step[2] which excludes the memory reserved for the training data.

At nearly 100,000 Gaussians, the proposed pipeline still trains at about 50 frames per second on a desktop GPU while incurring an additional 1 GB of GPU memory. When more images are kept in the keyframe window, the additional memory overhead will inevitably increase which requires careful analysis. Meanwhile, the training speed can be improved by maintaining an optimal number of Gaussians throughout training, which in turn demands a more robust densification strategy.

VI. Discussions

As noted at the beginning of Sec. III, this work has made a number of ideal assumptions in order to focus on the photometric accuracy of the trained 3DGS model under dynamic illumination conditions in space. This section discusses how these assumptions could be relaxed moving forward.

Initialization of Gaussians. The conventional initialization method via SfM is not compatible with RPO missions, as SfM algorithms struggle under changing illumination conditions and require a large batch of images. However, as portrayed in Fig. 8, proper initialization of the Gaussians can significantly accelerate the model convergence. While the model can still be trained from a completely random set of points, it initially suffers from a huge loss of Gaussians as the densification strategy kicks in, further slowing down the training.

Several remedies are being investigated and adopted in the literature. One is to leverage zero-shot shape abstraction algorithms, such as that of Park & D'Amico [46], which

[2]The memory is computed as the difference of the PyTorch's `torch.cuda.max_memory_allocated()` calls before and after the training step.

predict coarse, normalized 3D structures of the target and the camera poses simultaneously from single images. For instance, the concurrent work by Francesch Huc et al. [66] successfully initializes 3DGS models using point clouds sampled from the surfaces of reconstructed superquadric assemblies. Another potential solution is to employ additional sensor modalities such as depth cameras that can be used to initialize points along the visible surfaces [36].

Availability of Foreground Masks. Foreground masks allow background removal which helps concentrate Gaussians to the target RSO in question. Depth sensors can be used in this case as well to swiftly isolate the nearby objects from far-away backgrounds (e.g., Earth). One can also utilize general-purpose segmentation models (e.g., Segment Anything [67]) fine-tuned specifically for spaceborne missions.

Knowledge of Pose Labels. In real missions, pose labels are not available and must be jointly estimated as the 3DGS model is learned. This can be achieved through SLAM-like procedures [30], [36], [38], in which case the keyframe management system must rely on the covisibility of Gaussians rather than noisy pose estimates. Another possibility is to directly integrate the pose predictions from the aforementioned zero-shot method [46], [49] into the SLAM pipeline.

VII. Conclusions

This paper presented a novel pipeline to train a 3D Gaussian Splatting (3DGS) model using sequences of target Resident Space Object (RSO) images captured during Rendezvous and Proximity Operations (RPO) in space. Specifically, considering extreme and dynamic illumination conditions in spaceborne imagery and the fact that 3DGS assumes static scenes, the proposed pipeline incorporates (1) the Sun vector tracked by the servicer's onboard sensors and (2) shadow splatting, which leverages an existing splatting algorithm, to evaluate the visibility of each Gaussian given the known lighting direction at each image. The experimental results on high-fidelity synthetic images captured during the simulations of representative RPO scenarios indicate that the proposed method is capable of learning a 3DGS model of the target RSO from an image sequence, able to capture global effects of changing Sun directions and self-occlusion.

The proposed pipeline currently hinges on a number of ideal assumptions, such as known pose labels and initialization of Gaussians from a ground-truth 3D mesh model. Future plans to relax these assumptions are discussed, including the adoption of additional sensor modalities and SLAM-like mapping and tracking procedures.

Acknowledgment

The authors would like to thank Dr. Kuldeep Barad at Redwire Space Europe and Emily Bates and Pol Francesch Huc at SLAB for helpful discussions and constructive feedback.

References

[1] B. B. Reed, R. C. Smith, B. J. Naasz *et al.*, "The Restore-L Servicing Mission," *AIAA Space 2016*, 2016.

[2] M. Pyrak and J. Anderson, "Performance of Northrop Grumman's Mission Extension Vehicle (MEV) RPO imagers at GEO," in *Autonomous Systems: Sensors, Processing and Security for Ground, Air, Sea and Space Vehicles and Infrastructure 2022*, 2022, p. 121150A.

[3] G. S. Aglietti, B. Taylor, S. Fellowes *et al.*, "RemoveDEBRIS: An in-orbit demonstration of technologies for the removal of space debris," *The Aeronautical Journal*, vol. 124, no. 1271, p. 1–23, 2020.

[4] 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*, vol. 2, no. 2, p. 171, 2014.

[5] S. Sharma, J. Ventura, and S. D'Amico, "Robust model-based monocular pose initialization for noncooperative spacecraft rendezvous," *Journal of Spacecraft and Rockets*, vol. 55, no. 6, pp. 1414–1429, 2018.

[6] V. Capuano, S. R. Alimo, A. Q. Ho *et al.*, "Robust features extraction for on-board monocular-based spacecraft pose acquisition," in *AIAA Scitech 2019 Forum*, 2019.

[7] A. A. Grompone, "Vision-based 3D motion estimation for on-orbit proximity satellite tracking and navigation," 2015. [Online]. Available: https://calhoun.nps.edu/handle/10945/45863

[8] A. Petit, E. Marchand, and K. Kanani, "Vision-based space autonomous rendezvous: A case study," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 619–624.

[9] M. Kisantal, S. Sharma, T. H. Park *et al.*, "Satellite pose estimation challenge: Dataset, competition design and results," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 5, pp. 4083–4098, 2020.

[10] T. H. Park, M. Märtens, M. Jawaid *et al.*, "Satellite pose estimation competition 2021: Results and analyses," *Acta Astronautica*, vol. 204, pp. 640–665, 2023.

[11] T. H. Park, S. Sharma, and S. D'Amico, "Towards robust learning-based pose estimation of noncooperative spacecraft," in *2019 AAS/AIAA Astrodynamics Specialist Conference, Portland, Maine*, August 11-15 2019.

[12] T. H. Park and S. D'Amico, "Robust multi-task learning and online refinement for spacecraft pose estimation across domain gap," *Advances in Space Research*, vol. 73, no. 11, pp. 5726–5740, 2024, recent Advances in Satellite Constellations and Formation Flying.

[13] T. H. Park and S. D'Amico, "Bridging the domain gap for flight-ready spaceborne vision," 2024. [Online]. Available: https://arxiv.org/abs/2409.11661

[14] S. Sharma, C. Beierle, and S. D'Amico, "Pose estimation for non-cooperative spacecraft rendezvous using convolutional neural networks," in *2018 IEEE Aerospace Conference*, 2018, pp. 1–12.

[15] S. Sharma and S. D'Amico, "Neural network-based pose estimation for noncooperative spacecraft rendezvous," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 6, pp. 4638–4658, 2020.

[16] B. Chen, J. Cao, A. Parra *et al.*, "Satellite pose estimation with deep landmark regression and nonlinear pose refinement," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 2816–2824.

[17] P. F. Proença and Y. Gao, "Deep learning for spacecraft pose estimation from photorealistic rendering," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 6007–6013.

[18] L. Pasqualetto Cassinis, R. Fonod, E. Gill *et al.*, "Evaluation of tightly- and loosely-coupled approaches in CNN-based pose estimation systems for uncooperative spacecraft," *Acta Astronautica*, vol. 182, pp. 189–202, 2021.

[19] Y. Hu, S. Speierer, W. Jakob *et al.*, "Wide-depth-range 6D object pose estimation in space," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 15 865–15 874.

[20] A. Garcia, M. A. Musallam, V. Gaudilliere *et al.*, "LSPnet: A 2d localization-oriented spacecraft pose estimation neural network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2021, pp. 2048–2056.

[21] T. H. Park and S. D'Amico, "Adaptive neural-network-based unscented kalman filter for robust pose tracking of noncooperative spacecraft," *Journal of Guidance, Control, and Dynamics*, vol. 46, no. 9, pp. 1671–1688, 2023.

[22] T. H. Park and S. D'Amico, "Online supervised training of spaceborne vision during proximity operations using adaptive kalman filtering," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 11 744–11 752.

[23] L. Pasqualetto Cassinis, T. H. Park, N. Stacey *et al.*, "Leveraging neural network uncertainty in adaptive unscented Kalman filter for spacecraft pose estimation," *Advances in Space Research*, vol. 71, no. 12, pp. 5061–5082, 2023.

[24] S. Kaki, J. Deutsch, K. Black *et al.*, "Real-time image-based relative pose estimation and filtering for spacecraft applications," *Journal of Aerospace Information Systems*, vol. 20, no. 6, pp. 290–307, 2023.

[25] T. H. Park, M. Märtens, G. Lecuyer *et al.*, "SPEED+: Next-generation dataset for spacecraft pose estimation across domain gap," in *2022 IEEE Aerospace Conference (AERO)*, 2022, pp. 1–15.

[26] M. A. Musallam, V. Gaudilliere, E. Ghorbel *et al.*, "Spacecraft recognition leveraging knowledge of space environment: Simulator, dataset, competition design and analysis," in *2021 IEEE International Conference on Image Processing Challenges (ICIPC)*, 2021, pp. 11–15.

[27] T. H. Park, J. Bosse, and S. D'Amico, "Robotic testbed for rendezvous and optical navigation: Multi-source calibration and machine learning use cases," in *2021 AAS/AIAA Astrodynamics Specialist Conference, Big Sky, Vitrual*, August 9-11 2021.

[28] L. Pasqualetto Cassinis, A. Menicucci, E. Gill *et al.*, "On-ground validation of a cnn-based monocular pose estimation system for uncooperative spacecraft: Bridging domain shift in rendezvous scenarios," *Acta Astronautica*, vol. 196, pp. 123–138, 2022.

[29] M. Olivares-Mendez, M. R. Makhdoomi, B. C. Yalçın *et al.*, "Zero-G Lab: A multi-purpose facility for emulating space operations," *Journal of Space Safety Engineering*, vol. 10, no. 4, pp. 509–521, 2023.

[30] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[31] B. Mildenhall, P. P. Srinivasan, M. Tancik *et al.*, "NeRF: Representing scenes as neural radiance fields for view synthesis," in *Computer Vision – ECCV 2020*, 2020, pp. 405–421.

[32] B. Kerbl, G. Kopanas, T. Leimkuehler *et al.*, "3D gaussian splatting for real-time radiance field rendering," *ACM Trans. Graph.*, vol. 42, no. 4, Jul. 2023.

[33] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4104–4113.

[34] J. L. Schönberger, E. Zheng, J.-M. Frahm *et al.*, "Pixelwise view selection for unstructured multi-view stereo," in *Computer Vision – ECCV 2016*, 2016, pp. 501–518.

[35] A. Rosinol, J. J. Leonard, and L. Carlone, "NeRF-SLAM: Real-time dense monocular SLAM with neural radiance fields," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 3437–3444.

[36] K. R. Barad, A. Richard, J. Dentler *et al.*, "Object-centric reconstruction and tracking of dynamic unknown objects using 3d gaussian splatting," in *2024 International Conference on Space Robotics (iSpaRo)*, 2024, pp. 202–209.

[37] Y. Fu, X. Wang, S. Liu *et al.*, "COLMAP-free 3D gaussian splatting," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 20 796–20 805.

[38] H. Matsuki, R. Murai, P. H. J. Kelly *et al.*, "Gaussian splatting SLAM," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 18 039–18 048.

[39] R. Martin-Brualla, N. Radwan, M. S. M. Sajjadi *et al.*, "NeRF in the Wild: Neural radiance fields for unconstrained photo collections," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 7206–7215.

[40] J. Kulhanek, S. Peng, Z. Kukelova *et al.*, "WildGaussians: 3D gaussian splatting in the wild," in *Advances in Neural Information Processing Systems*, 2024, pp. 21 271–21 288.

[41] Z. Bi, Y. Zeng, C. Zeng *et al.*, "GS3: Efficient relighting with triple gaussian splatting," in *SIGGRAPH Asia 2024 Conference Papers*, 2024.

[42] B. E. Tweddle, "Computer vision-based localization and mapping of an unknown, uncooperative and spinning target for spacecraft proximity operations," Ph.D. dissertation, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, 2013.

[43] H. Bay, A. Ess, T. Tuytelaars *et al.*, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[44] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental smoothing and mapping," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1365–1378, 2008.

[45] M. Dor and P. Tsiotras, "ORB-SLAM applied to spacecraft non-cooperative rendezvous," in *2018 Space Flight Mechanics Meeting*, 2018.

[46] T. H. Park and S. D'Amico, "Rapid abstraction of spacecraft 3D structure from single 2D image," in *AIAA SCITECH 2024 Forum*, 2024.

[47] A. H. Barr, "Superquadrics and angle-preserving transformations," *IEEE Computer Graphics and Applications*, vol. 1, no. 1, pp. 11–23, 1981.

[48] T. H. Park, E. Bates, and S. D'Amico, "Improving zero-shot abstraction of unknown spacecraft 3d shape as primitive assembly," in *12th International Workshop on Satellite Constellation & Formation Flying, Kaohsiung, Taiwan*, 2024.

[49] E. Bates and S. D'Amico, "Removing ambiguities in concurrent monocular single-shot spacecraft shape and pose estimation using a deep neural network," in *48th Rocky Mountain AAS GN&C Conference*, 2025.

[50] B. Caruso, T. Mahendrakar, V. M. Nguyen *et al.*, "3D reconstruction of non-cooperative resident space objects using Instant NGP-accelerated NeRF and D-NeRF," in *33rd AAS/AIAA Space Flight Mechanics Meeting, Austin, Texas*, 2023.

[51] A. Mergy, G. Lecuyer, D. Derksen *et al.*, "Vision-based neural scene representations for spacecraft," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2021, pp. 2002–2011.

[52] T. Müller, A. Evans, C. Schied *et al.*, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Trans. Graph.*, vol. 41, 7 2022.

[53] A. Pumarola, E. Corona, G. Pons-Moll *et al.*, "D-nerf: Neural radiance fields for dynamic scenes," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 10 313–10 322.

[54] M. Tancik, V. Casser, X. Yan *et al.*, "Block-NeRF: Scalable large scene neural view synthesis," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 8238–8248.

[55] Y. Yang, S. Zhang, Z. Huang *et al.*, "Cross-ray neural radiance fields for novel-view synthesis from unconstrained image collections," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 15 855–15 865.

[56] X. Chen, Q. Zhang, X. Li *et al.*, "Hallucinated neural radiance fields in the wild," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 12 933–12 942.

[57] D. Zhang, C. Wang, W. Wang *et al.*, "Gaussian in the wild: 3d gaussian splatting for unconstrained image collections," in *Computer Vision – ECCV 2024*, 2025, pp. 341–359.

[58] H. Dahmani, M. Bennehar, N. Piasco *et al.*, "SWAG: Splatting in-the-wild images with appearance-conditioned gaussians," in *Computer Vision – ECCV 2024*, 2025, pp. 325–340.

[59] M. Zwicker, H. Pfister, J. van Baar *et al.*, "EWA volume splatting," in *Proceedings Visualization, 2001. VIS '01.*, 2001, pp. 29–538.

[60] Z. Wang, A. Bovik, H. Sheikh *et al.*, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[61] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 611–625, 2018.

[62] "ESA science satellite fleet," Available at https://scifleet.esa.int/, 2025, accessed 2025/08/01.

[63] "Starlink spacex satellite," Available at https://sketchfab.com/3d-models/starlink-spacex-satellite-0a60f6720c5141c9a1c6d71aac108b31, 2025, accessed 2025/08/01.

[64] O. Montenbruck, M. Kirschner, S. D'Amico *et al.*, "E/I-vector separation for safe switching of the GRACE formation," *Aerospace Science and Technology*, vol. 10, no. 7, pp. 628–635, 2006.

[65] R. Zhang, P. Isola, A. A. Efros *et al.*, "The unreasonable effectiveness of deep features as a perceptual metric," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 586–595.

[66] P. Francesch Huc, E. Bates, and S. D'Amico, "Fast learning of non-cooperative spacecraft 3d models through primitive initialization," in *2025 AAS/AIAA Astrodynamics Specialist Conference, Boston, Massachusetts*, August 10-14 2025.

[67] A. Kirillov, E. Mintun, N. Ravi *et al.*, "Segment anything," 2023. [Online]. Available: https://arxiv.org/abs/2304.02643