

Subsecond 3D Mesh Generation for Robot Manipulation

Qian Wang, Omar Abdellall*, Tony Gao*, Xiatao Sun, Daniel Rakita
 Department of Computer Science, Yale University
 {peter.wang.qw262, omar.abdellall, t.gao, xiatao.sun, daniel.rakita}@yale.edu

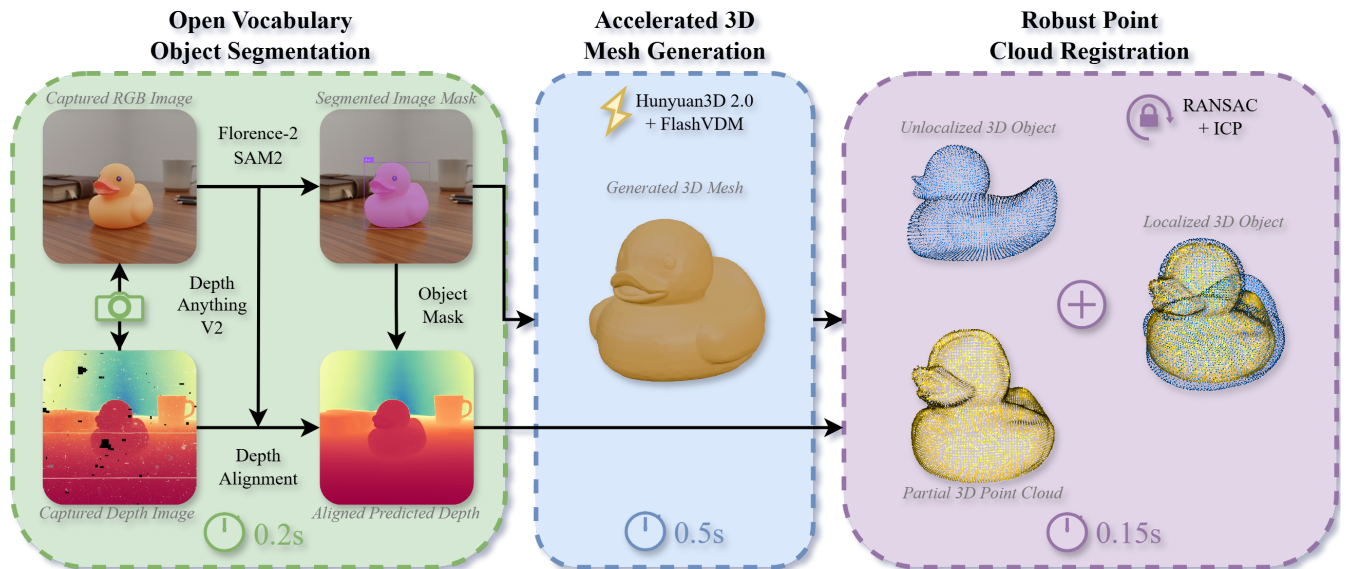


Fig. 1: Our system for sub-second 3D mesh generation from RGB-D input. The system combines three stages: (1) Open-vocabulary segmentation using Florence-2 and SAM2 with depth enhancement via Depth Anything v2 (0.2s), (2) Accelerated mesh generation using FlashVDM-distilled Hunyuan3D 2.0 (0.5s), and (3) Object registration via RANSAC and ICP to align the mesh with observed point cloud (0.15s). The 0.85s total runtime marks a critical step toward real-time robotic applications.

Abstract—3D meshes are a fundamental representation widely used in computer science and engineering. In robotics, they are particularly valuable because they capture objects in a form that aligns directly with how robots interact with the physical world, enabling core capabilities such as predicting stable grasps, detecting collisions, and simulating dynamics. Although automatic 3D mesh generation methods have shown promising progress in recent years, potentially offering a path toward real-time robot perception, two critical challenges remain. First, generating high-fidelity meshes is prohibitively slow for real-time use, often requiring tens of seconds per object. Second, mesh generation by itself is insufficient. In robotics, a mesh must be *contextually grounded*, i.e., correctly *segmented* from the scene and *registered* with the proper scale and pose. Additionally, unless these contextual grounding steps remain efficient, they simply introduce new bottlenecks. In this work, we introduce an end-to-end system that addresses these challenges, producing a high-quality, contextually grounded 3D mesh from a single RGB-D image in under one second. Our contribution is a system level design that integrates open-vocabulary object

segmentation, accelerated diffusion-based mesh generation, and robust point cloud registration, each optimized for both speed and accuracy. We demonstrate its effectiveness in a real-world manipulation task, showing that it enables meshes to be used as a practical, on-demand representation for robotics perception and planning. Open-source code and videos are located at the paper website.¹

I. INTRODUCTION

3D meshes are a fundamental representation in computer science and engineering. Defined by vertices, edges, and faces, they encode explicit object geometry and topology in a form that supports direct physical reasoning [6]. Unlike representations such as images, voxel grids, signed distance functions, or radiance fields, meshes provide an explicit, physics-ready blueprint of the world, enabling algorithms to operate directly on geometry and surface detail.

In robotics, a mesh aligns closely with how robots must interact with the world, supporting essential tasks such as predicting stable grasps, planning collision-free trajectories,

* Equal contribution
 This work was supported by Office of Naval Research award N00014-24-1-2124

¹<https://apollo-lab-yale.github.io/26-ICRA-subsecond-mesh-gen-website/>

and simulating contact dynamics [36, 16]. If objects in a robot’s environment could be represented as high-fidelity meshes, much of the complexity of perception would simplify into a representation that is easily usable by downstream planning and control strategies.

Despite the representational power of 3D meshes in robotics, translating this representation to real-world environments at scale is a significant challenge. Manually created meshes, as used in graphics and game development, are impractical for robotics: robots must operate in unstructured, ever-changing environments where new objects continually appear. Early robotic systems often circumvented this challenge by assuming a finite set of known objects, using fiducial markers like AprilTags [28] or database lookups of CAD models [21]. Although effective in constrained scenarios, such approaches are fundamentally unscalable in open-world environments.

Recent advances in generative modeling have begun to close this gap, producing detailed 3D meshes directly from RGB images. However, two critical challenges remain. First, high-fidelity mesh generation is prohibitively slow for real-time use, often requiring tens of seconds per object. Second, mesh generation alone is not sufficient in robotics. In this context, a mesh must be *contextually grounded*, i.e., correctly *segmented* from the scene and *registered* with the proper scale and pose. Compounding this challenge, unless these additional contextual grounding steps remain efficient, they simply introduce new bottlenecks.

In this paper, we introduce the first end-to-end system capable of generating a high-quality, contextually grounded (i.e., segmented and registered) 3D mesh from a single RGB-D image in under one second. Our pipeline combines three tightly integrated components: (1) open-vocabulary segmentation to identify and isolate objects of interest; (2) an accelerated diffusion-based model to produce high-fidelity meshes; and (3) robust point cloud registration to recover the correct scale and pose within the scene.

The mesh generation stage is based on Hunyuan3D 2.0 [37], but, to overcome its prohibitive latency for robotics, we incorporate FlashVDM [14], which applies progressive flow distillation to reduce the number of diffusion steps from dozens to as few as three. To further accelerate inference, we adopt hierarchical SDF decoding with adaptive key–value selection, reducing volumetric decoding costs by over 90%. Together, these techniques deliver generative fidelity comparable to full diffusion pipelines while reducing runtime from tens of seconds to well under one second. By unifying these advances into a single low-latency pipeline, our system transforms mesh acquisition from a slow, offline process into a practical, on-demand tool toward real-time robotics.

To demonstrate the efficacy of our system, we conduct an evaluation with three sub-experiments (§IV). First, we analyze the runtime of each component to confirm sub-second performance and identify bottlenecks. Second, we perform ablation studies comparing alternative designs for segmentation, generation, and registration, highlighting the trade-offs between speed and accuracy. Finally, we demon-

strate the system in a real-world manipulation task, showing that it enables reliable on-demand grasping and placement using generated meshes. These results collectively establish that fast, contextually grounded mesh generation is not only feasible but also effective for real-time robotic interaction. We conclude with a discussion of the limitations and implications of our work.

II. RELATED WORKS

Our proposed pipeline combines modules for object detection and segmentation, single-image 3D mesh generation, and object localization. This section reviews each of these areas and contextualizes our methodological choices.

A. Single-Image 3D Mesh Generation

Generating a 3D object from a single 2D image with both quality and speed is a long-standing challenge, defined by a fundamental tradeoff between inference speed and geometric quality. Early works [11, 29, 33] established foundational model architectures but suffered from either slow inference or low geometric fidelity.

More recently, large-scale generative models have reached a new level of quality. Models like Hunyuan3D 2.0 (H3D) [37] can generate high-resolution, textured 3D assets, while others like TRELIS [31] leverage structured priors to improve geometric consistency. However, the primary drawback of such models is their slow generation speed. Models like TripoSR [27], SF3D [5], and MeshRAG [24] focused on improving efficiency, but this speed often came at the cost of reduced quality and limited generalization to novel object categories. To break this quality-speed trade-off, our system builds upon H3D’s high-fidelity generation principles but applies two key modifications for acceleration. First, we integrate FlashVDM [14] to speed up the generation process. Second, we deliberately omit texture generation to focus solely on fast, high-quality shape reconstruction—a design choice motivated by the observation that most robotic tasks (grasping, collision avoidance, dynamics simulation) primarily require geometric rather than visual information.

B. Open-Vocabulary Object Segmentation

To generate a 3D mesh of a specific object, it must first be precisely segmented from the background. The field of universal image segmentation has produced powerful models like OneFormer [13]. Our work employs the Segment Anything Model 2 (SAM2) [19], which currently represents the state-of-the-art, offering strong zero-shot generalization for both images and videos. However, SAM2 requires an initial point, bounding box, or mask to identify the target object. In a robotic system operating autonomously, this prompting must be automated through object detection.

Traditional object detection has seen rapid progress, from real-time architectures like YOLOv12 [26] to Transformer-based models like DINOv2 [17], which have achieved higher accuracy on standard benchmarks. While effective, these models are typically trained on a fixed set of classes, fundamentally limiting their flexibility in open-world robotic

scenarios where novel objects frequently appear. The solution lies in Vision-Language Models (VLMs) that can detect objects from free-form text prompts. These models exist on a spectrum of scale and specialization. At the high end, massive generalist models like Flamingo [2] (80B parameters) and PaLM-E [9] (562B parameters) demonstrate impressive multi-modal reasoning but are computationally prohibitive for real-time pipelines. In contrast, more specialized and efficient models have been developed for targeted vision tasks. For instance, Florence-2 [32] (0.23B parameters) excels at grounding text in images for which the small size makes it ideal for a performance-critical system.

C. Object Localization and Spatial Grounding

Once a 3D mesh is generated, it must be localized within the robot’s coordinate frame, recovering both the correct scale and 6D pose relative to the observed scene. Recent literature has converged on render-and-refine methods [18] for this task. FoundationPose [30], for instance, performs 6D pose estimation by iteratively aligning rendered views of an object with sensor data, which has been deployed in multiple robotic systems [1, 23, 25]. However, these methods fundamentally rely on photorealistic rendering, a process which requires a textured mesh. As our pipeline deliberately forgoes texture generation to maximize speed, these render-based approaches are incompatible with our output.

Therefore, we employ direct geometric alignment via point cloud registration. We employ RANSAC [10], a fast and robust method for model fitting in the presence of outliers. We also investigated more recent registration algorithms, such as TEASER++ [34] for certifiable optimality and BUFFER-X [22] for zero-shot generalization; our empirical evaluations (§IV) demonstrate that a well-tuned RANSAC implementation provides an effective combination of speed, accuracy, and robustness for our specific task.

III. METHOD

Our system generates a registered, high-fidelity 3D mesh from a single RGB-D image in under one second. As illustrated in Fig. 1, the system consists of three sequential stages: open-vocabulary segmentation, accelerated mesh generation, and robust point cloud registration.

A. Open Vocabulary Image Segmentation

The first stage of the pipeline identifies and segments objects of interest from the input image. To enable flexible, open-vocabulary segmentation, we leverage the VLM *Florence-2* [32], which supports multiple segmentation paradigms. In robotics contexts, two functionalities of *Florence-2* are particularly relevant: (1) prompt-free, generic object-level segmentation and (2) text-guided object or category-level segmentation. In this work, we primarily use the text-guided mode. Given a prompt specifying target object classes (e.g., “fruits”) or regions (e.g., “object on the table”), *Florence-2* generates labeled bounding boxes for each instance. These serve as coarse outlines, which are then refined using SAM2 [19], yielding precise, pixel-level

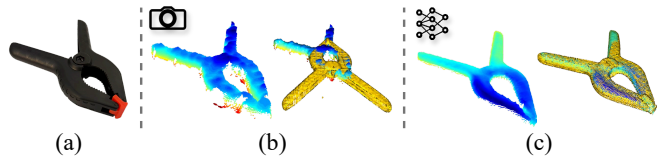


Fig. 2: Effect of Depth Map Quality on Registration. (a) The source object. (b) A noisy point cloud from the raw depth camera feed results in a failed registration (misaligned overlap). (c) The cleaner, more coherent point cloud predicted by DAV2 enables a successful and accurate registration.

instance masks. Finally, these masks are then used to crop both the RGB image and its associated depth map, producing segmented object images with transparent backgrounds and corresponding partial depth maps.

B. Processing Depth Input

While the RGB image drives mesh generation, accurate depth is essential for scaling and localizing the result. Raw measurements from consumer-grade depth cameras suffer from noise, missing, regions, and artifacts on reflective or textureless surfaces (Fig. 2b). To address this issue, we employ Depth Anything v2 (DAV2) [35], a monocular depth estimation model that predicts clean, geometrically consistent depth maps from RGB images.

However, monocular predictions lack metric scale. We resolve this issue through median-based alignment:

$$s = \frac{\text{median}(D_{\text{sensor}} \cap M)}{\text{median}(D_{\text{DAV2}} \cap M)}, \quad D_{\text{metric}} = s \cdot D_{\text{DAV2}} \quad (1)$$

where M is the object mask, and the median is computed only over valid (non-zero) depth values. We use median rather than mean as it is robust to the depth outliers common in consumer sensors. This approach combines the geometric consistency of learned depth with the metric accuracy of physical measurements, providing a reliable foundation for downstream stages in the pipeline.

C. Vecset Diffusion Model for 3D Mesh Generation

For 3D shape generation, we build on *Hunyuan3D 2.0 (H3D)* [37], a state-of-the-art flow-based diffusion transformer designed for high-fidelity mesh synthesis from a single image. The framework consists of two main components: a Shape VAE that compresses 3D meshes into a structured latent space, and a conditional diffusion model that generates these latents from a single image.

The Shape VAE consists of an encoder \mathcal{E} and a decoder \mathcal{D} . The encoder maps a 3D mesh M , represented as a Signed Distance Function (SDF), into a compact set of N latent vectors $Z = \{z_i\}_{i=1}^N \in \mathbb{R}^{N \times D}$, where $Z = \mathcal{E}(M)$. The decoder reconstructs the SDF from these latents: $\widehat{\text{SDF}} = \mathcal{D}(Z)$. The VAE is trained to minimize a reconstruction loss, typically a weighted L2 norm on the SDF values, combined with a KL-divergence regularization term on the latent distribution.

The generative model is a flow-based diffusion transformer, conditioned on image features c extracted by a

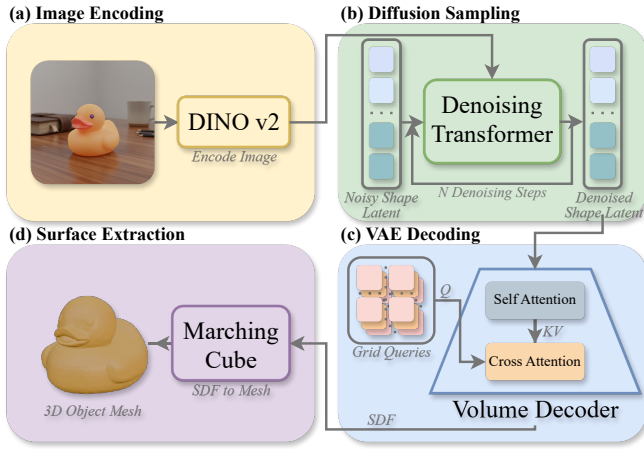


Fig. 3: Standard Vecset Diffusion Model (VDM) pipeline for 3D mesh generation. The process consists of four stages: (a) image encoding using DINOv2 to extract visual features, (b) iterative diffusion sampling requiring N denoising steps (typically 50+), (c) VAE decoding to convert latent vectors to a dense SDF volume, and (d) mesh extraction via marching cubes. The two primary bottlenecks preventing real-time performance are the iterative diffusion process (b) and volumetric decoding (c).

DINOv2 [17] backbone (seen in Fig. 3(a)). It learns to transform a simple prior distribution p_0 (e.g., a standard normal distribution $\mathcal{N}(0, I)$) into the complex data distribution of shape latents $p_1(Z|c)$. This transformation is learned via a time-dependent vector field $v_\theta(Z_t, t, c)$ that follows the probability path from noise to data, governed by the ordinary differential equation (ODE):

$$\frac{dZ_t}{dt} = v_t(Z_t, c), \quad (2)$$

where Z_t is the latent representation at time $t \in [0, 1]$. A flow-matching objective is used to train the model, which provides a stable and efficient regression target for the vector field. The loss function is formulated as:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{t, p_1(Z_1|c), p_0(Z_0)} \left[\|v_\theta(Z_t, t, c) - (Z_1 - Z_0)\|_2^2 \right] \quad (3)$$

where Z_1 is a ground-truth latent from the data distribution, Z_0 is a sample from the noise distribution, and the term $(Z_1 - Z_0)$ is the vector field of a linear probability path.

D. Accelerating 3D Generation

The standard diffusion model inference process, outlined above, requires solving an ODE over many steps, leading to runtimes far too slow for robotics (often tens of seconds per mesh). We overcome this limitation by integrating the FlashVDM (FVDM) framework [14], which provides both diffusion acceleration and volume decoding optimizations. The following paragraphs summarize the key FVDM techniques we adopt.

1) *Vector-Set Diffusion Acceleration*: Diffusion sampling (Fig. 3b) often requires dozens of sequential denoising steps, making it a dominant runtime cost. Following FVDM, we

employ *Progressive Flow Distillation* to distill the multi-step teacher model into a student capable of producing comparable results in as few as 3 steps where the original H3D model requires at least 50 steps.

2) *Volume Decoding Acceleration*: The second bottleneck arises in decoding (Fig. 3(c)), where shape latents must be converted into a dense SDF volume (e.g., 384^3 voxels). A naive implementation requires millions of costly cross-attention evaluations. Following FVDM, we adopt two complementary acceleration techniques: 1) *Hierarchical Volume Decoding*: Instead of decoding the entire high-resolution grid, a coarse grid is decoded first, and only the voxels near the object’s surface are refined. 2) *Adaptive KV Selection*: The locality of latent tokens is exploited by using a lightweight attention pass to pre-select only the most relevant key-value pairs for each query point.

E. Object Registration

The final stage aligns the canonically generated mesh with the object’s real-world observation. We formulate this process as a rigid point cloud registration problem. Let the source point cloud derived from our generated mesh be $P_S = \{\mathbf{p}_i^S\}_{i=1}^{N_S} \subset \mathbb{R}^3$ and the target point cloud from the depth sensor be $P_T = \{\mathbf{p}_j^T\}_{j=1}^{N_T} \subset \mathbb{R}^3$. The goal is to find an optimal scale $s \in \mathbb{R}^+$, rotation $R \in SO(3)$, and translation $\mathbf{t} \in \mathbb{R}^3$ that best align P_S with P_T .

Our registration procedure proceeds as follows:

1) *Initial Scaling*: We first uniformly scale P_S to match the bounding box diagonal of P_T .

2) *Feature Extraction and Matching*: We compute Fast Point Feature Histograms (FPFH) [20] for both point clouds. For each point \mathbf{p}_i , FPFH computes a k -dimensional feature descriptor $\mathbf{f}(\mathbf{p}_i) \in \mathbb{R}^k$ that encodes the local geometry of its neighborhood. We then establish a set of putative correspondences $\mathcal{C} = \{(\mathbf{p}_i^S, \mathbf{p}_i^T)\}_{i=1}^N$ by finding mutual nearest neighbors in this high-dimensional feature space.

3) *Robust Pose Estimation with RANSAC*: The correspondence set \mathcal{C} from FPFH matching contains a significant fraction of outliers. To robustly estimate the transformation, we employ RANSAC (Random Sample Consensus) [10]. The algorithm iteratively performs the following steps: 1) A minimal subset of correspondences (e.g., 3 pairs) is randomly selected from \mathcal{C} . 2) A rigid transformation (R, \mathbf{t}) is computed from this minimal sample. 3) The hypothesis is tested against all other correspondences in \mathcal{C} . Correspondences that agree with the transformation (i.e., where $\|sR\mathbf{p}_i^S + \mathbf{t} - \mathbf{p}_i^T\|_2$ is below a distance threshold) are counted as inliers. 4) Repeat: This process is repeated for a fixed number of iterations, and the transformation with the largest inlier set is chosen as the best hypothesis. This iterative process discards outliers and provides a robust initial alignment for the two point clouds.

4) *Refinement with ICP*: The estimate from RANSAC is used to initialize an Iterative Closest Point (ICP) [4] algorithm for fine-grained refinement. ICP iteratively performs two steps until convergence: (1) For each transformed source point $sR\mathbf{p}_i^S + \mathbf{t}$, find its closest neighbor in the target cloud,

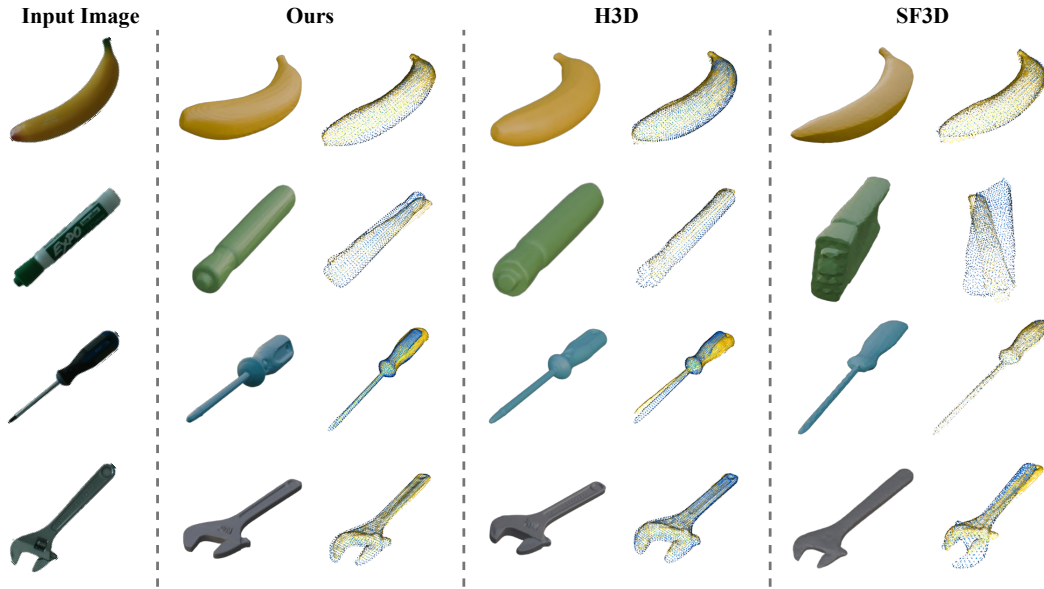


Fig. 4: Qualitative Comparison of Generated Meshes and Registration Results. Our method (left) achieves high geometric quality nearly identical to the slow H3D baseline (middle), while the fast SF3D baseline (right) produces significant artifacts.

$\mathbf{p}_i^{T'} \in P_T$; and (2) Given correspondences $\{(x_i, y_i)\}_{i=1}^N$ with $x_i = sR\mathbf{p}_i^S + \mathbf{t}$ fixed from the previous iteration, solve

$$\min_{R', \mathbf{t}'} \sum_{i=1}^N \|R'x_i + \mathbf{t}' - y_i\|_2^2 \quad \text{s.t. } R' \in SO(3). \quad (4)$$

Let $\bar{x} = \frac{1}{N} \sum_i x_i$, $\bar{y} = \frac{1}{N} \sum_i y_i$, and $x'_i = x_i - \bar{x}$, $y'_i = y_i - \bar{y}$. Form $H = \sum_i x'_i y'_i{}^\top$ and compute its SVD $H = U\Sigma V^\top$. The optimal update is

$$R' = V \text{diag}(1, 1, \det(VU^\top)) U^\top, \quad (5)$$

$$\mathbf{t}' = \bar{y} - R'\bar{x}, \quad (6)$$

the standard orthogonal Procrustes solution [3, 12].

This process yields an accurate and robust alignment of the generated mesh in the scene, ready for downstream manipulation tasks.

IV. EVALUATION

We evaluate our system on three fronts: (1) component latency and bottlenecks, (2) comparisons against alternative designs, and (3) a real-world robotic manipulation task. Experiments were run on a workstation with AMD 7960X CPU, 256GB RAM, and a RTX 5000 Ada GPU.

A. Component-wise Time Breakdown

We measure the runtime of each stage’s sub components, averaged over 10 runs on 25 YCB objects [7]. As shown in Table I, the entire pipeline, from a single RGB image to a fully registered mesh, executes in 824 ms on average. Mesh generation dominates the runtime at 500ms (60.7%), with the distilled diffusion process being the primary contributor. The segmentation and registration stages are well-optimized at 184ms and 140ms respectively, with no single component creating a prohibitive bottleneck.

The relatively high variance in registration time (± 61 ms) stems from object complexity—simple geometric primitives converge quickly while complex, asymmetric objects require more RANSAC iterations. Despite this variance, even worst-case scenarios remain under our 1-second target.

TABLE I: Component Runtime Breakdown. Average runtime and percentage of total pipeline time for each subcomponent, measured over 250 trials.

Pipeline Stage	Time (ms)	Time (%)
Open-Vocabulary Segmentation	184 ± 7.3	22.3%
- Florence-2 Detection	(55 ± 2.5)	(6.7%)
- SAM2 Mask Refinement	(14 ± 0.3)	(1.7%)
- Depth Anything v2	(93 ± 6.3)	(11.3%)
Accelerated 3D Mesh Generation	500 ± 37	60.7%
- DINOv2 Image Encoding	(10 ± 0.5)	(1.2%)
- Distilled Diffusion (3 steps)	(225 ± 2)	(27.3%)
- Hierarchical SDF Decoding	(104 ± 3.4)	(12.6%)
- Marching Cube (Res: 128)	(43 ± 22)	(5.2%)
Object Registration	140 ± 61	17%
- FPFH Feature Extraction	(8 ± 4)	(1%)
- RANSAC Pose Estimation	(79 ± 20)	(9.6%)
- ICP Refinement	(57 ± 53)	(7%)
Total End-to-End Time	824 ± 95	

B. Ablation Studies

1) *Procedure*: We systematically evaluate alternative components for each pipeline stage using 25 YCB objects. For each alternative, we replace only the specified component while keeping others fixed, enabling direct assessment of each design choice’s impact on the complete system.

2) *Metrics*: In addition to average component time, component GPU peak memory, and total time, we evaluate the geometric quality of the final registered mesh against its ground-truth localized model using two widely adopted metrics from previous works [5]: 1) Chamfer Distance (CD);

TABLE II: Ablation Study. Each row evaluates a complete pipeline, with checkmarks indicating the active module for each configuration. The results validate that our system (top row) achieves the optimal balance of speed and geometric accuracy.

Depth Input		Mesh Generation					Registration			Metrics				
DAv2 + Align (ours)	DAv2	Depth Cam.	H3D + FVDM (ours)	H3D	TRELLIS	SF3D	RANSAC (ours)	TEASER++	BUFFER-X	Component time (s)	Component GPU Peak Memory (GB)	Total time (s)	Mesh Chamfer Distance (mm)	F-Score (%)
✓			✓				✓			0.09 ± 0.01	1.96	0.83 ± 0.10	0.45 ± 0.90	89.9 ± 21.8
	✓		✓				✓			0.09 ± 0.01	1.96	0.83 ± 0.09	106 ± 51.1	0.12 ± 0.41
		✓	✓				✓			0.03 ± 0.01	N/A	0.77 ± 0.11	0.62 ± 1.29	87.7 ± 23.6
✓			✓				✓			0.50 ± 0.04	5.11	= row1	= row1	= row1
✓				✓			✓			30.1 ± 0.17	7.45	30.6 ± 0.27	0.34 ± 0.65	90.6 ± 19.5
✓					✓		✓			4.41 ± 1.08	10.6	4.78 ± 1.12	44.2 ± 118	59.8 ± 34.2
✓						✓	✓			0.53 ± 0.03	6.17	0.83 ± 0.07	0.79 ± 1.00	75.2 ± 28.4
✓			✓				✓			0.14 ± 0.06	N/A	= row1	= row1	= row1
✓			✓					✓		0.08 ± 0.05	N/A	0.78 ± 0.09	0.79 ± 1.22	77.5 ± 25.0
✓			✓						✓	0.28 ± 0.13	1.05	0.98 ± 0.10	118 ± 159	49.9 ± 43.8

This metric computes the average symmetric closest-point distance between two point clouds, providing a general measure of surface deviation. 2) F-Score: The harmonic mean of precision and recall, calculated at a 2cm threshold. This measures the percentage of the mesh surface that is accurately reconstructed. Furthermore, these two metrics together give a sense of how the object’s pose is misaligned against ground truth.

3) *Results*: The results, summarized in Table II, validate that our chosen components achieve the best overall balance of speed, accuracy, and efficiency.

Our DAV2 + alignment approach is critical for accurate registration. Using DAV2 alone fails catastrophically (106mm CD) due to lack of metric scale. Raw depth is faster but noisier, resulting in slightly degraded registration quality (0.62mm vs. 0.45mm CD).

Results from the mesh generation component validate our design choice to adopt FVDM’s acceleration of H3D for robotic applications. The FVDM-accelerated H3D achieves quality nearly identical to the original (89.9% vs. 90.6% F-Score) while being 37× faster—a critical improvement that makes subsecond mesh generation feasible. SF3D matches our speed but sacrifices too much quality (75.2% F-Score). TRELLIS is both slower and less accurate than ours.

For Registration, our classical PPFH + RANSAC + ICP (RANSAC) approach provides the most accurate and robust alignment. PPFH + TEASER++ + ICP (TEASER++) is marginally faster (80ms vs. 140ms) but less accurate. The learning-based BUFFER-X suffers from significant generalization errors, making it unsuitable for reliable deployment.

These quantitative findings are further supported by our qualitative results in Fig. 4, showing that our pipeline is a carefully optimized system, with each component chosen not only for individual performance but also for its contribution to the overall speed-accuracy balance.

C. Real-world Online Pick-and-Place

1) *Procedure*: We validate our system’s practicality through a sequential pick-and-place task using a UFactory

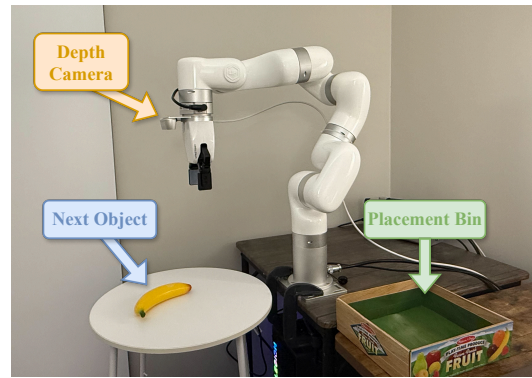


Fig. 5: Real-world manipulation setup. The xArm7 robot uses our generated meshes to grasp and transfer unknown objects. The system processes each object in real-time without prior CAD models.

XArm7 robot (Fig. 5). The robot must perceive novel objects placed on a table, generate their 3D meshes, plan collision-free grasps, and transfer them to a target bin.

2) *Implementation Details*: Once an object’s mesh is generated and registered in the camera frame, its pose is transformed to the robot’s base frame using a pre-calibrated extrinsic matrix. To find a stable grasp, we use a simple sampling-based planner sufficient for this demonstration; the method samples a set of end-effector poses by perturbing the translation and rotation around the object’s centroid, retaining the first collision-free grasp found. Before execution, promising grasps are validated in a PyBullet simulation [8] to filter out kinematically unreachable poses. Finally, a collision-free trajectory is computed using the RRT algorithm [15], with the generated mesh used for collision checking.

TABLE III: Pick-and-Place Performance. Our system achieves both high reliability and fastest task completion.

System Configuration	Total Success Rate	Makespan (s)
Ours	92%	122
SF3D	60%	129
H3D	96%	416

3) *Metrics*: We evaluate system performance using two complementary metrics: 1) Total success rate: Percentage of objects successfully grasped, lifted, and placed in the target bin without drops or collisions. 2) Makespan: Total time to complete all 10 sequential pick-and-place operations, from initial perception to final placement.

4) *Results*: Table III demonstrates our system’s practical advantages. While H3D achieves marginally higher success (96% vs. 92%), its 30-second generation time makes the complete task take 3.4× longer (416s vs. 122s). This delay would be prohibitive in dynamic environments where objects frequently change. SF3D’s poor mesh quality leads to cascading failures: inaccurate geometry causes registration errors, which result in grasps missing the actual object surface. Of its 10 failures, 7 were complete misses and 3 were unstable grasps that slipped during lifting.

V. DISCUSSION

Our work demonstrates that high-quality, contextually grounded 3D meshes can be generated from single RGB-D images in under one second, making them viable for real-time robotic applications. By integrating open-vocabulary segmentation, FlashVDM-accelerated mesh synthesis, and robust geometric registration, we achieve both the speed required for interactive systems and the geometric fidelity needed for manipulation tasks. Real-world experiments validate this balance—our system completed pick-and-place tasks in 122 seconds with 92% success, significantly faster than prior methods while maintaining reliability.

A. Limitations

Several constraints bound the current system’s applicability. First, we omit texture generation to maintain sub-second performance, limiting applications requiring appearance information for semantic reasoning or human interaction. Second, the system assumes full object visibility and fails under heavy occlusion or partial views, though our speed enables reactive regeneration as viewpoints change. Third, our mesh representation only handles rigid objects, excluding deformable materials common in many robotic tasks. Finally, while robust in controlled settings, deployment in cluttered, unstructured environments will require improvements in segmentation and depth estimation.

Despite these limitations, this work establishes that sub-second mesh generation is both technically feasible and practically valuable for robotics. The ability to generate meshes on-demand opens new possibilities for scene understanding and sim-to-real transfer, while our integration of graphics advances into robotics suggests further untapped opportunities from generative AI. As these technologies mature, we anticipate rapid 3D reconstruction becoming a fundamental capability for interactive robotic systems.

REFERENCES

- [1] Aditya Agarwal, Gaurav Singh, Bipasha Sen, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. Scenecomplete: Open-world 3d scene completion in cluttered real world environments for robot manipulation. *arXiv preprint arXiv:2410.23643*, 2024.
- [2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.
- [3] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(5):698–700, 1987.
- [4] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. Spie, 1992.
- [5] Mark Boss, Zixuan Huang, Aaryaman Vasishtha, and Varun Jampani. Sf3d: Stable fast 3d mesh reconstruction with uv-unwrapping and illumination disentanglement. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16240–16250, 2025.
- [6] Mario Botsch, Leif Kobbelt, Mark Pauly, Pierre Alliez, and Bruno Lévy. *Polygon mesh processing*. CRC press, 2010.
- [7] Berk Calli, Arjun Singh, James Bruce, Aaron Walsman, Kurt Konolige, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. Yale-cmu-berkeley dataset for robotic manipulation research. *The International Journal of Robotics Research*, 36(3):261–268, 2017.
- [8] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. <http://pybullet.org>, 2016.
- [9] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Ayzan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, et al. Palm-e: An embodied multimodal language model. 2023.
- [10] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [11] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. In *The Twelfth International Conference on Learning Representations*.
- [12] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642, 1987.
- [13] Jitesh Jain, Jiachen Li, Mang Tik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. Oneformer: One transformer to rule universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2989–2998, 2023.
- [14] Zeqiang Lai, Yunfei Zhao, Zibo Zhao, Haolin Liu,

- Fuyun Wang, Huiwen Shi, Xianghui Yang, Qingxiang Lin, Jingwei Huang, Yuhong Liu, et al. Unleashing vecset diffusion model for fast shape generation. *arXiv preprint arXiv:2503.16302*, 2025.
- [15] Steven LaValle. Rapidly-exploring random trees: A new tool for path planning. *Research Report 9811*, 1998.
- [16] Yilang Liu, Haoxiang You, and Ian Abraham. Sample-based hybrid mode control: Asymptotically optimal switching of algorithmic and non-differentiable control modes, 2025. URL <https://arxiv.org/abs/2510.19074>.
- [17] 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.
- [18] Evin Pinar Örnek, Yann Labbé, Bugra Tekin, Lingni Ma, Cem Keskin, Christian Forster, and Tomas Hodan. Foundpose: Unseen object pose estimation with foundation features. In *European Conference on Computer Vision*, pages 163–182. Springer, 2024.
- [19] 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.
- [20] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *2009 IEEE international conference on robotics and automation*, pages 3212–3217. IEEE, 2009.
- [21] Kulunu Samarawickrama, Gaurang Sharma, Alexandre Anglraud, and Roel Pieters. 6d assembly pose estimation by point cloud registration for robotic manipulation. In *2024 IEEE 20th International Conference on Automation Science and Engineering (CASE)*, pages 846–853. IEEE, 2024.
- [22] Minkyun Seo, Hyungtae Lim, Kanghee Lee, Luca Carlone, and Jaesik Park. Buffer-x: Towards zero-shot point cloud registration in diverse scenes. *arXiv preprint arXiv:2503.07940*, 2025.
- [23] Xiatao Sun, Yinxing Chen, and Daniel Rakita. Prism-dp: Spatial pose-based observations for diffusion-policies via segmentation, mesh generation, and pose tracking. *arXiv preprint arXiv:2504.20359*, 2025.
- [24] Xiatao Sun, Chen Liang, Qian Wang, and Daniel Rakita. Mesh rag: Retrieval augmentation for autoregressive mesh generation, 2025. URL <https://arxiv.org/abs/2511.16807>.
- [25] Xiatao Sun, Yuxuan Wang, Shuo Yang, Yinxing Chen, and Daniel Rakita. Hybrid diffusion policies with projective geometric algebra for efficient robot manipulation learning. *arXiv preprint arXiv:2507.05695*, 2025.
- [26] Yunjie Tian, Qixiang Ye, and David Doermann. Yolov12: Attention-centric real-time object detectors. *arXiv preprint arXiv:2502.12524*, 2025.
- [27] Dmitry Tochilkin, David Pankratz, Zexiang Liu, Zixuan Huang, Adam Letts, Yanguang Li, Ding Liang, Christian Laforte, Varun Jampani, and Yan-Pei Cao. Triposr: Fast 3d object reconstruction from a single image. *arXiv preprint arXiv:2403.02151*, 2024.
- [28] John Wang and Edwin Olson. Apriltag 2: Efficient and robust fiducial detection. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4193–4198. IEEE, 2016.
- [29] Zhengyi Wang, Yikai Wang, Yifei Chen, Chendong Xiang, Shuo Chen, Dajiang Yu, Chongxuan Li, Hang Su, and Jun Zhu. Crm: Single image to 3d textured mesh with convolutional reconstruction model. In *European conference on computer vision*, pages 57–74. Springer, 2024.
- [30] 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.
- [31] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. *arXiv preprint arXiv:2412.01506*, 2024.
- [32] Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu, and Lu Yuan. Florence-2: Advancing a unified representation for a variety of vision tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4818–4829, 2024.
- [33] Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191*, 2024.
- [34] Heng Yang, Jingnan Shi, and Luca Carlone. Teaser: Fast and certifiable point cloud registration. *IEEE Transactions on Robotics*, 37(2):314–333, 2020.
- [35] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in Neural Information Processing Systems*, 37:21875–21911, 2024.
- [36] Haoxiang You, Yilang Liu, and Ian Abraham. Accelerating visual-policy learning through parallel differentiable simulation, 2025. URL <https://arxiv.org/abs/2505.10646>.
- [37] 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.