Mesh-as-a-Service: Automated 3D Modeling *Fast as L-AI-ghtning*

Mathijs Henquet, Thomas Bellucci, Chihab Amghane, Jasper Steringa
Royal NLR - Netherlands Aerospace Centre
Amsterdam, Netherlands
Mathijs.Henquet@nlr.nl, Thomas.Bellucci@nlr.nl
Chihab.Amghane@nlr.nl, Jasper.Steringa@nlr.nl

Lodewijck Foorthuis
Royal Netherlands Air Force
Breda, Netherlands
LR.Foorthuis@mindef.nl

ABSTRACT

The Royal Netherlands Air Force (RNLAF) is revolutionizing its tactical training capabilities by developing a state-of-the-art Multi-Ship-Multi-Type (MSMT) simulator center at Gilze-Rijen Air Force Base. This initiative necessitates the creation of high fidelity virtual models such as a virtual Auxiliary Power Unit (APU) for the CH-47F Chinook helicopter's Rear Crew Trainer (RCT) simulator. However, traditional methods of 3D modeling are resource-intensive and time-consuming, posing a challenge for high fidelity virtual model development. To address this challenge, this paper proposes using Artificial Intelligence (AI) techniques, including Surface-Aligned Gaussian Splatting (SuGaR) and object segmentation, to efficiently generate 3D models from pictures of real-world objects.

We outline criteria for meshes to be compatible with live simulators, focusing on polygon count and fidelity, and evaluate the capability of various experimental AI techniques in meeting these requirements. As the resulting objects have to be isolated from the environment, we examine AI segmentation techniques to automate this. We present our findings by developing 3D models from existing image datasets and a virtual APU, which will demonstrate that these methods offer a favorable trade-off between quality, labor, and cost. Using experimental AI is challenging due to the demand for specialized skills and the plethora of techniques involved. To increase accessibility, we build on our lessons learned and propose a Mesh-as-a-Service platform to facilitate the integration of real-world objects into simulated environments.

Our research demonstrates that AI-driven methods may one day transform the landscape of 3D mesh generation, presenting a scalable and efficient alternative to traditional approaches. The development of 'mesh-as-a-service' will make the creation of digital twins and the rapid construction of immersive virtual training scenarios more accessible. These innovations have the potential to set new standards in military simulation and training, with implications that extend beyond the RNLAF to the defense sector worldwide.

ABOUT THE AUTHORS

Mathijs Henquet has studied Mathematics and Computer Science at Utrecht University. At the Royal NLR he focuses on AI/ML innovations and Spatial Computing based methods.

Thomas Bellucci is an R&D engineer at the Royal NLR at the department of Training & Simulation and works on research projects involving Artificial Intelligence and machine learning. He studied Artificial Intelligence (AI) at the University of Amsterdam (UvA) and Vrije Universiteit (VU), completing his MSc program summa cum laude.

Chihab Amghane has studied Artificial Intelligence at Radboud University and is currently pursuing a PhD at Tilburg University while working as an R&D Engineer at the Royal NLR.

Jasper Steringa, studied Astronomy at the University of Groningen (RUG). His current R&D work at the Royal NLR focuses on applying data analytics and AI/ML techniques to improve aircraft fleet sustainment.

Lodewijck Foorthuis earned his Master's degree in Aerospace Engineering from Delft University of Technology. He has managed numerous simulation studies in the military domain and is project manager for the Royal Netherlands Air Force, focusing on tactical helicopter simulation projects.

Mesh-as-a-Service: Automated 3D Modeling *Fast as L-AI-ghtning*

Mathijs Henquet, Thomas Bellucci, Chihab Amghane, Jasper Steringa
Royal NLR - Netherlands Aerospace Centre
Amsterdam, Netherlands
Mathijs.Henquet@nlr.nl, Thomas.Bellucci@nlr.nl
Chihab.Amghane@nlr.nl, Jasper.Steringa@nlr.nl

Lodewijck Foorthuis
Royal Netherlands Air Force
Breda, Netherlands
LR.Foorthuis@mindef.nl

INTRODUCTION

The Royal Netherlands Air Force (RNLAF) is currently facing a significant challenge in training tactical helicopter operations in complex scenarios, particularly for Chinook loadmasters. The existing simulators for the CH-47F Chinook are limited in fidelity, and live-flight training is not only expensive and unsustainable but also restricted by available airspace and training opportunities. This limitation hinders the ability of the RNLAF to train its loadmasters in realistic and dynamic scenarios, which is essential for maintaining operational readiness and effectiveness.

To address this gap, the RNLAF is the world's first military that is developing an innovative, high-fidelity Multi-Ship-Multi-Type (MSMT) simulation center at Gilze-Rijen Air Force Base, which includes a CH-47F Chinook Rear Crew Trainer (RCT). The RCT is a critical component of the MSMT simulation center, as it enables loadmasters to train in a realistic and immersive environment, focusing on their specific tasks and responsibilities. The RCT will allow loadmasters to train in a variety of scenarios, including cargo loading and unloading, aerial delivery, and helicopter insertion and extraction operations.

The development of the RCT is of considerable importance, as it fills a critical gap in the training methodology of the RNLAF. Loadmasters require specific training, which cannot be adequately addressed in a live environment due to safety and logistical constraints. The RCT will enable loadmasters to train in a safe and effective manner, focusing on their specific skills and tasks. For the development of the RCT, both real and virtual components are used. The development of these virtual components currently uses traditional 3D modeling techniques. Unfortunately, traditional 3D modeling techniques used to create virtual models for the RCT simulator are often time-consuming, labor-intensive, and require significant expertise. The resulting models may not accurately capture the complexity and detail of real-world objects, leading to a lack of realism and immersion in the simulation environment.

To address these shortcomings, this paper explores the potential of Artificial Intelligence (AI) techniques, specifically 3D Gaussian Splatting (3DGS), to revolutionize the 3D mesh generation process. 3DGS offers a promising solution by enabling the rapid creation of accurate and detailed virtual models from collections of 2D images. The required manual modeling efforts would be reduced to only modeling the interactive elements. The combined approach of AI to automate tasks which typically require manual labour has the potential to significantly reduce the time and cost associated with creating high-fidelity virtual models, while also improving the realism and effectiveness of the RCT simulator. In this work we set out to research whether the current AI methods are sufficiently mature to replace part of the manual 3D modeling labor required for creating non-interactive 3D reconstructions. Our approach builds on top of state of the art AI methods and leverages their combined strengths to form a pipeline enabling the generation of 3D meshes from images, we call this pipeline *Mesh-as-a-Service*.

BACKGROUND: AUTOMATED 3D MODELING

In this section we will detail fundamental research on which the Mesh-as-a-Service pipeline is built. We start with an overview of the field of 3D reconstruction. This is followed by a concise explanation of the 3DGS technique and SuGaR – an refinement to improve extracted meshes from 3DGS. We then briefly discuss recent advancements in the field of image segmentation both in 2D and 3D. The background section is concluded with a brief description of surface reconstruction techniques from the volumetric representations.

3D Reconstruction

The 3D reconstruction of polygonal meshes is a critical process in computer graphics and computer vision, aiming to create a 3D surface representation from data such as images or videos. Traditional 3D reconstruction approaches can be divided into two phases: pose detection, where the position and orientation of each camera view and objects are determined in the 3D space, and dense reconstruction, where this information is used to create a detailed 3D model in the form of a 3D mesh. Both of these steps can be executed on a variety of inputs, ranging from plain photogrammetry to methods augmented with depth information such as structured light or laser scanning techniques. The extra depth information provided by these methods support the reconstruction steps, but require additional specialized hardware.

Photogrammetry reconstructs 3D models by processing multiple photographs taken from different angles around the object of interest. For pose detection, an algorithm called Structure from Motion (SfM) is often used. This technique involves several stages: (1) feature detection, wherein high contrast key points, also called landmarks, are identified in the images; (2) feature matching, where key points across images are matched; and (3) triangulation, where the 3D coordinates of each key point and camera poses are calculated iterative using alternating steps of triangulation and bundle adjustment. This yields camera poses and a sparse point cloud of features. This sparse point cloud is consequently densified from which a mesh can be extracted using a variety of methods. Photogrammetry is particularly effective for its ability to work with conventional cameras. However, it can struggle with textureless or reflective surfaces and typically requires significant computational resources for large datasets.

Structured light scanning involves projecting artificial patterns onto the object's surface and analyzing the deformation of these patterns captured by cameras. Conversely, laser scanning or *LiDAR* utilizes lasers to measure distances to the surface, generating a dense point cloud representation. Structured light scanning and laser scanning allow the capture of a reasonably good level of detail depending on the resolution of the hardware. Structured light scanning is relatively cost-effective, while laser scanning offers exceptional depth accuracy. However, both methods face challenges as they require dedicated capturing hardware or can be sensitive to ambient lighting conditions.

Recent advancements in 3D reconstruction have introduced novel techniques such as Neural Radiance Fields (NeRFs) (Mildenhall et al., 2021). Diverging from traditional methods, NeRF represent scenes as continuous volumetric representations, also called density fields, parameterized by a deep neural network, enabling high-fidelity reconstructions from sparse and unstructured input data like images. Later, 3DGS (Kerbl, Kopanas, Leimkühler, & Drettakis, 2023) replaces the deep neural network with a continuous collections of 3D Gaussian. As these can be rendered with traditional rasterization methods this yields a significant performance gain over NeRF, which are raytraced. Moreover, the more explicit 3D information captured by 3DGS lends itself to mesh extraction as done in (Guédon & Lepetit, 2024b).

3D Gaussian Splatting (3DGS)

In 3D Gaussian Splatting (Kerbl et al., 2023), a scene is represented using a set $\mathcal{G}$ of 3D Gaussians:

$$\mathcal{G} = \{ (\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, \alpha_i, \mathbf{SH}_i) \mid i = 1, \ldots, N \} \qquad \text{with } N \text{ approx. millions} \tag{1}$$

Each Guassian is determined by $\boldsymbol{\mu}_i \in \mathbb{R}^3$ the 3D position of the Gaussian, $\boldsymbol{\Sigma}_i \in \mathbb{R}^{3\times3}$ is the covariance matrix defining its shape and orientation, $\alpha_i \in [0,1]$ is the opacity, and $\mathbf{SH}_i$ are the spherical harmonic coefficients for view-dependent appearance. The spherical harmonics are typically represented up to a certain degree, e.g. 4, each degree d having $2d+1$ coefficients per color channel. The SH degrees can be interpreted as follows:

- Degree 0 (1 coefficient per color): The average color of the Gaussian; known as the albedo shading.
- Degree 1 (3 coefficients per color): Linear directional lighting information; diffuse or Lambertian shading.
- Degrees 2+: Higher-frequency lighting effects; complex view-dependent appearance eg specular highlights.

This formulation captures geometry and appearance in a unified framework. Its ability to capture rich, view-dependent appearance allows for the faithful reconstruction of objects with inconsistent lighting and glossy surfaces.

To render an image, these 3D Gaussians are projected onto the image plane and composited in a back-to-front order similar to triangle rasterization. The color of a pixel is determined by the accumulated contributions of all Gaussians that project onto it, modulated by their opacity and view-dependent appearance as encoded in the spherical harmonics. The scene is represented by smooth Gaussians, allowing the render pipeline to remain completely differentiable, enabling the use of conventional Gradient Descent-based optimization to optimize the scene. Together with an ad-hoc method of pruning and cloning Gaussians this yields a conceptually simple algorithm for 3D volumetric reconstruction.

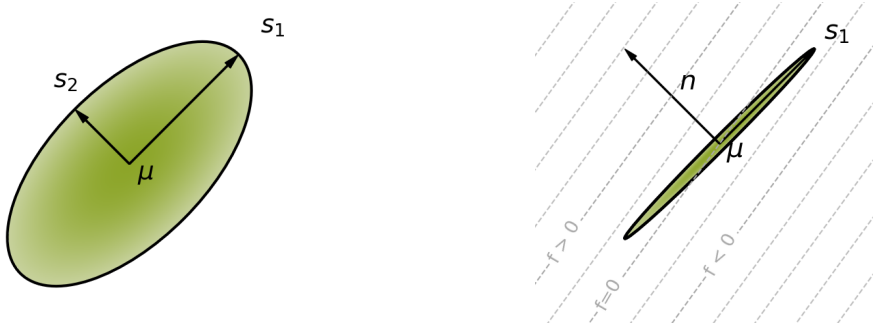
Surface Aligned Gaussians

As observed in the Surface-Aligned Gaussian Splatting (SuGaR) paper (Guédon & Lepetit, 2024b), the Gaussians produced by vanilla 3DGS are often not suitable for 3D reconstruction. This is because Gaussians tend to not align well with the surfaces of objects. Thus in SuGaR a regularization term is introduced that encourages the Gaussians to be well distributed, aligned with the surface (e.g. flat) and have limited overlap with the neighbouring Gaussians.

To grasp the approach used by SuGaR, consider how the covariance matrix of a Gaussian $\mathbf{\Sigma} \in \mathbb{R}^{3\times3}$ can be reparameterized with a rotation matrix R and an ordered scale vector $\mathbf{s} \in \mathbb{R}^3$. This is the diagonalization of $\mathbf{\Sigma}$:

$$\mathbf{\Sigma} = R \begin{bmatrix} s_1 & 0 & 0 \\ 0 & s_2 & 0 \\ 0 & 0 & s_3 \end{bmatrix} R^T, \qquad R \in \mathrm{SO}(3) \text{ a rotation matrix; and } s_1 \geq s_2 \geq s_3 \in \mathbb{R} \tag{2}$$

In Figure 1a an example of a 2D Gaussian is visualized, there are now 2 scale parameters $s_1 \geq s_2$ corresponding to the major and minor axes of the ellipse approximated by the Gaussian. We say that a Gaussian is flat if its last and thus smallest scale parameter is much smaller then the others. In this case, see Figure 1b, the axis corresponding to the smallest scale parameter is the normal $\mathbf{n}$ of the Gaussian.



(a) Regular 2D Gaussian (3DGS) **(b) Surface Aligned 2D Gaussian (SuGaR)**

Figure 1. Abstract 2D illustration of vanilla Gaussians (3DGS) and surface aligned Gaussian (SuGaR)

To do surface alignment, an approximation of a Signed Distance Field (SDF) f of the scene is maintained. This is a function so that the contour line $f = 0$ tracks the surface of the objects in the scene. A schematic 2D situation is displayed in Figure 1b with contour lines of f. A flat Gaussian is aligned with the surface if it lies along the contour lines, or equivalently that the normal $\mathbf{n}$ aligns with the gradient ∇f. SuGaR then introduces a loss which incentives Gaussians to become flat and surface aligned, we refer the reader to (Guédon & Lepetit, 2024b) for details.

Segmentation

Image segmentation is the process of partitioning an image into its constituent regions or objects, is a fundamental task in computer vision and image processing with diverse applications. With recent advancements in 3D reconstruction techniques from 2D images (see previous section) the idea to perform segmentation in 3D has become increasingly desirable. A common direction in this domain is to leverage 2D segmentation models for 3D segmentation.

Recent advancements have focused on making segmentation models more generalizable and efficient. Meta's Segment Anything Model (SAM) (Kirillov et al., 2023) represents a significant leap in this direction. SAM leverages a transformer-based architecture, which has shown remarkable success in various vision tasks due to its ability to model long-range dependencies and contextual information. SAM is designed to be a universal segmentation model, capable of segmenting any object in an image without task-specific training. This is achieved through a combination of large-scale pre-training on diverse datasets and a point prompt-based interface that allows the model to adapt to different segmentation tasks dynamically (Kirillov et al., 2023).

One innovative method for achieving 3D segmentation from a powerful 2D segmentation model such as SAM is the GARField technique (Kim et al., 2024). This approach involves training a scale-dependent affinity field on top of a 3D reconstruction model. The affinity field is designed such that 3D points exhibit similar affinities at a given scale if they frequently appear together in segments of the same real-world scale. To generate these segments the SAM model is used to sample a large set of segmentations on the training images of the 3D reconstruction. An extra loss ensures that the affinity field reflects the hierarchical structure of 3D scenes. This enables a flexible and detailed decomposition of the 3D model, with powerful potential applications for asset extraction.

Surface Reconstruction

Traditional volumetric surface reconstruction methods offer a compelling approach to high-fidelity mesh reconstruction of 3D objects represented as continuous density fields. One prominent approach is the Marching Cubes algorithm, pioneered by (Lorensen & Cline, 1998), which efficiently extracts isosurfaces from volumetric data by iteratively shrinking a discrete 3D voxel grid. While effective in generating high resolution polygon meshes, for this method to yield satisfactory results, excessively high resolution voxel grids must be used, resulting in high vertex counts.

To remedy the limitations of marching cubes, Delaunay triangulation was introduced (Drysdale, McElfresh, & Snoeyink, 2001) which generates a mesh from a sparse point cloud by connecting neighboring points using triangulation. A similar algorithm, known as Ball pivoting (Bernardini, Mittleman, Rushmeier, Silva, & Taubin, 1999) constructs meshes by iteratively growing spheres around surface points, connecting them to form triangles. More recently, Poisson surface reconstruction was proposed (Kazhdan, Bolitho, & Hoppe, 2006) and used by SuGaR (Guédon & Lepetit, 2024b) to reconstruct surfaces from a 3D Gaussian Splatting model. Poisson reconstruction leverages oriented points on a level set of a volumetric representation to solve for a smooth surface. 3D meshes reconstructed using the Poisson method typically preserve fine surface details present in the input point cloud while still producing a smooth mesh.

SYSTEM DESIGN

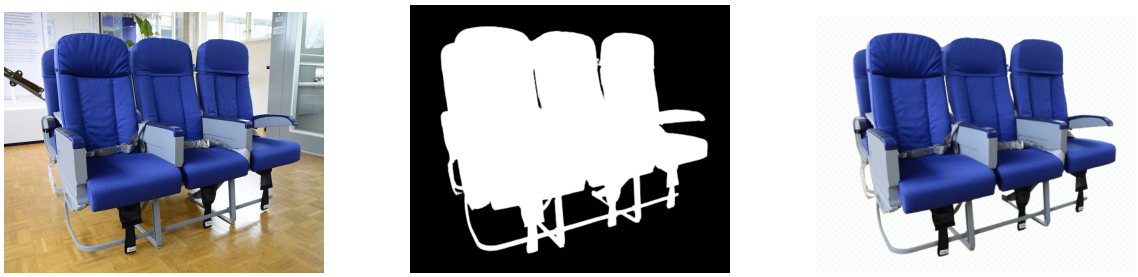
Our method synthesizes various state-of-the-art approaches detailed above, for the RCT APU use case. We first provide a brief overview of the main steps, focusing on user interaction with the system, and then delve into the technical details in the next subsections.

1. **Data Collection:** The user captures an object of interest as a series of photographs using an ordinary, consumer-grade camera (typically 50 to 300 images).
2. **Preparation:** Using a browser-based interface, we generate, verify and refine automatically generated 2D masks that isolate the object from its background.
3. **Processing:** A 3DGS model is trained using the captured images of the object of interest and masks, This processing step typically takes $\sim$2 hours on a modern GPU, which can be executed without further intervention.
4. **Inspection:** Upon completion of the 3D reconstruction, the result can be visually inspected.
5. **Mesh Extraction and Texturing:** After selecting an object and polygon budget a textured mesh is extracted.

The final output is a high-quality, textured mesh representing the object of interest with user-defined levels of detail.

Preparation

The preparation phase begins with the presentation of a scene through a series of photographs captured using an ordinary camera without prior calibration. To enhance the robustness of dense pose estimation against changing backgrounds, we isolate the object of interest by creating a set of 2D masks. This isolation is achieved by employing Meta's Segment Anything Model (SAM) (Kirillov et al., 2023) on a set of point prompts generated for each image. These point prompts are automatically generated from a textual prompt (e.g., "aeroplane") using the CLIP object detection model (Radford et al., 2021), applied as a sliding window over square patches of each image. The resulting masks can be visually inspected and corrected by the user through a browser-based interface, ensuring correct isolation of the object of interest.



(a) Training image of seats **(b) Generated 2D background mask** **(c) Seats with background masked**

Figure 2. Example of automatic masking using SAM and CLIP with "airplane cabin seats" as the textual prompt.

Processing

The images are first processed using the Structure from Motion (SfM) algorithm from COLMAP (Schonberger & Frahm, 2016) to obtain camera poses. By incorporating the generated masks into SfM, we ensure that pose detection features are generated only for the object of interest, making the pipeline robust to changes in the background.

Using the pose-detected images, we create a dense 3D Gaussian Splatting reconstruction with several additional regularizers. We employ the adaptive control loop tweak introduced by AbsGS (Ye, Li, Liu, Qiao, & Dou, 2024) to aid in the reconstruction of fine details. Furthermore, following the approach of SuGaR (Guédon & Lepetit, 2024b), we incorporate a regularization term that encourages Gaussians to be flat and aligned to the surface, making the reconstruction more suitable for mesh extraction later. The resulting 3D reconstruction is further augmented with segmentation information using the GARField method (Kim et al., 2024). This process determines a scale-dependent affinity field, indicating which parts of the scene belong together at a certain scale.

The final result is a surface-aligned 3D Gaussian Splatting model with hierarchical segmentation information. The resulting scene can be visualized for the user in our browser-based tool. From this interface, the user can select desired objects and assign a specific polygon budget. Using this information, we create meshes with the desired polygon budget, allowing for efficient rendering and manipulation of the reconstructed object.

Mesh Extraction and Texturing

The final stage involves generating high-quality textured meshes from the surface-aligned and segmented Gaussian Splatting model. Following the reconstruction process outlined in SuGaR (Guédon & Lepetit, 2024b), we employ Poisson surface reconstruction (Kazhdan et al., 2006) to convert this model into highly-detailed polygon meshes of up to one million vertices. This approach ensures a smooth high-quality surface, surpassing the results of preliminary experiments using the marching cubes surface reconstruction algorithm.

We sample 3D points on the surface of the Gaussian density field, relying on the depth maps of the Gaussians as seen from the training camera views. Poisson reconstruction is then invoked using Open3D (Zhou, Park, & Koltun, 2018)

to reconstruct a hole-free surface mesh from the point set and their estimated surface normals. We perform surface decimation using the Quadric Error Metric Decimation method (Garland & Heckbert, 1997) implemented in Open3D. This process reduces the polygon count of the mesh to a desired amount while preserving essential geometric features and significantly decreasing the computational burden for subsequent rendering.

For texturing, we use PyTorch3D (Ravi et al., 2020). First, we perform UV unwrapping to map the decimated surface mesh onto a 2D UV map. The resulting UV unwrapped mesh is assigned an initial random texture map to optimize. To mitigate artifacts from reflective non-lambertian materials (e.g. metals, glass, varnished/painted), we re-render the training images using the Gaussian Splatting model with higher spherical harmonics disabled, effectively reproducing the RGB training images with albedo shading properties only. The resulting images I_i serve as a more suitable basis for albedo texture mapping. Using PyTorch3D's differentiable rendering, we render the textured mesh from the viewpoint of a training image yielding R_i and optimize the texture map to minimize the error with the reference I_i

$$\mathcal{L} = \sum_{i \in \mathcal{I}} \|I_i - R_i\|^2 \tag{3}$$

The resulting textures are applied to the mesh, yielding a final textured 3D model that accurately represents the original object without specular lighting properties.

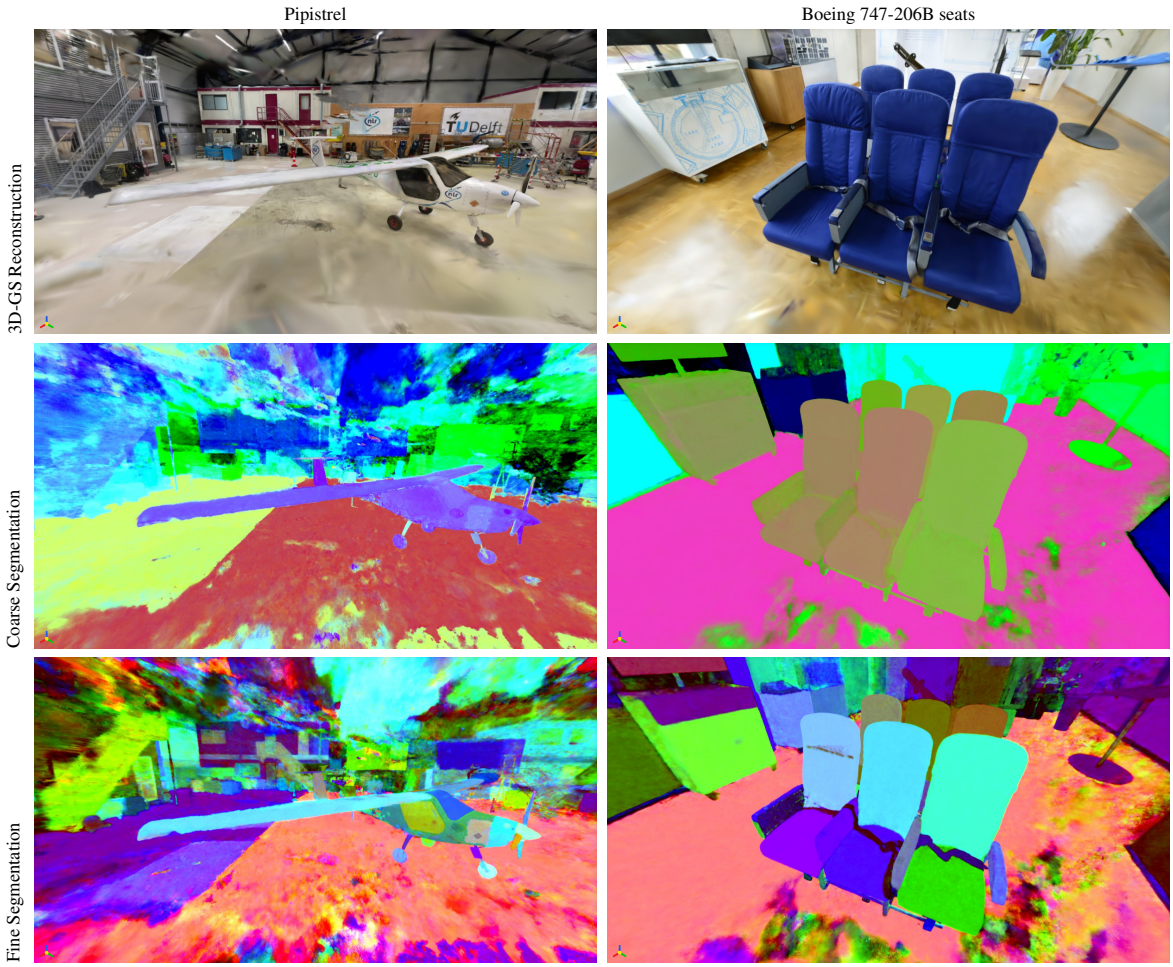


Figure 3. The 3D reconstruction and coarse and fine segmentation of the Pipistrel and Boeing 747-206B seats.

RESULTS

Before developing the RCT for the RNLAF, we evaluate our method on three additional datasets. The first dataset comprises recordings of the Pipistrel Velis Electro, an electric airplane owned by NLR. The second dataset is a set of Boeing 747-206B seats. The final dataset features the Cirrus SR20, also known as `plane` from the Nerfstudio dataset (Tancik et al., 2023). We assess the hierarchical segmentation and our mesh extraction method across these datasets.

Figure 3 shows the trained 3D Gaussian splat together with the 3D object segmentation for the Pipistrel and Boeing 747-206B seats dataset, based on the trained scale-dependent affinity field. On coarse scales, the Pipistrel and set of seats are interpreted as a single object. In the fine scale on the other hand different panels, wheels, wings and propellor blades of the Pipistrel are each interpreted as distinctive objects in the scene. Similarly, the seat belts, armrests and backrests of the seat are all seen as separate components for each individual chair. From the segmentation information we extract textured meshes with 10k polygons as displayed in Figure 4.

Figure 4. Reconstructed meshes obtained from the surface aligned 3D-GS with segmentation information

The APU for the RNLAF was successfully reconstructed. This dataset was particularly challenging as it was recorded ad-hoc without any prior instructions. As such it contained a few challenging artifacts such as moving objects in the background and changing lighting conditions between the images. For the changing background the masking as described in the previous section was crucial to enable correct pose detection. The ability of 3DGS to represent angle dependent viewing effects enabled it to reconstruct the object despite the changing lighting conditions. Renders of the reconstructed APU are available upon request from the authors.

DISCUSSION

The 3D model of the APU has been presented to the RNLAF and is currently being integrated into the Rear Crew Trainer. The results have been met with enthusiasm from our users, they have particularly highlighted the ease and quality of the 3D reconstruction process, which underscores the potential of this technology.

Table 1. Comparison of Mesh-as-a-Service with traditional alternatives.

	3D artist	Structured Light / LiDAR	Photogrammetry	Mesh-as-a-Service
Recording	-	~ Specialized hardware	✓ Consumer camera	✓ Consumer camera
Processing time	× Days	✓ Several minutes	~ 1-7 hours	✓ 2 hours
Man hours / cost	× Expensive	~ Medium expensive	✓ Cheap	✓ Cheap
Details	✓ High detail	~ Medium detail	× Low detail	✓ Good detail
Limitations	× Expensive	× Requires dedicated hardware	× Requires textured surfaces	✓ Flexibility with data
Maturity	✓ Mature	✓ Mature	✓ Mature	~ Emerging technology

Table 1 provides a comparative analysis of Mesh-as-a-Service against traditional 3D modeling methods. It highlights several key advantages, particularly its cost-effectiveness and flexibility. Unlike traditional 3D artists who often specialize in specific areas, such as hard surfaces or soft surfaces, Mesh-as-a-Service offers a highly automated and versatile alternative. The recording process is straightforward, requiring only a consumer-grade camera, and the processing time is relatively quick, making it a flexible option.

In terms of details, traditional 3D artists can produce high quality models, but this comes at a high cost and requires significant man hours. Structured Light and LiDAR also offer decent detail. Photogrammetry, while cheap, often struggles with capturing fine details such as cables and intricate textures. Mesh-as-a-Service provides good detail at a fraction of the cost and time of a 3D artist, making it a highly efficient option.

Regarding limitations, traditional 3D artists are expensive and time-consuming. Structured Light and LiDAR require dedicated hardware, which can be a significant investment and not always available. Photogrammetry requires textured surfaces to function effectively. Mesh-as-a-Service stands out for its flexibility with data; photos can be taken using consumer hardware or even mobile phones without any special instructions, making it accessible and easy to use.

Limitations

While our approach has demonstrated significant potential, several limitations could be addressed to realize its impact:

Image Dataset Quality: The quality of the results is dependent on the comprehensiveness of the image dataset. This can be improved through guided image capture or automation using drones, ensuring a more complete and varied set of images for better reconstruction.

Reflective Surfaces: While our method can handle specular highlights, accurately reconstructing true reflections remains challenging. Reflective surfaces are often reconstructed as semi-transparent, with reflections appearing behind or beneath them. The currently used method of using depth maps to estimate surfaces tends to effectively ignore these semi-transparent surfaces, indicating a potential area for improvement.

Poisson Reconstruction: Poisson reconstruction involves numerous hyperparameters and sometimes produces surfaces with either spurious or insufficient detail, leading to a loss of fidelity in the final output. Automatically selecting the right hyperparameters, as demonstrated in (Guédon & Lepetit, 2024a), or exploring alternative reconstruction methods like (Yu, Sattler, & Geiger, 2024), could enhance the preservation of detailed features.

Although AI based 3D reconstruction and segmenting are novel technologies, we have demonstrated that they are viable for practical, real-world applications. As the technology matures, we expect it to become a valuable tool in 3D modeling and reconstruction.

NEXT STEPS: TOWARDS MESH-AS-A-SERVICE

Many groundbreaking innovations fail to take off because they are not translated into viable, real-world applications. Simply presenting AI techniques and demonstrating their feasibility is not enough; these innovations have to be integrated into practical operational concepts. To address this, we have developed a concept called Mesh-as-a-Service (see Figure 5 for an overview) which packages 3D modeling, segmentation, and mesh extraction into an all-in-one service for generating 3D meshes. In this section, we explore two possible defense-related use cases for operational and training purposes.

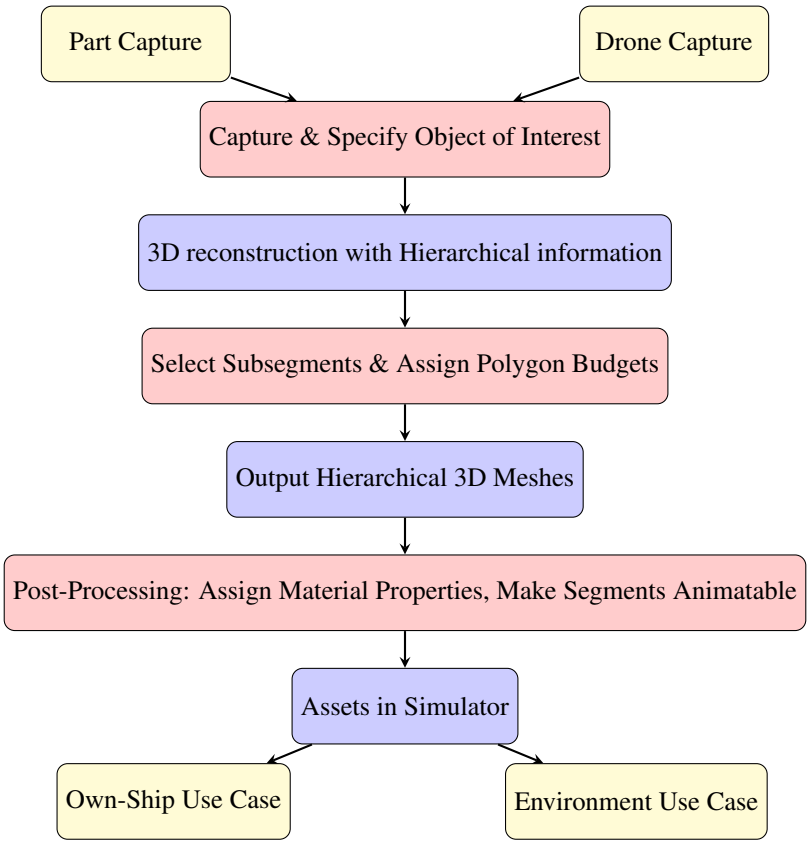


Figure 5. Conceptual workflow diagram for Mesh-as-a-Service and the two use-cases we sketch. Red nodes specify user actions and blue nodes represent steps automated by Mesh-as-a-Service.

Use Case: Own-Ship

This use case involves creating detailed 3D models of platforms like military vehicles, aircraft, or naval vessels, similar to the APU. Users capture images of platform parts without specialized training which are subsequently procssed by the Mesh-as-a-Service system to generate meshes with polygon budgets to meet computational constraints.

An unexplored capability of our pipeline is hierarchical 3D segmentation. This technique could be used to break down the platform into smaller subcomponents. For example, in the case of a cockpit, the segmentation process could create assets with articulatable parts, such as control sticks, buttons, and levers. For the exterior of the platform, hierarchical segmentation divides the structure into logical parts, each of which can be assigned material properties, such as metal or aluminum. This can be used to create accurate thermal and physical simulations.

To fully leverage this use case, we must develop intuitive tools for users to effectively select and define hierarchical subdivisions. Additionally, our system should produce not only 3D meshes but rich simulator objects with embedded material properties. AI algorithms can assist in assigning these properties based on image analysis or predefined criteria. A user-friendly interface should enable users to review, adjust, and manipulate these properties.

This enhances simulation-based training by providing flexibility and ease of adjustment. This means training scenario's can be updated more efficiently to align with platform changes or when learning objectives evolve.

Use Case: Environments

The environments use case focuses on scanning and mapping large environments, such as battlefields, urban areas, or training grounds. Long-range fixed-wing drones can be used to capture large areas, while a swarm of smaller drones can map environments in greater detail. These scans can then be processed centrally to create 3D models of the environment.

3D segmentation can then be used to extract individual assets from the scene like buildings, vehicles, terrain features, and other objects of interest. Each extracted asset and its parts should be assigned material properties, enabling realistic simulations that consider the physical characteristics of the objects. The integration of GNSS data from the images allows the assets to be automatically placed on maps, enhancing the utility for rapid planning and training.

To leverage this use case, we must address the challenge of creating large-scale scenes. Recent developments have shown promising ways to handle this, making it possible to create very large 3D Gaussian models (Ren et al., 2024). We must also develop user-friendly tools that additionally allow users to select and define subdivisions, assign material properties, and link 3D assets to geographical coordinates.

This enhances simulation-based training by enabling rapid building of a database of assets to create scenarios. Quick digital twinning of operational environments allows for the preparation of missions using the most recent data. With a scan from yesterday, teams can train today to execute tomorrow's mission with precision and confidence.

CONCLUSION

In this paper, we demonstrated the effectiveness of integrating state-of-the-art AI methods to create a pipeline for generating 3D meshes from images. We successfully developed an APU model for use in the RCT simulator, highlighting the potential of these innovative 3D reconstruction methods.

To maximize the impact of this study, this pipeline should be packaged into a solution like Mesh-as-a-Service, making rapid and flexible 3D modeling accessible. This will facilitate the creation and updating of 3D models of platforms, ensuring training material remains current; enable the rapid building of a database of 3D assets to create scenarios; and pave the way for next-day digital twinning of mission environments to train for tomorrow's mission. By making our research accessible to the global defense and aerospace community, we envision a future where militaries can leverage AI-driven 3D mesh generation to revolutionize their simulation-based training capabilities.

REFERENCES

Bernardini, F., Mittleman, J., Rushmeier, H., Silva, C., & Taubin, G. (1999). The ball-pivoting algorithm for surface reconstruction. *IEEE transactions on visualization and computer graphics*, *5*(4), 349–359. (On p. 5)

Drysdale, R. S., McElfresh, S., & Snoeyink, J. S. (2001). On exclusion regions for optimal triangulations. *Discrete Applied Mathematics*, *109*(1-2), 49–65. (On p. 5)

Garland, M., & Heckbert, P. S. (1997). Surface simplification using quadric error metrics. In *Proceedings of the 24th annual conference on computer graphics and interactive techniques* (pp. 209–216). (On p. 7)

Guédon, A., & Lepetit, V. (2024a). Gaussian frosting: Editable complex radiance fields with real-time rendering. *arXiv preprint arXiv:2403.14554*. (On p. 10)

Guédon, A., & Lepetit, V. (2024b). Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 5354–5363). (On pp. 3, 4, 5, and 6)

Kazhdan, M., Bolitho, M., & Hoppe, H. (2006). Poisson surface reconstruction. In *Proceedings of the fourth eurographics symposium on geometry processing* (Vol. 7). (On pp. 5 and 6)

Kerbl, B., Kopanas, G., Leimkühler, T., & Drettakis, G. (2023). 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, *42*(4), 1–14. (On p. 3)

Kim, C. M., Wu, M., Kerr, J., Goldberg, K., Tancik, M., & Kanazawa, A. (2024). Garfield: Group anything with radiance fields. *arXiv preprint arXiv:2401.09419*. (On pp. 5 and 6)

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., ... others (2023). Segment anything. In *Proceedings of the ieee/cvf international conference on computer vision* (pp. 4015–4026). (On pp. 5 and 6)

Lorensen, W. E., & Cline, H. E. (1998). Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field* (pp. 347–353). (On p. 5)

Mildenhall, B., Srinivasan, P., Tancik, M., Barron, J., Ramamoorthi, R., & Nerf, R. N. (2021). Representing scenes as neural radiance fields for view synthesis., 2021, 65. *DOI: https://doi. org/10.1145/3503250*, 99–106. (On p. 3)

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., ... others (2021). Learning transferable visual models from natural language supervision. In *Icml 2021* (pp. 8748–8763). (On p. 6)

Ravi, N., Reizenstein, J., Novotny, D., Gordon, T., Lo, W.-Y., Johnson, J., & Gkioxari, G. (2020). Accelerating 3d deep learning with pytorch3d. *arXiv preprint arXiv:2007.08501*. (On p. 7)

Ren, K., Jiang, L., Lu, T., Yu, M., Xu, L., Ni, Z., & Dai, B. (2024). Octree-gs: Towards consistent real-time rendering with lod-structured 3d gaussians. *arXiv preprint arXiv:2403.17898*. (On p. 11)

Schonberger, J. L., & Frahm, J.-M. (2016). Structure-from-motion revisited. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 4104–4113). (On p. 6)

Tancik, M., Weber, E., Ng, E., Li, R., Yi, B., Kerr, J., ... Kanazawa, A. (2023). Nerfstudio: A modular framework for neural radiance field development. In *Acm siggraph 2023 conference proceedings*. (On p. 8)

Ye, Z., Li, W., Liu, S., Qiao, P., & Dou, Y. (2024). Absgs: Recovering fine details for 3d gaussian splatting. *arXiv preprint arXiv:2404.10484*. (On p. 6)

Yu, Z., Sattler, T., & Geiger, A. (2024). Gaussian opacity fields: Efficient high-quality compact surface reconstruction in unbounded scenes. *arXiv preprint arXiv:2404.10772*. (On p. 10)

Zhou, Q.-Y., Park, J., & Koltun, V. (2018). Open3d: A modern library for 3d data processing. *arXiv preprint arXiv:1801.09847*. (On p. 6)