DOI: 10.34749/3061-1466.2026.49

ZeroShop: Automated Metric Mesh Generation for Zero-Shot 6D Object Pose Estimation

Stefan Lechner
Automation and Control Institute
TU Wien
1040 Vienna, Austria
e1608096@student.tuwien.ac.at

Philipp Ausserlechner
Automation and Control Institute
TU Wien
1040 Vienna, Austria
philipp.ausserlechner@tuwien.ac.at

Markus Vincze
Automation and Control Institute
TU Wien
1040 Vienna, Austria
markus.vincze@tuwien.ac.at

Abstract

Robotic manipulation of unseen objects relies on zero-shot 6D pose estimation, which typically requires a 3D mesh as a reference. While constructing accurate meshes requires specialized scanning hardware and manual editing, recently proposed Novel View Synthesis (NVS) techniques, such as 2D Gaussian Splatting (2DGS) and Sparse Voxels Rasterization (SVRaster), produce accurate surface reconstructions as a byproduct, potentially eliminating the need for specialized equipment. This work presents an automated image-based mesh generation pipeline that integrates object segmentation, camera registration, point cloud generation, metric height estimation, and NVS mesh generation, eliminating the need for expensive hardware and human intervention. Leveraging 2DGS and SVRaster with MASt3R-SfM or Visual Geometry Grounded Transformer (VGGT), the pipeline produces accurate meshes in minutes, with the VGGT/SVRaster combination reducing reconstruction time to seconds. Grounding near-view object-centric images with far-view scanning scene images using MASt3R yields consistent object height estimates. On the BOP YCB-V benchmark, meshes generated with our pipeline achieve competitive performance with state-of-the-art zero-shot pose estimation methods. Real-life robotic grasping experiments further indicate robust performance even under moderate scale errors. The source code is available at https://github.com/St333fan/meshgen-zeroshop.

1 Introduction

The rise of Machine Learning (ML) and Generative Artificial Intelligence (GenAI) has significantly enhanced the ability of robotic systems to navigate complex and dynamic environments, moving beyond the constraints of controlled settings [1]. However, planning and locomotion in robots are of limited utility without reliable object detection to facilitate environmental interaction. Although Vision-Language Models are increasingly embedding object representations within their parameters [2], state-of-the-art (SOTA) zero-shot object detection, segmentation, and 6D pose estimation methods still rely on reference objects, typically represented as 3D meshes for feature matching [3] [4] [5]. Figure 1 illustrates this with a robot detecting an object in a real-world scene, segmenting object-specific pixels, and estimating its pose to enable grasp planning and manipulation.

The Third Austrian Symposium on AI, Robotics, and Vision (AIROV26).

This work is licensed under CC BY 4.0, excluding materials attributed to other sources or explicitly excluded.

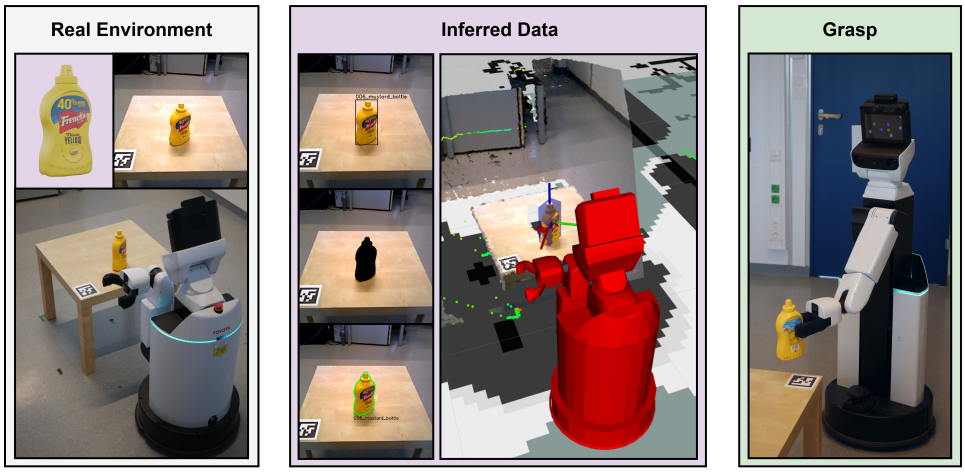

Figure 1: A robot extracts environmental information relevant to a target object for grasping. This data, including image location and corresponding pixel values, is then used to infer both a 2D pose (green) and a 6D pose (blue), defined within the robot coordinate system. Subsequently, this pose information facilitates grasp planning for object manipulation.

Conventionally, meshes are acquired using specialized scanning hardware, followed by manual refinement in a Computer-Aided Design program, or via Structure from Motion (SfM) algorithms [6], which yield comparatively lower-quality results [7]. Unfortunately, hardware scanners are expensive and SfM-based pipelines are highly feature-dependent, struggle with textureless objects, are scale-ambiguous, and require long processing times [6]. Recent ML-based approaches leveraging Vision Transformers for camera registration [8] [9] and scene reconstruction utilizing radiance fields [10] [11] have significantly closed the gap to hardware-based scanning. Combined into an automated pipeline, these methods eliminate the need for specialized scanning hardware and manual mesh generation.

In this work, we present an automated mesh generation pipeline that uses Grounded-SAM [12] for object segmentation, MASt3R-SfM [8] or VGGT [9] for camera pose estimation, and 2DGS [11] or SVRaster [10] for mesh reconstruction, while leveraging MASt3R [13] to estimate object height. All methods were tested on object data generated from the YCB-V subset of the Benchmark for 6D Object Pose Estimation (BOP) [4]. Logically, BOP was also used to compare all meshes within model-based tasks testing with SOTA open-source models CNOS [14], SAM-6D [15] and FoundationPose [16]. Finally, the best-performing mesh generation method was applied to real-world supermarket objects to generate accurate object meshes, which were subsequently validated through robotic grasping experiments.

This paper is organized as follows: Section 2 provides an overview of related work in the domains of 3D reconstruction, NVS, and pose estimation. Section 3 outlines the proposed automated mesh generation pipeline. Section 4 details the experimental setup and presents the results. Finally, Section 5 summarizes the paper and suggests directions for future research.

2 Related Works

The industry-standard reconstruction pipeline utilizes COLMAP [6] for camera registration and sparse point cloud estimation. The resulting point cloud is subsequently densified and meshed using Poisson Reconstruction [17]. While performing best with feature-rich or large objects, it suffers from lengthy processing times and relies on good initialization. Its successor GLOMAP [18] addresses processing time with comparable reconstruction quality, yet both rely on handcrafted features increasingly replaced by ML-based foundation models such as CroCo [19]. Notable integrations include DUSt3R [20], MASt3R [13], and MASt3R-SfM [8], whose learned features are more robust than handcrafted alternatives and increasingly used for feature matching [21] [8]. Importantly, MASt3R, trained on

metric data [13], is able to estimate scene scale; however, its performance is validated primarily for large-scale environments and has received limited testing in small-scale scenarios. Other specifically trained approaches include VGGT [9] and VGGSfM [22] for camera registration and point cloud generation. Currently, MASt3R-SfM and VGGT generate the most viable initial point clouds and camera positions, though both may remain inconsistent for direct mesh reconstruction. On the RealEstate10K benchmark [23], MASt3R-SfM and VGGT perform comparably, while VGGT with Bundle Adjustment (BA) achieves SOTA performance; however, reconstruction quality drops under extreme input rotations [9].

While these methods provide camera positions and an initial sparse point cloud, the resulting data is often incomplete or misaligned. NVS methods address this limitation by optimizing a scene representation from the input images, minimizing the rendering error across all views. The scene is encoded as radiance fields, which can subsequently be used to densify the point cloud [24] [25] [26]. This has been further addressed by 2DGS [11], which applies 2D Gaussian surfaces placed in a 3D space, ensuring alignment with the surfaces of a scene. This results in dense, on-surface-aligned points, where each point represents the midpoint of a Gaussian surface. Alternatively, SVRaster [10] adopts a different approach, initiating reconstruction from registered camera frames and directly generating view-consistent voxels, later fused into a mesh. Sun et al. [10] also presented comparative evaluations of NVS methods, demonstrating a favorable balance between accuracy and inference time for both SVRaster and 2DGS. Beyond rendering alignment, integrated loss functions also leverage object surface alignment guided by segmentation masks, typically extracted using methods such as Segment Anything Model (SAM) [27], Grounded-SAM [12], or Grounding DINO [28]. Following NVS reconstruction, post-processing into a mesh typically involves Poisson Reconstruction [17] or Marching Cubes [29], with texture either registered from source images or increasingly generated via diffusion models [30].

The reconstructed meshes are then used as reference models in model-based zero-shot object detection, segmentation, and 6D pose estimation. Among the available methods benchmarked in BOP [4], three stand out in terms of performance, open-source availability, and ease of integration: CNOS [14], SAM-6D [15], and FoundationPose [16]. CNOS combines DINOv2 [31] for zero-shot detection with SAM [27] or Fast-SAM [32] for segmentation in a straightforward pipeline. SAM-6D [15] extends CNOS by adding a geometric matching term for improved detection and segmentation, with subsequent pose estimation employing a two-stage point matching model. FoundationPose [16] offers a unified framework for 6D pose estimation and tracking of novel objects, supporting both model-based and model-free scenarios via LLM-aided synthetic training and a neural implicit representation. Controversially, the synthetic training data is subject to copyright issues, and a version trained without it has been released, potentially at the cost of reduced accuracy. For subsequent evaluation, the de facto benchmark is BOP [4], offering both 3D object models and real-world counterparts from its YCB-V subset. However, as open-source datasets [4] [33] [34] [35] are often included in model training data, novel data should also be considered.

3 Methodology

Building on the prerequisite of avoiding specialized scanning hardware, the developed meshing pipeline relies solely on a generic camera and ML methods. Figure 2 illustrates the four main stages of object generation: object segmentation, camera registration and point cloud generation, metric height estimation, and meshing supported by NVS. Each step is described in detail in the following sections.

3.1 Data Acquisition and Object Segmentation

To enable reliable 3D reconstruction and accurate segmentation, it is necessary to acquire images that provide complete and uniform coverage of the object surface. A camera should move along a trajectory encircling the object at multiple elevations, consistently maintaining focus on its center. The first image captures the frontal view to establish the coordinate axes. In addition, capturing data as a video stream accelerates acquisition, promotes consistent illumination, and allows flexible frame extraction. Note that capturing the object solely from a top-down perspective leaves a hole at the bottom of the resulting mesh. Once captured, the object is segmented from the background using Grounded-SAM [12], a zero-shot promptable segmentation model. Using the prompt "object in the

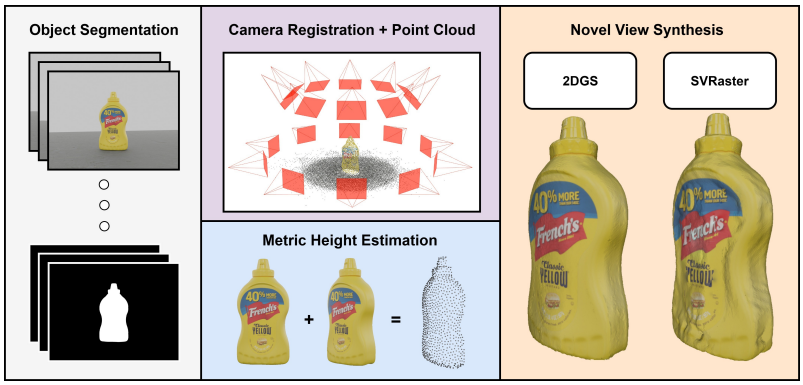

Figure 2: Object generation proceeds in four stages: Object Segmentation generates image masks from a video, followed by Camera Registration and Point Cloud generation, which estimates camera poses and a sparse point cloud. Metric Height Estimation then isolates object-specific 3D points from masked front-view images to estimate the object dimensions. Finally, the point cloud is densified and meshed using the NVS methods 2DGS and SVRaster.

middle." on the first frame, it tracks and generates the object mask across all frames, as seen in Figure 2, enabling accurate segmentation for diverse objects.

3.2 Camera Registration and Point Cloud Generation

Starting from an unordered collection of scene images without known camera intrinsics and extrinsics, it is possible to estimate these parameters while reconstructing 3D scene geometry, as seen in Figure 2. To solve this, MASt3R-SfM integrates an ML stereo model into a conventional SfM framework by using MASt3R features for image retrieval and pairwise matching, followed by camera pose estimation and point cloud reconstruction within a COLMAP-based optimization stage. This hybrid design reduces matching complexity from quadratic to linear, enables robust registration even under purely rotational motion, and avoids reliance on RANSAC, while still benefiting from BA for global consistency. In contrast, VGGT adopts a fully feed-forward approach that jointly processes multiple images to directly predict camera poses, depth maps, point tracks, and 3D point maps in a single forward pass. This yields significantly faster inference and eliminates explicit correspondence search and incremental reconstruction, but requires higher GPU memory and exhibits reduced robustness under large viewpoint changes. While MASt3R-SfM trades runtime efficiency for scalability and accuracy through optimization, VGGT prioritizes real-time performance by learning geometric reasoning end to end from large-scale data.

3.3 Metric Height Estimation

While VGGT does not estimate metric scale, MASt3R-SfM and MASt3R fail to recover the correct scale from near-view object-centric images alone. The solution integrates both near-view object and far-view scene images into a single reconstruction using MASt3R, chosen for its computational efficiency. Since scale is an optimization parameter across image pairs [13], scene images must outnumber object images. A good compromise uses four scene and two object images; the latter being the minimum required by the stereo backbone of MASt3R, producing a point cloud of the full scene with the object registered within it, as shown in Figure 3. Object height is estimated by projecting the point cloud into the reference camera coordinate frame, applying the object mask, and subtracting the lowest from the highest point coordinate. Crucially, scene images can be captured without the object present.

3.4 NVS Mesh Generation

In the last step of Figure 2, a mesh is generated with 2DGS and SVRaster, followed by post-processing with texture, given a pose based on the first reference frame, and scaled to the estimated dimensions. 2DGS and SVRaster utilize the registered cameras, the point cloud, and object masks to iteratively

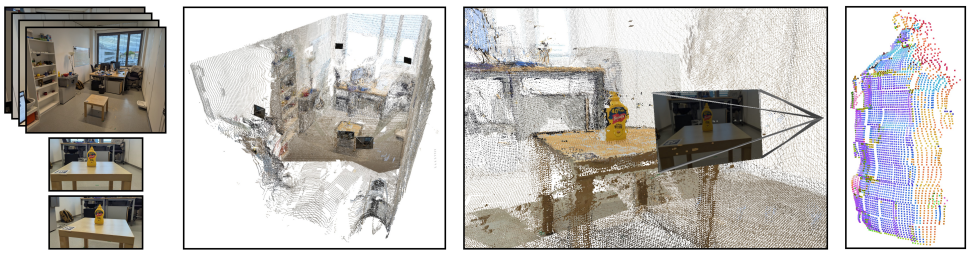

Figure 3: To generate an accurate metric scene point cloud, four scene images (excluding the object) are registered with two images of the object using MASt3R. Subsequently, the first registered image and its corresponding object mask are utilized to extract object-specific points, which are then employed to estimate the height of the scanned object.

reduce loss functions [11] [10], representing the object as densified 2D Gaussian disks or voxels. The radiance fields are then converted into a dense object-specific point cloud and meshed with Poisson Reconstruction (2DGS) or Marching Cubes (SVRaster).

4 Experiments and Results

To assess the quality and robustness of the proposed pipeline, four sub-goals were examined. First, camera registration and point cloud generation with VGGT/MASt3R-SfM and 2DGS/SVRaster to determine which configuration reconstructs the greatest number of YCB-V objects. Second, the MASt3R scaling method was applied to real-world objects and the estimated scale was compared to the true object height. Third, the standardized BOP benchmark was used to validate the object meshes in 2D segmentation and 6D pose estimation tasks. Finally, robotic manipulation was examined in real-world settings using CNOS and FoundationPose.

4.1 Mesh Generation Reconstruction Rate and Quality

To evaluate the automated mesh generation pipeline, the optimal combination of MASt3R-SfM, VGGT, 2DGS, and SVRaster is identified based on reconstruction success. The best configuration is subsequently evaluated on real YCB-V and supermarket objects to assess reconstruction differences using real-world data.

4.1.1 Virtual YCB-V Objects

To establish a baseline for the mesh reconstruction success rate, all 21 virtual YCB-V ground-truth (GT) object models were rendered in BlenderProc [36] to generate scene images. Additionally, scenes were differentiated by high- and low-feature surfaces, with masked object images also included to assess the necessity of surface information for camera registration and NVS. The first two cases comprised 20 images each, while the masked case, unconstrained by surface limitations, yielded a denser representation of 30 images, including views from below. The render scenes are depicted in Appendix A.

The reconstruction success achieved per combination is detailed in Table 1, where both registration methods exhibit distinct strengths. In particular, MASt3R-SfM with high-feature surfaces is the only combination achieving successful reconstructions for all objects. Although VGGT surpassed MASt3R-SfM in the segmented task, it failed to reconstruct the featureless YCB-V bowl object. Figure 4 visualizes qualitative differences in mesh quality between 2DGS (left) and SVRaster (right) across three geometrically distinct objects, using MASt3R-SfM initialization. In summary, SVRaster generates meshes with object dimensions comparable to 2DGS but with a less smooth surface, while training in seconds compared to minutes for 2DGS.

4.1.2 Real Supermarket Objects

Given that only MASt3R-SfM successfully reconstructed all virtual YCB-V object scenes, it was selected for evaluation on real-world objects. As detailed in Section 3.1, a video was recorded,

Table 1: 2DGS and SVRaster mesh reconstruction success initiated by VGGT and MASt3R-SfM, utilizing the rendered data from the 21 BOP YCB-V GT objects. The surface task is divided into a low- and high-feature surface, and seg indicates registration utilizing object masks.

	MASt3R-SfM			VGGT		
	low	high	seg	low	high	seg
2DGS	20	21	16	20	18	20
SVRaster	20	21	15	19	18	20

Figure 4: Overview of the reconstruction quality of three geometrically distinct YCB-V objects. For each pair, the left object is from 2DGS and the right from SVRaster.

20 equally spaced frames were extracted, and processed through the pipeline. All objects were successfully reconstructed as meshes; examples are depicted in Figure 5 with 2DGS (left) and SVRaster (right). Upon application of the texture, no apparent differences are visible. All generated meshes and an example of surface quality are provided in Appendix B.

Figure 5: Overview of the reconstruction quality of objects that are used in the grasp evaluation. For each pair, the left mesh is from 2DGS and the right from SVRaster.

4.2 Object Scaling

The height of real YCB-V and supermarket objects were estimated following the procedure from Section 3.3; the results are shown in Figure 6 with the real height plotted against the percentage error. The results indicate an estimation error within $\pm 10\%$ for most objects, with a tendency for estimation error to decrease with increasing object height. A notable exception is the "toothbrush", which exhibits a disproportionately large error. The complete results are provided in Appendix C.

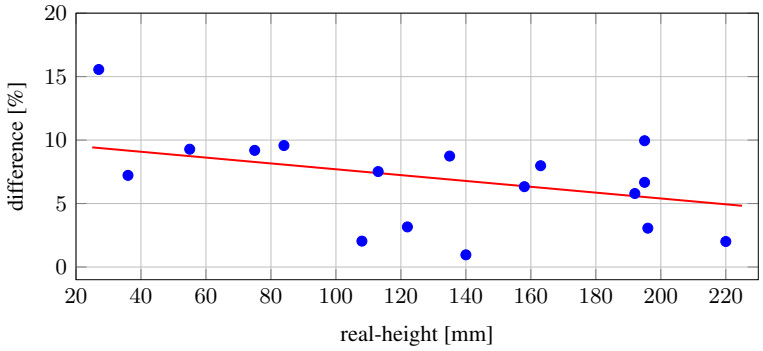

Figure 6: The relationship between the real object height and the absolute measurement error in percent for real YCB-V and supermarket objects. Individual measurements are represented as blue points, with a red line indicating the trend of decreasing error as object height increases.

4.3 BOP with NVS Meshes

To evaluate the suitability of the reconstructed meshes for object segmentation and pose estimation, a benchmark validation was conducted using the BOP framework. Official YCB-V performance scores from previous work [4] serve as a reference. As a first step, the Average Precision (AP) and Average Recall (AR) [37] scores reported in the original papers using virtual GT objects from the YCB-V dataset were reproduced. Subsequently, the reconstructed objects from Section 4.1.1 were used to benchmark each method and assess whether comparable accuracy could be maintained. These reconstructed objects were generated with BlenderProc scenes, they possess an arbitrary scale and were therefore aligned to their corresponding GT counterparts using the ICP algorithm to ensure the correct height. Importantly, the CNOS mask filtering used in the original FoundationPose implementation was not publicly available and had to be re-implemented. As a result, GT masks were also used to evaluate FoundationPose. To analyze how the generated meshes affect the AP score for segmentation and the AR score for pose estimation in relation to the official BOP meshes, four distinct scenarios were defined:

- CNOS on GT/NVS meshes, with FastSAM
- Instance Segmentation Model (ISM) SAM-6D on GT/NVS meshes, with SAM
- Pose Estimation Model (PEM) SAM-6D with ISM masks on GT/NVS meshes
- FoundationPose with GT/CNOS masks on GT/NVS meshes

AP/AR scores are presented in Table 2, divided into segmentation and pose estimation tasks, with the submission row reporting the official scores of the BOP benchmark. Focusing on the segmentation task, the AP scores of both methods, CNOS and SAM-6D ISM, were successfully reproduced. When employing NVS-generated meshes, all methods exhibit a comparable decrease in performance, with only a marginal deviation relative to the scores obtained using SVRaster meshes. When using FoundationPose on GT masks, the GT BOP meshes achieve performance that exceeds the official AR score. Furthermore, the 2DGS and SVRaster meshes exhibit only a minor decrease in performance. When employing CNOS-based segmentation, the accuracy decreased across all evaluated scenarios. In addition, the official reported scores could not be fully reproduced, and the performance gap between the GT BOP meshes and the 2DGS/SVRaster meshes increased further. In comparison, the official SAM-6D PEM scores were almost reproducible and exhibited a smaller performance gap

compared to 2DGS and SVRaster than FoundationPose/CNOS. However, the difference between 2DGS and SVRaster is slightly greater than that observed in the FoundationPose evaluation.

Table 2: Official YCB-V BOP AP (segmentation) and AR (pose estimation) scores compared to reproduced scores across different mesh datasets. FoundationPose was additionally benchmarked with GT masks because the official filtering of the CNOS masks is not publicly available.

	Segmentation		Pose Estimation		
	CNOS	ISM (SAM-6D)	FoundationPose		PEM (SAM-6D)
Submission	0.599	0.605	0.882		0.845
			GT	CNOS	
BOP	0.6	0.603	0.915	0.731	0.832
2DGS	0.56	0.561	0.889	0.543	0.751
SVRaster	0.541	0.567	0.877	0.539	0.71

4.4 Robotic Object Manipulation

After evaluating mesh quality, the influence of the height estimation of Section 3.3 on segmentation and pose estimation remains to be examined. Therefore, we evaluated real reconstructed meshes within a robotic grasp pipeline. Based on preliminary experiments, FoundationPose with CNOS/SAM was adopted for pose estimation and evaluated with a grasp success test using the meshes depicted in Figure 5, each manually annotated with grasp positions. The test scenario is defined as follows: each object is positioned at the same location in front of the robot, within its grasping range, and then rotated four times by 90° around the z-axis. Subsequently, the object is flipped onto its top and rotated four times again, resulting in a total of eight distinct poses per object. An attempt to grasp is classified as successful if the robot establishes a stable grasp and lifts the object. In instances where the pipeline fails at any stage, it is re-executed; this repeated execution is defined as a re-grasp.

Considering only a single grasp attempt, the pipeline using 2DGS meshes achieved a grasp success rate of 85.9%, while the pipeline using SVRaster achieved 89.1%. When re-grasps were included, the success rate for 2DGS increased to 93.8%, while that for SVRaster increased to 90.6%. The two primary outlier cases were the inverted "toothbrush", which was not detectable using CNOS, and the upright "soap", for which FoundationPose consistently estimated the pose as lying horizontally. In addition, the following observations were made: CNOS exhibited difficulty in segmenting the "gelatin_box"; the robot lost its otherwise stable grasp on the "mustard_bottle" twice during lifting; and the "razors," which were meshed in a compressed configuration with an underestimated object height, appear to be affected by the cumulative error in one specific pose.

5 Conclusion

The proposed automated mesh generation pipeline, which integrates Grounded-SAM, MASt3R-SfM/VGGT, and 2DGS/SVRaster, consistently produces geometrically accurate meshes that allow accurate zero-shot object pose estimation. Grasping experiments conducted with a robot and CNOS/FoundationPose demonstrated that the NVS-generated meshes and their associated scaling are adequate for real-world object manipulation. However, this approach has limitations when applied to deformable, low-profile, or semi-transparent objects. These shortcomings are primarily attributable to scaling inaccuracies, incomplete mesh reconstruction, or unreliable robotic sensory data.

Future work will focus on improving the reconstruction of occluded surfaces and extending the pipeline to transparent objects such as glass and plastic. Another promising research direction involves rendering NVS in conjunction with depth information predicted by ML depth models and using GenAI models for missing parts. Finally, testing the generated meshes in a supermarket, where hundreds of objects appear across long, interconnected tasks, would further validate the robustness of our approach.

Acknowledgments and Disclosure of Funding

This work was supported by the EU-program EC Horizon 2020 for Research and Innovation under grant agreement No. 101120823 project MANiBOT funded by the European Union.

References

[1] Ranjan Sapkota, Yang Cao, Konstantinos I Roumeliotis, and Manoj Karkee. Vision-language-action models: Concepts, progress, applications and challenges. *arXiv preprint arXiv:2505.04769*, 2025.

[2] Yiming Zuo, Karhan Kayan, Maggie Wang, Kevin Jeon, Jia Deng, and Thomas L Griffiths. Towards foundation models for 3d vision: How close are we? *arXiv preprint arXiv:2410.10799*, 2024.

[3] Felix Gorschlüter, Pavel Rojtberg, and Thomas Pöllabauer. A survey of 6d object detection based on 3d models for industrial applications. *Journal of imaging*, 8(3):53, 2022.

[4] Van Nguyen Nguyen, Stephen Tyree, Andrew Guo, Mederic Fourmy, Anas Gouda, Taeyeop Lee, Sungphill Moon, Hyeontae Son, Lukas Ranftl, Jonathan Tremblay, et al. Bop challenge 2024 on model-based and model-free 6d object pose estimation. *arXiv preprint arXiv:2504.02812*, 2025.

[5] Shibiao Xu, Shunpeng Chen, Rongtao Xu, Changwei Wang, Peng Lu, and Li Guo. Local feature matching using deep learning: A survey. *Information Fusion*, 107:102344, 2024.

[6] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016.

[7] MW Wilkinson, Robin R Jones, Christopher E Woods, SR Gilment, Ken JW McCaffrey, Sotiris Kokkalas, and JJ Long. A comparison of terrestrial laser scanning and structure-from-motion photogrammetry as methods for digital outcrop acquisition. *Geosphere*, 12(6):1865–1880, 2016.

[8] Bardienus Pieter Duisterhof, Lojze Zust, Philippe Weinzaepfel, Vincent Leroy, Yohann Cabon, and Jerome Revaud. Mast3r-sfm: a fully-integrated solution for unconstrained structure-from-motion. In *2025 International Conference on 3D Vision (3DV)*, pages 1–10. IEEE, 2025.

[9] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025.

[10] Cheng Sun, Jaesung Choe, Charles Loop, Wei-Chiu Ma, and Yu-Chiang Frank Wang. Sparse voxels rasterization: Real-time high-fidelity radiance field rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16187–16196, 2025.

[11] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 conference papers*, pages 1–11, 2024.

[12] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024.

[13] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European conference on computer vision*, pages 71–91. Springer, 2024.

[14] Van Nguyen Nguyen, Thibault Groueix, Georgy Ponimatkin, Vincent Lepetit, and Tomas Hodan. Cnos: A strong baseline for cad-based novel object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2134–2140, 2023.

[15] Jiehong Lin, Lihua Liu, Dekun Lu, and Kui Jia. Sam-6d: Segment anything model meets zero-shot 6d object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 27906–27916, 2024.

[16] Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17868–17879, 2024.

[17] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, 2006.

[18] Linfei Pan, Dániel Baráth, Marc Pollefeys, and Johannes L Schönberger. Global structure-from-motion revisited. In *European Conference on Computer Vision*, pages 58–77. Springer, 2024.

[19] Philippe Weinzaepfel, Thomas Lucas, Vincent Leroy, Yohann Cabon, Vaibhav Arora, Romain Brégier, Gabriela Csurka, Leonid Antsfeld, Boris Chidlovskii, and Jérôme Revaud. Croco v2: Improved cross-view completion pre-training for stereo matching and optical flow. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17969–17980, 2023.

[20] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024.

[21] Riku Murai, Eric Dexheimer, and Andrew J Davison. Mast3r-slam: Real-time dense slam with 3d reconstruction priors. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16695–16705, 2025.

[22] Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. Vggsfm: Visual geometry grounded deep structure from motion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21686–21697, 2024.

[23] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018.

[24] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[25] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5438–5448, 2022.

[26] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

[27] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[28] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pages 38–55. Springer, 2024.

[29] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pages 347–353. ACM, 1998.

[30] Zibo Zhao, Zeqiang Lai, Qingxiang Lin, Yunfei Zhao, Haolin Liu, Shuhui Yang, Yifei Feng, Mingxin Yang, Sheng Zhang, Xianghui Yang, et al. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *arXiv preprint arXiv:2501.12202*, 2025.

[31] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[32] Xu Zhao, Wenchao Ding, Yongqi An, Yinglong Du, Tao Yu, Min Li, Ming Tang, and Jinqiao Wang. Fast segment anything. *arXiv preprint arXiv:2306.12156*, 2023.

[33] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision*, 120:153–168, 2016.

[34] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10901–10911, 2021.

[35] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, et al. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 803–814, 2023.

[36] Maximilian Denninger, Martin Sundermeyer, Dominik Winkelbauer, Youssef Zidan, Dmitry Olefir, Mohamad Elbadrawy, Ahsan Lodhi, and Harinandan Katam. Blenderproc. *arXiv preprint arXiv:1911.01911*, 2019.

[37] Tomáš Hodaň, Martin Sundermeyer, Bertram Drost, Yann Labbé, Eric Brachmann, Frank Michel, Carsten Rother, and Jiří Matas. Bop challenge 2020 on 6d object localization. In *European Conference on Computer Vision*, pages 577–594. Springer, 2020.

A Rendered BlenderProc Scenes

Figure 7: YCB-V object rendering with high/low feature surface and extracted segmented RGB masks; rendering angles of $0°$, $45°$, and $-45°$ to the horizontal plane.

B Real Reconstructed Objects

Figure 8: Reconstructed YCB-V objects, with the left/upper object illustrating reconstruction using 2DGS, followed by SVRaster.

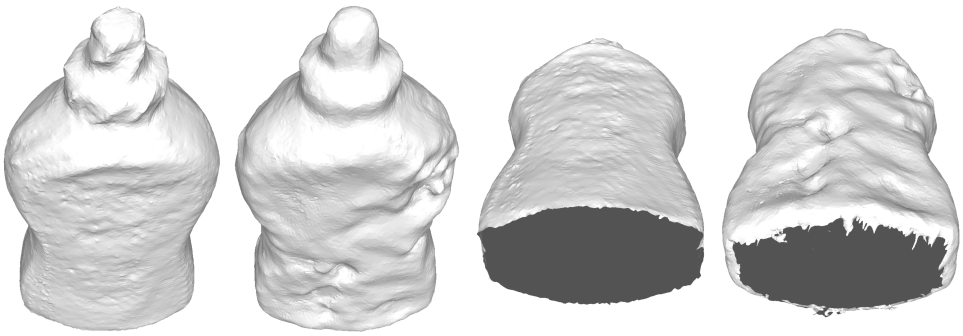

Figure 9: Mesh quality comparison between 2DGS and SVRaster on real-world data, showing top and bottom views, with 2DGS yielding more accurate surfaces.

Figure 10: Reconstructed supermarket objects, with the left/upper object illustrating reconstruction using 2DGS, followed by SVRaster. Upon application of the texture, no apparent differences become visible.

C Metric Height Estimation

Table 3: Comparison between the actual heights of YCB-V objects and those estimated using the MASt3R registration method.

	height [m]	mast3r-height [m]	difference [mm]	difference [%]
mustard_bottle	0.192	0.2031	11.12	5.79
potted_meat_can	0.084	0.092	8.04	9.57
bowl	0.055	0.0601	5.1	9.28
cracker_box	0.22	0.2244	4.42	2.01
master_chef_can	0.14	0.1413	1.34	0.96
gelatin_box	0.075	0.0819	6.89	9.19
large_marker	0.122	0.1259	3.85	3.16
extra_large_clamp	0.036	0.0334	-2.6	-7.22

Table 4: Comparison between the actual heights of the supermarket objects and those estimated using the MASt3R registration method.

	height [m]	mast3r-height [m]	difference [mm]	difference [%]
soap	0.158	0.148	-10.0	-6.33
ahorn_sirup	0.163	0.15	-13.0	-7.98
tomato_paste	0.195	0.208	13.0	6.67
kokos_can	0.113	0.1215	8.5	7.52
hand_cream	0.135	0.1468	11.8	8.74
wet_wipes	0.108	0.1058	-2.2	-2.04
razors	0.195	0.1756	-19.4	-9.95
balsamic	0.196	0.202	6.0	3.06
toothbrush	0.027	0.0312	4.2	15.56