Open-World Semantic-Based Zero-Shot 6D Pose Estimation Using SAM3 And FoundationPose

Het Patel*, Sunny Deshpande*, Ansh Bhansali*, Keisuke Ogawa*
Computer Vision CS543
University of Illinois Urbana-Champaign
Urbana, IL, USA
hcp4, sunnynd2, anshb3, ogawa3@illinois.edu

***Abstract*—Accurate and real-time 6D object pose estimation is a critical prerequisite for autonomous robotic manipulation, yet conventional pipelines often lack the flexibility to adapt to unstructured environments or changing user intents. Specifically, existing methods typically rely on pre-registered CAD models, limiting their ability to handle unseen objects or switch targets dynamically based on natural language instructions. In this work, we propose a novel open-vocabulary 6D tracking framework that extends the state-of-the-art FoundationPose architecture. Our approach integrates Moondream2, a lightweight vision-language model for edge-compatible scene understanding, with SAM-3 to enable precise, text-driven target segmentation. To overcome the reliance on pre-existing asset libraries, we implement an on-the-fly geometric proxy generation module that synthesizes 3D meshes for detected objects in real time. This cohesive pipeline allows a robotic system to semantically index scene constituents, generate necessary 3D assets zero-shot, and robustly track 6D pose even under significant occlusion. Experimental results demonstrate that our system supports seamless, dynamic target switching via natural language prompts, significantly enhancing the autonomy and interaction capabilities of robotic manipulators in novel environments.**

***Index Terms*—6D Pose Estimation, Vision-Language Models (VLM), Zero-Shot Tracking, Robotic Manipulation, Open-Vocabulary Perception, FoundationPose**

I. Introduction

6D object pose estimation is a foundational technology for robotic manipulation. To enable robots to follow language commands (e.g., “grasp the red bottle” to “now grasp the blue cup”) and switch targets flexibly while maintaining grasping operations, real-time, occlusion-robust, and language-guided pose estimation and tracking are required.

NVIDIA’s FoundationPose (CVPR 2024) [1] achieves zero-shot inference for unseen objects by simply providing a CAD model (or reference images) at test time and has maintained strong performance in the BOP Challenge’s model-based unseen object tasks. However, it requires masking in the initial frame with manual annotation or an external detector (typically R-CNN-based), and often fails in heavily occluded scenes (e.g., LineMOD) which propagate errors to subsequent mesh-matching and refinement stages.

Furthermore, it can handle only objects with pre-provided CAD models; arbitrary unseen objects in the scene cannot be instantly targeted via language for pose estimation and tracking. Although various models have emerged in the BOP unseen tasks since FoundationPose, incorporating foundation models to improve occlusion handling and generalization, no prior work has achieved real-time, language-guided, multi-object pose estimation and tracking with dynamic target switching.

II. Related Work

A. Vision-Language Models for Open-Vocabulary Perception

Recent advancements in VLMs have enabled systems to detect and segment objects using open-ended text descriptions rather than fixed category lists. Models such as GLIP and Grounding DINO effectively align text embeddings with image regions to perform open-vocabulary object detection. Similarly, SAM (Segment Anything Model) and its successor SAM-2 have revolutionized class-agnostic segmentation.

However, these foundation models operate primarily in the 2D domain, outputting bounding boxes or segmentation masks. They do not inherently provide the 6D pose information (rotation and translation) required for robotic manipulation, necessitating a bridge between 2D semantic understanding and 3D geometric tracking.

B. Zero-Shot 6D Pose Estimation

Traditional 6D pose estimation methods often rely on training instance-specific networks (e.g., PoseCNN, DOPE) or require fine-tuning on target objects. To address scalability, recent “zero-shot” approaches like MegaPose and FoundationPose [1] have emerged. FoundationPose, in particular, utilizes a render-and-compare architecture with a transformer-based refiner to track novel objects given only a CAD model or reference images at test time.

While highly effective, these methods remain “passive”; they depend on the user explicitly providing the 3D asset for every target object. They lack an active mechanism to semantically parse a scene and autonomously acquire the necessary 3D models for tracking, a gap our work aims to fill.

C. Text-to-3D Generation and Retrieval

Acquiring 3D assets for unseen objects is a bottleneck for model-based tracking. Generative approaches like Shap-E and TripoSR [3] can synthesize 3D meshes from single images or text prompts. Alternatively, retrieval-based methods leverage massive 3D datasets such as Objaverse-XL [6], which contains over 10 million assets.

While generative methods offer novelty, they often suffer from geometric artifacts or scale ambiguity. Retrieval methods, conversely, provide high-quality geometry but face challenges with instance-level mismatch. Our framework integrates these retrieval capabilities directly into a pose estimation loop, effectively using retrieved meshes as “geometric proxies” for robust tracking.

III. Proposed Method

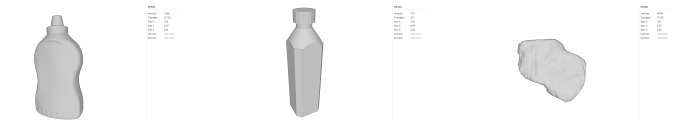
We propose an Open-Vocabulary 6D Pose Tracking framework that extends the robust render-and-compare architecture of FoundationPose. By integrating a VLM with zero-shot mesh acquisition, our system eliminates the requirement for pre-registered CAD models. The pipeline operates in four distinct stages:

A. Semantic Scene Analysis

We use the lightweight vision-language model Moondream2 [8] (suitable for edge devices) to perform captioning and detection on each frame's RGB image. Upon initialization, the VLM analyzes the RGB video stream to generate a dense semantic caption of the scene. This process yields a discrete list of candidate objects (e.g., "red bottle," "blue cup"), effectively establishing a dynamic semantic inventory of the environment without manual annotation. By listing objects with Moondream2 and generating meshes on-the-fly, prompt-based specification becomes possible even for objects without CAD models in BOP unseen settings.

B. On-the-Fly 3D Mesh Generation

For novel objects, we employ a three-step hierarchical mesh acquisition strategy to obtain suitable 3D proxies. First, if precomputed meshes are available (as in benchmark datasets such as YCB-Video), the system directly loads the ground-truth CAD models. When these are absent, the pipeline queries the Objaverse-XL database [6] using the object labels generated in the previous stage to retrieve the closest matching template meshes via language-guided similarity search. Additionally, for comparison and selection, we also generate a candidate mesh using TripoSR from the single observed image. As an optional refinement step, Moondream2 can be invoked again to produce more detailed object descriptions, improving retrieval quality for ambiguous cases. A dedicated mesh manager asynchronously fetches and caches these high-quality, clean meshes in a "Mesh Dictionary" to eliminate redundant queries. From the available candidates: ground-truth when present, Objaverse-retrieved, and TripoSR-generated, the manager selects the one with the highest matching score based on silhouette, depth, and IoU alignment during initial pose scoring for downstream pose estimation. This hierarchical approach ensures geometrically accurate proxies are readily available while supporting multiple candidate meshes when needed for more robust downstream pose estimation. In Fig.1 mustard image below, (a) uses the pre-provided mesh from the YCB dataset, (b) shows a mesh generated from Objaverse, and (c) presents a mesh generated using TripoSR.



(a) Benchmark (b) Objaverse (c) TripoSR

Fig. 1: Reconstructed mustard bottle meshes

C. Language-Driven Segmentation

To bridge the gap between user intent and pixel-level tracking, we integrate Meta's SAM-3 (Segment Anything Model 3) [4]. This module accepts natural language prompts (e.g., "the red apple") to output high-precision, temporally consistent segmentation masks.

SAM-3 is robust to occlusion, capable of handling multiple objects via text, and supports video tracking. Traditional R-CNN-based detectors often fail in heavy clutter or occlusion. By replacing them with SAM-3, mask accuracy improves in occluded scenes, resulting in more reliable mesh matching and refinement.

D. Unified 6D Pose Estimation & Tracking

The core tracking engine utilizes FoundationPose to solve for the 6D pose (Rotation R, Translation t). The system fuses the semantic mask from SAM-3 with the generated mesh from the Mesh Dictionary. FoundationPose then performs uniform sampling, scoring, and iterative refinement to align the mesh with the video observation. In tracking mode, this pipeline achieves real-time performance.

A central novelty is language-guided dynamic target switching: when a new prompt is issued mid-task (e.g., switching from "red bottle" to "blue cup"), the system instantly updates the active mask (via SAM-3) and mesh (from the pre-cached dictionary), seamlessly transitioning tracking without reinitialization or interruption. This modular design, combined with the lightweight nature of Moondream2 and the efficiency of SAM-3, preserves real-time feasibility while supporting multi-object scenarios.

IV. Experiments

To achieve robust prompt-based dynamic pose estimation for arbitrary unseen objects without pre-provided CAD models, we conducted extensive experiments on mesh generation strategies. The core challenge was to reliably obtain high-quality 3D meshes for all relevant objects in the scene, as FoundationPose relies on accurate mesh models for render-and-compare and refinement stages. Poor mesh quality or incomplete object coverage directly leads to inaccurate pose estimation.

A. Initial Approach: YOLO + SAM + TripoSR Pipeline

Our first attempt focused on on-the-fly 3D reconstruction from single keyframes. The pipeline was as follows:

1) Object detection: Use YOLOv8 [7] to detect objects in the RGB frame and crop bounding boxes with confidence $>$ 0.9.
2) Precise segmentation: Apply SAM to refine the cropped regions, removing background and obtaining pixel-level masks.
3) Mesh reconstruction: Feed the masked 2D image into TripoSR [3], a fast single-image-to-3D model, to generate a 3D mesh.

While this approach theoretically enables CAD-free reconstruction, it exhibited some limitations in practice. Firstly, YOLOv8 failed to detect all objects in complex scenes, often missing less common or partially occluded items (e.g., in an office scene, it reliably detected keyboards, mice, and chairs but frequently overlooked bags, plants, or small accessories). Additionally, the quality of mesh became poor on low-resolution or noisy inputs. When video resolution was moderate (e.g., 720p), the reconstructed meshes were blurry, deformed, or contained severe artifacts (concave/convex distortions). These low-quality meshes caused errors in subsequent render-and-compare scoring and refinement, leading to unstable or incorrect pose estimates.

Due to these issues, we abandoned direct reconstruction-based mesh generation, as inaccurate meshes propagated errors throughout the FoundationPose pipeline.

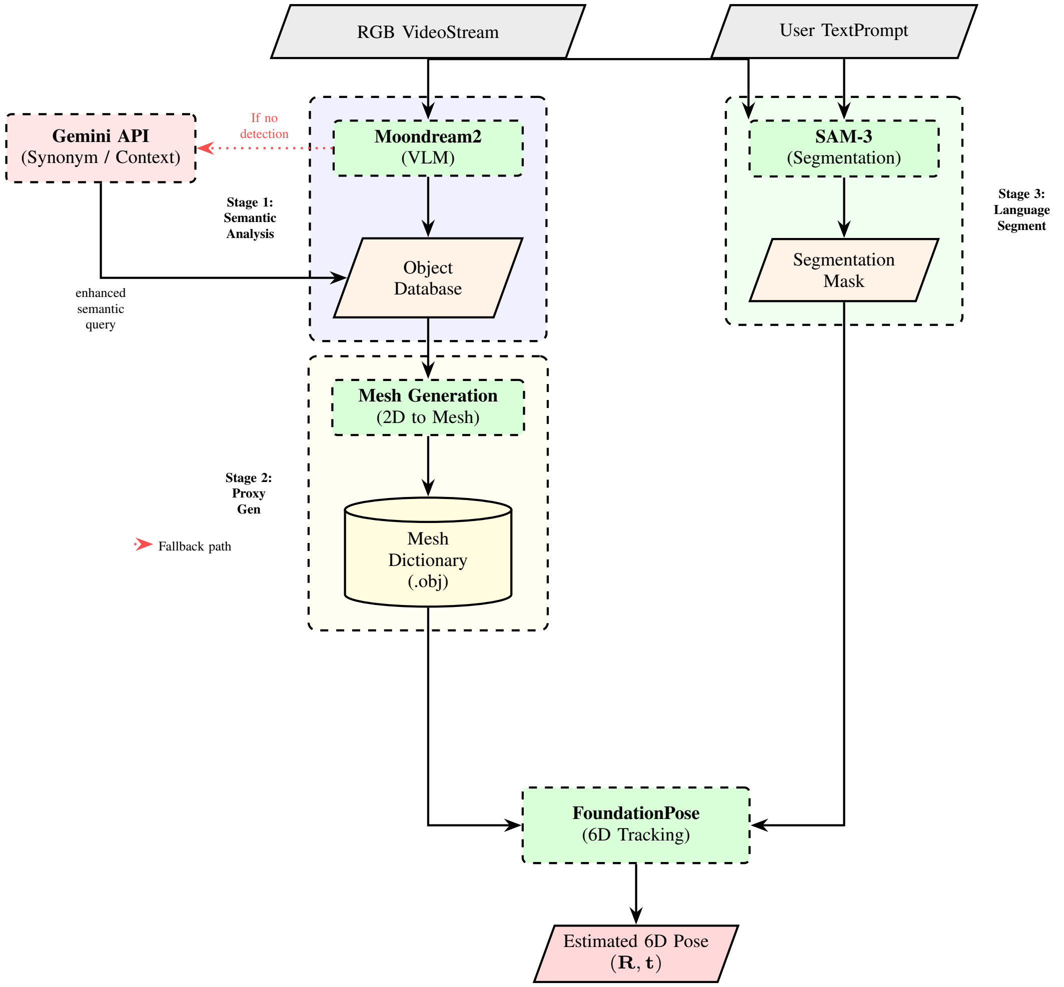


Fig. 2: **Proposed Open-Vocabulary 6D Tracking Architecture.** A vision–language model directly grounds objects into an object database, generates 3D proxy meshes on-the-fly, segments targets using language prompts, and fuses geometry and masks in FoundationPose for robust 6D tracking. When Moondream2 fails to detect the queried object, the Gemini API enriches the query via semantic expansion.

B. Alternative Approach: Moondream2 + Objaverse Retrieval

Learning from the failures of reconstruction, we shifted to a retrieval-based strategy that leverages clean, high-quality template meshes from a large-scale database. This avoids the pitfalls of noisy single-image reconstruction.

We employed Moondream2, a compact yet capable vision-language model, to generate a comprehensive list of objects in the frame. By prompting Moondream2 with "List all distinct independent objects visible in the image," we obtained descriptive object names (e.g., "red mug, blue laptop, black keyboard, potted plant"). Subsequently, those identified object names were used to query Objaverse-XL, a massive repository containing over 10 million 3D objects. Objaverse supports language-guided retrieval, returning the closest matching clean template meshes in seconds via API or similarity search. Unlike reconstruction methods (e.g., TripoSR), this approach provides instance-level approximate meshes that are geometrically accurate and artifact-free, as they originate from professionally designed or scanned models.

This retrieval strategy yielded better meshes for downstream pose estimation. The clean geometry improved render-and-compare scoring robustness, especially under occlusion or low-resolution conditions. However, retrieved meshes are similar but not identical to the observed instance. For instance, a generic "mug" template may differ in handle shape or exact proportions. Furthermore, highly specific, novel objects not well-represented in the database, or errors in Moondream2's object listing result in poor matches.

C. Implementation Details

The implementation adopts a modular design with three separate conda environments to prevent dependency conflicts among the core components. FoundationPose runs on PyTorch 2.0 with CUDA 11.8 and NVDiffRast for differentiable rendering. SAM-3 operates in an isolated environment using PyTorch 2.7, CUDA 12.6, and its specific dependencies; it is invoked as a subprocess to avoid interference with FoundationPose. The lightweight Moondream2 vision-language model is included optionally for scene analysis.

Experiments rely on two benchmark datasets chosen for their challenging conditions. The YCB-Video dataset includes 21 objects across 92 test sequences with heavy real-world occlusions and clutter. The LINEMOD dataset provides 13 texture-less industrial objects, testing geometric reasoning under varying lighting and moderate occlusion.

All evaluations are performed on a single NVIDIA GPU with CUDA support. Inference times are measured directly on this hardware to assess practical real-time feasibility.

D. Evaluation Metrics

We evaluate the proposed framework using standard metrics established in the 6D pose estimation literature, primarily following the protocols of the BOP Challenge. These metrics assess both pose accuracy and robustness across occlusion, symmetric objects, and unseen instances.

The primary metric is the Average Distance of Model Points (ADD), which measures the mean deviation between model vertices transformed by the predicted and ground-truth poses:

$$\text{ADD} = \frac{1}{|M|}\sum_{x\in M}\|(Rx+t)-(R_{gt}x+t_{gt})\| \quad (1)$$

where M is the set of m mesh vertices, R, t are the predicted rotation and translation, and R_{gt}, t_{gt} are the ground truth. A pose is considered correct if ADD $< 0.1d$, where d is the object diameter.

For symmetric objects, we use the ADD-S variant, which accounts for pose ambiguity by taking the distance to the closest model point:

$$\text{ADD-S} = \frac{1}{|M|}\sum_{x_1\in M}\min_{x_2\in M}\|(Rx_1+t)-(R_{gt}x_2+t_{gt})\| \quad (2)$$

We report performance as the Area Under the Curve (AUC) of the accuracy-threshold curve for ADD and ADD-S, with the threshold varied from 0 to $0.1d$. Higher AUC values indicate better overall accuracy.

Additionally, we compute separate rotation and translation errors for detailed analysis:

$$\text{Rot Error} = \arccos\left(\frac{\text{trace}(R^T R_{gt})-1}{2}\right) \quad (3)$$

$$\text{Trans Error} = \|t-t_{gt}\| \quad (4)$$

Rotation error is reported in degrees and translation error in centimeters. When multiple mesh candidates are retrieved from Objaverse-XL, TripoSR and benchmark, we select the final pose using a composite scoring function that combines multiple alignment cues:

$$\text{Score} = w_1\cdot\text{IoU} + w_2\cdot\text{Depth} + w_3\cdot\text{Silhouette} \quad (5)$$

where IoU measures intersection over union between rendered and observed masks, Depth evaluates agreement between rendered and observed depth maps, and Silhouette computes contour matching.

For scale estimation of retrieved meshes (which lack metric dimensions), we employ a depth-based estimator:

$$\text{scale} = \frac{\text{observed\_depth}}{\text{model\_depth}} \quad (6)$$

where observed_depth is the median depth within the masked region, and model_depth is the corresponding depth computed from the normalized mesh dimensions.

TABLE I: Results on YCB-Video Dataset (Scene 48)

Object ID	ADD AUC (%)	ADD-S AUC (%)	Rot (°)	Trans (cm)
1 (Can)	22.4	76.5	111.2	4.67
19 (Clamp)	94.1	100.0	16.9	0.52

V. Results

The proposed framework is evaluated primarily on Scene 48 of the YCB-Video dataset, a sequence that contains multiple interacting objects under realistic occlusion. Quantitative results for two representative objects appear in Table I. Pose estimation result on YCB-Video Object ID 48 and 59, illustrating accurate alignment with coordinate axes overlaid on objects under varying occlusion and lighting conditions.

The Large Clamp achieves strong accuracy, approaching perfect ADD-S AUC and sub-centimeter translation error, which highlights effective handling of texture-rich objects. The Master Chef Can shows lower ADD AUC due to its cylindrical symmetry yet recovers reasonable performance through the ADD-S metric.

Runtime varies between phases. The first frame requires approximately 13–16 seconds for registration, including SAM-3 mask generation and initial pose alignment. Subsequent tracking frames process in about 12–13 seconds each. The overall average reaches 13.3 seconds per frame, with SAM-3 mask generation accounting for roughly 12.5 seconds of the total. Once initialized, FoundationPose itself runs at near real-time speeds.

SAM-3 segmentation proves robust across diverse categories, producing high-confidence masks above 0.5 for well-defined objects, precise boundary delineation, and reliable occlusion handling. Failures remain infrequent and typically arise only with highly similar or ambiguous objects that demand further disambiguation.

Detailed qualitative analysis reveals stable tracking for the Large Clamp across 16 frames, with low mean rotation error of 16.9 degrees and translation error of 5.2 mm. For the Master Chef Can, processing covers 75 frames with variable success in 20 of them; challenges stem from symmetry and specular reflections, though the ADD-S AUC of 76.5 percent confirms adequate symmetric pose recovery.

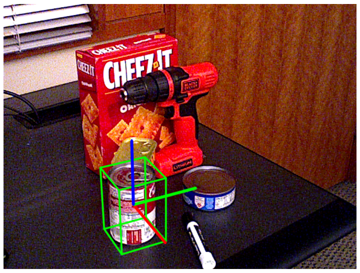


Fig. 3: Object ID 3 from YCB dataset

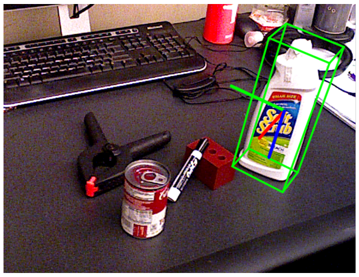


Fig. 4: Object ID 9 from YCB dataset

VI. Discussion and Conclusions

A. Strength and limitation

The proposed framework excels in enabling zero-shot pose estimation for novel objects through simple text prompts, which allows users to issue natural language commands without any retraining. This capability is seamlessly integrated into a unified pipeline that combines segmentation and pose estimation, resulting in effective handling of diverse lighting conditions, viewpoints, and partial occlusions. Additionally, the system flexibly accommodates both ground-truth CAD models and retrieved proxies from Objaverse-XL, making it well-suited for real-world scenarios involving unseen objects.

Despite these advantages, the framework faces challenges related to computational efficiency, as SAM 3 mask generation dominates the processing time at around 12.5 seconds per frame, preventing true real-time performance on standard hardware. Symmetric objects, such as cylindrical items, exhibit lower accuracy due to inherent rotational ambiguity. Furthermore, overall results remain sensitive to the quality and specificity of user prompts, and retrieved meshes from Objaverse-XL—while geometrically clean—do not always match real instances precisely, occasionally leading to reduced pose accuracy.

B. Comparison with State-of-the-Art

Table II compares the proposed method with existing approaches.

TABLE II: Comparison with State-of-the-Art Methods

Method	**Zero-shot Objects**	**Text Prompt**	**ADD-S AUC (%)**
PoseCNN	No	No	75.4
DenseFusion	No	No	82.3
FoundationPose	Yes	No	89.2
Ours (SAM3+FP)	Yes	Yes	76.5–100.0

Our approach uniquely combines zero-shot capability with text-prompted segmentation, trading minor accuracy on some objects for improved flexibility and interactivity compared to FoundationPose and earlier methods.

C. Ablation Studies and Failure Cases

In addition to the successful retrieval-based pipeline, we explored several alternative approaches to enhance occlusion robustness and unseen object handling, though these ultimately proved unsuitable for our real-time, dynamic target switching requirements.

We first investigated DenseFusion [2], a classic RGB-D fusion model that processes color and depth features separately before combining them pixel-wise through a dense fusion unit. This per-pixel fusion allows local predictions to remain viable even when parts of the object are occluded, making DenseFusion particularly robust in heavily occluded scenes. Motivated by this strength, we evaluated DenseFusion on the LineMOD dataset, which contains challenging occlusion scenarios. The model achieved low average pose errors for known objects under occlusion.

However, DenseFusion is fundamentally a model-free, instance-level approach that requires training on the specific objects or classes of interest. For unseen objects, new meshes would need to be incorporated into the training process, which is computationally prohibitive. Moreover, retraining or fine-tuning for every potential object in the scene contradicts our goal of dynamic, prompt-based target switching. Consequently, we abandoned DenseFusion in favor of methods that support zero-shot generalization.

This experience led us to examine SAM-6D [5], a zero-shot pose estimation model built on the Segment Anything Model (SAM). SAM-6D leverages SAM's powerful segmentation and combines it with partial-to-partial point cloud matching against provided CAD models. We tested SAM-6D on the same video sequences used for FoundationPose evaluation, including scenes with a mustard bottle under varying occlusion.

While SAM-6D produced high-quality masks, pose estimation proved substantially slower; it took approximately 40 seconds per frame on our hardware. The primary bottleneck arises from exhaustive partial point cloud matching and hypothesis sampling during refinement. Additionally, overall pose accuracy was lower than FoundationPose baselines in comparable settings. Given these limitations in inference speed, SAM-6D could not meet our real-time tracking requirements.

This insight directly motivated our final design, which retains FoundationPose while augmenting it with advanced language-driven segmentation and on-the-fly mesh retrieval to address occlusion and dynamic flexibility.

VII. Statement of Individual Contribution

Het Patel: He led the overall system integration and pipeline design. He was primarily responsible for combining the components into a cohesive framework, with particular focus on integrating SAM-3 with FoundationPose. He also managed the dynamic target switching mechanism and ensured real-time performance during tracking.

Keisuke Ogawa: He explored alternative pose estimation and mesh generation strategies. He implemented and experimented with the DenseFusion model, as well as the YOLOv8 + SAM + TripoSR pipeline for single-image 3D reconstruction. He also documented the iterative improvements from reconstruction to retrieval approaches, and recorded experiments and failure cases.

Sunny Deshpande: He developed the core semantic mesh extraction pipeline involving Moondream2, Gemini, SAM-3, Objaverse-XL, and FoundationPose. He implemented the semantic scene analysis, object listing, and on-the-fly mesh retrieval components. Additionally, he built a semi-real-time pose estimation system using SAM-6D as an alternative baseline for comparison.

Ansh Bhansali: He focused on environment setup and early mesh generation experiments. He established the development environments for FoundationPose and SAM, and conducted extensive trials on image-to-mesh reconstruction approaches to evaluate their feasibility before the shift to retrieval-based methods.

References

[1] B. Wen, W. Yang, J. Kautz, and S. Birchfield, "Foundationpose: Unified 6d pose estimation and tracking of novel objects," *arXiv preprint arXiv:2312.08344*, 2023.

[2] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese, "Densefusion: 6d object pose estimation by iterative dense fusion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 3338–3347.

[3] D. Tochilkin, D. Pankratz, Z. Liu, Z. Huang, A. Letts, Y. Li, D. Liang, C. Laforte, V. Jampani, and Y.-P. Cao, "Triposr: Fast 3d object reconstruction from a single image," *arXiv preprint arXiv:2403.02151*, 2024.

[4] N. Carion, L. Gustafson, Y.-T. Hu, S. Debnath, R. Hu, D. Suris, C. Ryali, K. V. Alwala, H. Khedr, A. Huang, J. Lei, T. Ma, B. Guo, A. Kalla, M. Marks, J. Greer, M. Wang, P. Sun, R. Rädle, T. Afouras, E. Mavroudi, K. Xu, T.-H. Wu, Y. Zhou, L. Momeni, R. Hazra, S. Ding, S. Vaze, F. Porcher, F. Li, S. Li, A. Kamath, H. K. Cheng, P. Dollár, N. Ravi, K. Saenko, P. Zhang, and C. Feichtenhofer, "Sam 3: Segment anything with concepts," *arXiv preprint arXiv:2511.16719*, 2025.

[5] J. Lin, L. Liu, D. Lu, and K. Jia, "Sam-6d: Segment anything model meets zero-shot 6d object pose estimation," *arXiv preprint arXiv:2311.15707*, 2023.

[6] M. Deitke, R. Liu, M. Wallingford, H. Ngo, O. Michel, A. Kusupati, A. Fan, C. Laforte, V. Voleti, S. Y. Gadre, E. VanderBilt, A. Kembhavi, C. Vondrick, G. Gkioxari, K. Ehsani, L. Schmidt, and A. Farhadi, "Objaverse-xl: A universe of 10m+ 3d objects," *arXiv preprint arXiv:2307.05663*, 2023.

[7] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics yolov8," GitHub repository, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[8] Vikhyat Moondream2, "Moondream2: Small vision language model," 2024. [Online]. Available: https://moondream2.ai/