POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Zero-shot Model-free 6D Object Pose Estimation in RGB-D Videos

Tesi di Laurea Magistrale in
Mathematical Engineering - Ingegneria Matematica

Author: **Simone Paloschi**

Student ID: 251393
Advisor: Prof. Federica Arrigoni
Co-advisors: Angelo Iacoella, Prof. Giacomo Boracchi
Academic Year: 2025-26

Abstract

Knowing the position and orientation of an object in three-dimensional space, collectively referred to as its 6D pose, is a foundational requirement in robotics, augmented reality systems, and many other applications. Classical methods for this task require a 3D model of the object prepared in advance, and even recent model-free approaches still require a dedicated collection of reference images with known poses before tracking can begin. This thesis studies and demonstrates the potential of a fully zero-shot pipeline: given only two points on the object to track in the first frame, the system generates a 3D mesh of the object, recovers its real-world scale, and tracks its pose throughout the RGB-D video, a standard video format that captures both color and depth for each frame, all without any prior knowledge of the object's shape or appearance. The central contribution is a mesh update scheme that progressively refines both the geometry and the texture of the generated mesh using the depth and color observations accumulated from frames during tracking. Our experiments show that zero-shot model-free tracking is viable at a small accuracy cost relative to methods that rely on a prepared 3D model, and that the mesh refinement stage consistently improves both geometry, texture quality, and pose accuracy. These results open practical possibilities for pose tracking in settings where preparing a 3D model or a reference collection is too costly or impractical.

Keywords: 6D object pose estimation, pose tracking, zero-shot, model-free, mesh refinement, RGB-D video

Abstract in lingua italiana

La posizione e l'orientamento di un oggetto nello spazio tridimensionale, noti collettivamente come posa 6D, costituiscono un requisito importante per la robotica, la realtà aumentata e molte altre applicazioni. I metodi classici richiedono un modello 3D dell'oggetto preparato in anticipo, e anche i più recenti approcci *model-free* richiedono una raccolta dedicata di immagini di riferimento con pose note prima che il tracciamento possa avere inizio. Questa tesi studia e dimostra il potenziale di una pipeline completamente *zero-shot*: dati solo due punti dell'oggetto da tracciare nel primo fotogramma, il sistema genera una mesh 3D dell'oggetto, ne recupera la scala reale e ne traccia la posa lungo tutto il video RGB-D, un formato video standard che acquisisce sia il colore che la profondità per ogni fotogramma, senza alcuna conoscenza preliminare della forma o dell'aspetto dell'oggetto. Il contributo centrale è uno schema di aggiornamento della mesh, che raffina progressivamente sia la geometria che la texture della mesh generata utilizzando le osservazioni di profondità e colore accumulate dai fotogrammi durante il tracciamento. I nostri esperimenti mostrano che il tracciamento *zero-shot* privo di modello è praticabile con un costo in termini di precisione ridotto rispetto ai metodi che si affidano a un modello 3D preparato, e che la fase di raffinamento della mesh migliora in modo consistente la geometria, la qualità della texture e la precisione della posa. Questi risultati aprono possibilità concrete per il tracciamento della posa in contesti in cui preparare un modello 3D o una raccolta di riferimento è troppo costoso o impraticabile.

Parole chiave: posa 6D di oggetti, tracciamento della posa, zero-shot, model-free, raffinamento di mesh, video RGB-D

Contents

1 | Introduction

The pose of a rigid object describes its location and orientation in three-dimensional space, encoding where it is and how it is oriented. Knowing the pose of an object allows a system to understand the object's spatial boundaries, reason about its configuration, and interact with it precisely. This makes pose estimation a foundational capability in computer vision, with applications wherever machines must perceive and act on objects.

1.1. 6D Pose Estimation

Recovering a full spatial description of an object requires six parameters: three for translation, specifying where in space the object is located, and three for rotation, specifying how it is oriented. This is known as **6D pose estimation** [12, 58], the problem of recovering this six-degree-of-freedom transformation from image observations. Figure 1.1 illustrates the 6D pose of four objects: the origin of the axes encodes translation, while their orientation encodes rotation. A natural and practically important extension of pose estimation is its application to video sequences, where the goal is to track the object's pose continuously over time rather than estimating it independently for each frame; this sub-field is known as **6D pose tracking** [57].

1.2. Motivation and Applications

Continuous, reliable 6D pose estimates are a prerequisite for many downstream tasks. In robotics [23, 55], accurate pose tracking enables a robot to grasp, manipulate, and avoid colliding with objects, even as they move; in augmented reality [33], it underlies the stable overlay of virtual content onto real-world objects; in autonomous driving, pose estimates of surrounding vehicles and obstacles inform environment perception, trajectory planning, and collision avoidance [1]. In all three domains, not all objects can be assumed to be known in advance, and single-frame estimation alone is insufficient: tracking propagates spatial information across frames, making pose estimates more robust and consistent over time at lower computational cost than independent per-frame inference.

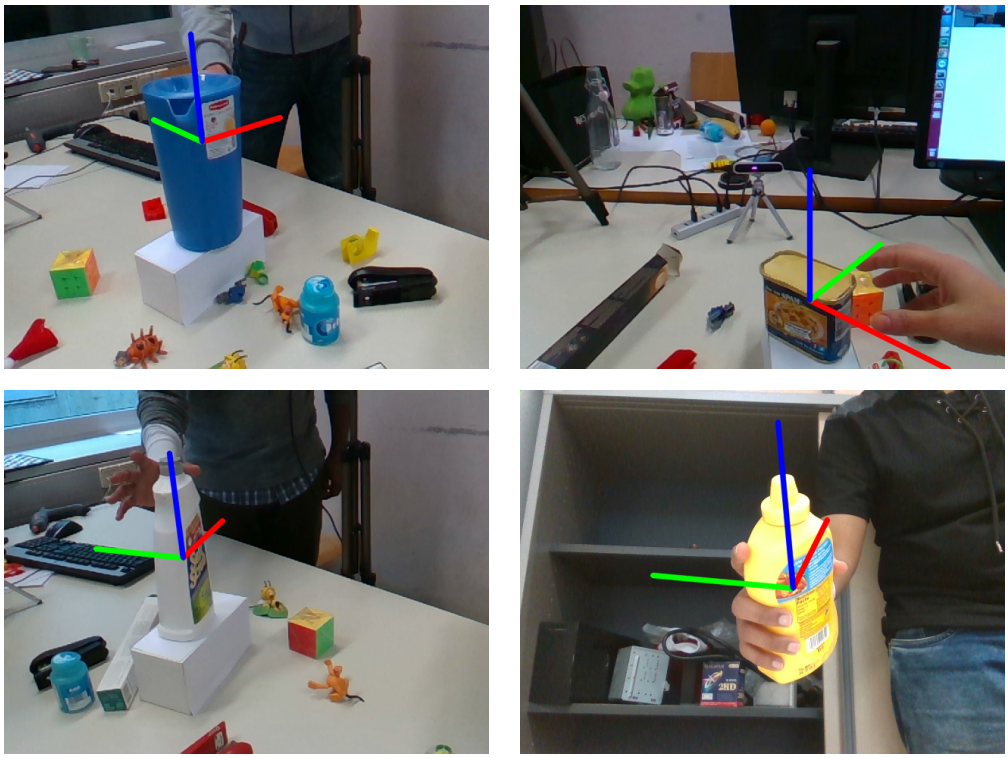

Figure 1.1: First frames from HO3D [9] sequences with the estimated 6D pose overlaid. The axes origin encodes translation; axis directions encode rotation.

1.3. The Classical Approach and Its Limitations

Classical 6D pose estimation methods are **model-based**: they require a textured CAD model of the object, a virtual 3D representation prepared offline before deployment [58]. Pose estimation then proceeds by matching the observed image against synthetic renderings of the model. While effective, this paradigm is costly to deploy: CAD models must be created or obtained for every object of interest, a labor-intensive process that limits these methods to closed-world settings where all objects are known in advance.

1.4. Model-free and Zero-shot Settings

Model-free methods reduce the dependency on a CAD model to a set of reference images with known poses [27, 57]: a dedicated capture session must be conducted outside the test sequence, with each image associated with its camera pose, and the resulting set serves as the object's 3D representation for all downstream tracking. This removes the need for a 3D model, but still requires a costly per-object preparation phase.

The ideal scenario pushes this further: tracking and reconstructing a novel object from

a single RGB-D video, using only a lightweight prompt on the first frame to identify the target object. Notably, the system needs no separate reference capture, no CAD model, and no object-specific training.

An RGB-D video is a sequence of frames, each pairing a standard color image with a per-pixel depth map that records the distance from the camera to the scene.

The segmentation prompt can be as minimal as two clicks on the object or a short text description.

This is the zero-shot model-free setting [15, 22]. It is strictly harder than both model-based and reference-image settings because the system must simultaneously infer the object's shape and track its motion without any geometric prior or multi-view reference observations to anchor either task.

1.5. Tracking and Reconstruction Complementarity

The zero-shot setting, however, exposes a natural opportunity. As tracking proceeds, the system accumulates posed RGB-D observations of the object from multiple viewpoints, progressively covering its surface. Since pose estimators such as FoundationPose [57] operate by comparing the object mesh against the observed image and the portion of the mesh visible in any given frame largely overlaps with what was visible in previous frames, this incremental coverage is sufficient to sustain accurate pose estimation. The key challenge is then to transfer the information from posed RGB-D observations back to the mesh accurately and efficiently, which is precisely what the method developed in this work addresses.

1.6. Two Competing Paradigms

Within the zero-shot setting, two main approaches have emerged. The first, exemplified by BundleSDF [56] and 6DOPE-GS [15], follows an **online reconstruction** strategy: pose and geometry are estimated jointly from scratch as the video unfolds. This approach represents the object faithfully in observed regions but cannot represent unseen parts, since geometry is built exclusively from what has been observed so far. The alternative is **mesh completion**: methods such as HIPPo [28] and UA-Pose [21] leverage image-to-3D generative models [4, 30, 60] to produce a complete mesh from a single image, providing immediate tracking capability from the first frame, and then refine it online as new observations arrive. The complete initial mesh is a key advantage over online reconstruction, but it comes at the cost of hallucinated geometry on unseen parts, scale ambiguity inher-

ent to generative outputs, and potential discrepancies between the generated appearance and the real object texture.

1.7. This Thesis

This thesis proposes a fully zero-shot pipeline that requires only a segmentation prompt [17], such as two points on the object, in the first frame. The pipeline chains five stages: object segmentation and mask propagation [5, 17], single-view mesh generation with SAM-3D [4], a two-round FoundationPose-based scale recovery procedure, FoundationPose [57] tracking, and a novel confidence-based geometric vertex update for mesh refinement.

SAM-3D is applied directly on the first frame, removing the need for a separate reference capture; robustness to partial occlusion is therefore essential, as the first frame may not show the object from an ideal viewpoint, and SAM-3D is shown in this work to handle this reliably. Figure 1.2 shows the SAM-3D mesh rendered at the estimated pose on the first frame of four sequences, alongside the original RGB image and their overlay.

Scale recovery is designed as a search over a single uniform scaling factor rather than a per-axis procedure, a simplification compared to other methods [18] justified by the accurate object proportions that SAM-3D produces. The mesh completion stage, our main contribution, exploits the posed RGB-D observations accumulated during tracking to refine both geometry and appearance. Each mesh vertex accumulates evidence across frames in the form of observed position and color estimates, derived from depth unprojection under the current pose and segmentation mask. Since both estimates are imperfect and depth sensors introduce additional noise, each vertex is also assigned a confidence scalar that weighs observations over time: the mesh used for pose estimation is updated only when sufficient consistent evidence supports a vertex, successfully filtering unreliable estimates, while improving fidelity where real observations are available and preserving the generative prior on unobserved parts.

The pipeline is evaluated on HO3D [9], reaching an ADD-S AUC of 82.0%, where ADD-S is a standard pose accuracy metric, within 2.5 points of the model-based upper bound (84.5%) that uses the ground-truth CAD model, while substantially outperforming model-free baselines Any6D [18] (70.6%) and UA-Pose [21] (68.1%). Texture quality, measured by PSNR, surpasses the model-based baseline on 12 of 13 sequences because vertex colors are learned directly from the video under scene-accurate lighting, rather than from a fixed CAD texture. These results show that zero-shot model-free tracking is viable at minimal accuracy cost relative to model-based approaches, and that even a simple geometric completion stage consistently improves both geometry and pose metrics.

The main contributions of this thesis are:

- A fully zero-shot pipeline for 6D pose tracking and mesh completion from a single RGB-D video, requiring only a segmentation prompt.
- An empirical study of the current capabilities and limitations of zero-shot model-free 6D pose tracking, demonstrating how close this setting can get to model-based performance.
- A scale recovery procedure adapted to SAM-3D meshes, based on two consecutive rounds of FoundationPose hypothesis scoring.
- A confidence-based mesh completion scheme that refines geometry and texture from posed RGB-D observations while preserving the generative prior on unseen parts.

1.8. Thesis Structure

Chapter 2 formalizes the 6D pose tracking and mesh completion problem, defining inputs, outputs, and assumptions. Chapter 3 reviews the relevant literature on object segmentation, pose estimation, and 3D reconstruction. Chapter 4 presents the proposed pipeline in detail, analyses the challenges of each paradigm, and provides a comparative analysis against competing methods. Chapter 5 describes the experimental setup and presents quantitative and qualitative results on HO3D. Chapter 6 summarizes the findings and discusses directions for future work.

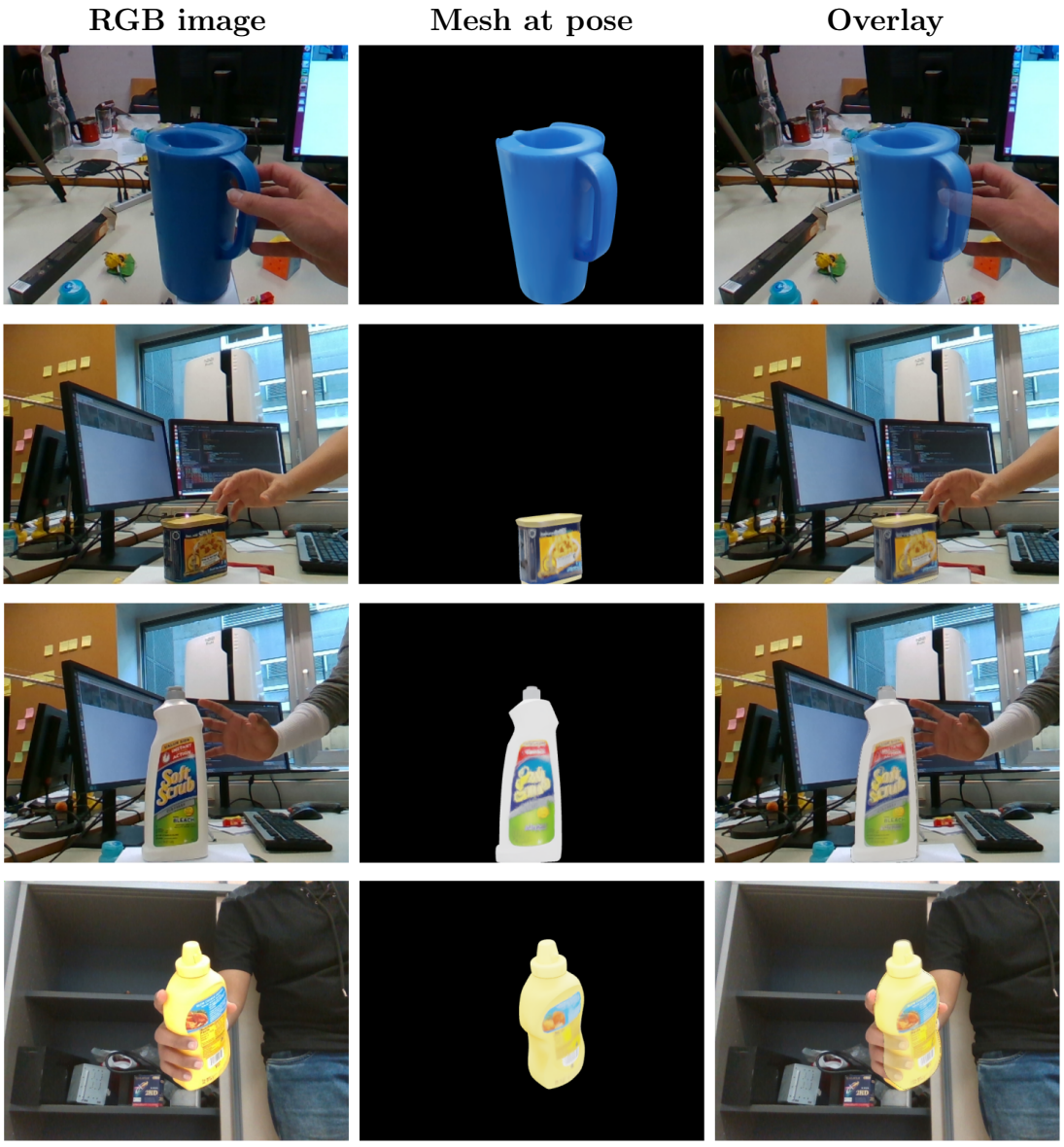

Figure 1.2: First frames from HO3D [9]. Each row shows the original RGB image, the SAM-3D mesh rendered at the estimated pose, and the overlay (60% mesh, 40% RGB).

2 | Problem Formulation

This chapter defines the problem of **6D object pose tracking and mesh completion** from RGB-D video sequences. The objective is to estimate the full 6-degree-of-freedom pose of unknown objects throughout a video sequence while simultaneously reconstructing a complete 3D mesh representation of the tracked object.

Given an RGB-D video sequence containing an object of interest, the goal is to produce two complementary outputs: (1) a temporal sequence of 6D poses describing the object's position and orientation in each frame and (2) a unified 3D mesh model capturing the complete surface geometry of the object. This dual estimation problem is particularly challenging when no prior knowledge of the object's geometry or appearance is available.

2.1. Input Data

The input to the problem is an RGB-D video sequence, the camera calibration parameters and an initial selection of the object we want to track and reconstruct.

2.1.1. RGB-D Video

The video is a sequence of RGB-D frames (Figure 2.1). Each frame has width W and height H, forming a grid of pixels. For every pixel (u, v) we have:

- three color values $(R, G, B) \in [0, 255]$;
- one depth value $D(u, v) \in \mathbb{R}$, which corresponds to the distance from the camera.

We can think of the depth channel as a depth map that links a pixel to its depth. Considering T as the total number of frames, the video will have dimensions:

$$T \times W \times H \times 4$$

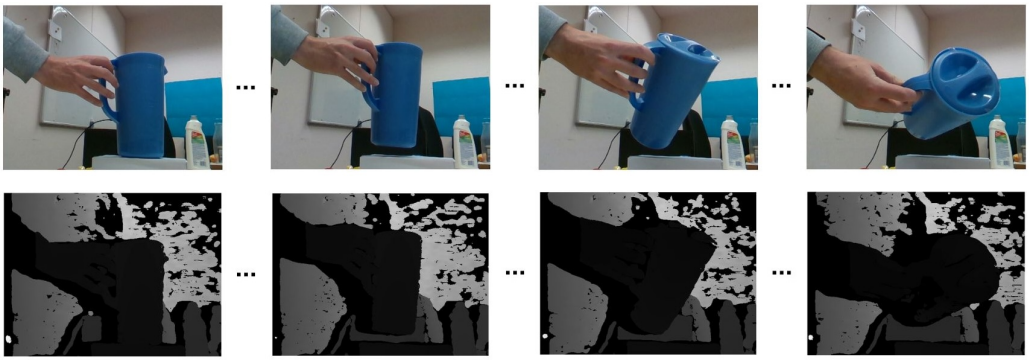

Figure 2.1: Example RGB-D frames from the HO3D dataset [9]. In the depth map, pixel brightness encodes distance: darker pixels are closer. Black pixels in the background indicate missing depth measurements due to sensor errors.

2.1.2. Camera Model

To connect 3D points to image pixels, we use the pinhole camera model. The camera and its intrinsic parameters are described by the calibration matrix $\mathbf{K}$:

$$\mathbf{K} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

where:

- f_x, f_y are the focal lengths in pixels;
- (c_x, c_y) is the principal point (the optical center in the image);
- s is the skew, usually zero for most modern cameras.

Given a 3D point in camera coordinates $P_{\text{cam}} = [X, Y, Z]^T$, its projection onto the 2D image plane at pixel $p = [u, v]^T$ is:

$$Z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Using depth information $Z(u, v) = D(u, v)$, we can invert this relation to go from a pixel

(u, v) back to the 3D point:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = Z(u, v)\, \mathbf{K}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

This allows us to recover a 3D point cloud from each RGB-D frame.

2.1.3. Segmentation Prompt

To specify the object of interest, we assume an initial selection in the first frame of the video (Figure 2.2). This selection can be provided, for example, by:

- two points clicked on the object
- a small region around the object
- a description of the object

This prompt identifies which part of the scene we want to track and reconstruct over time. All subsequent processing focuses on the same selected object.

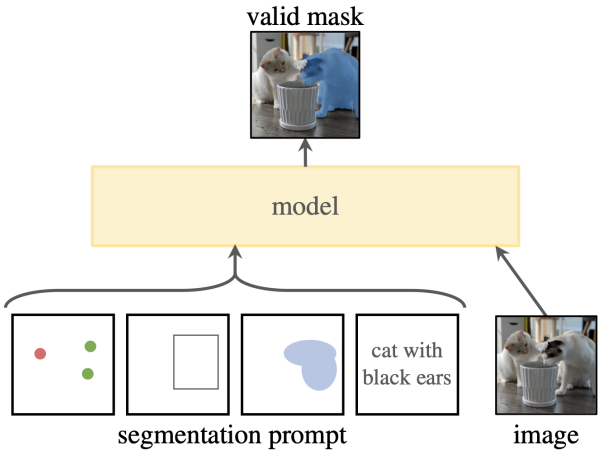

Figure 2.2: Example input prompts for object selection, from [17].

2.2. Output

The goal of this problem is to produce two synchronized outputs: a sequence of 6D poses tracking the object's motion over time, see Figure 2.3, and a complete 3D mesh model of the object's surface.

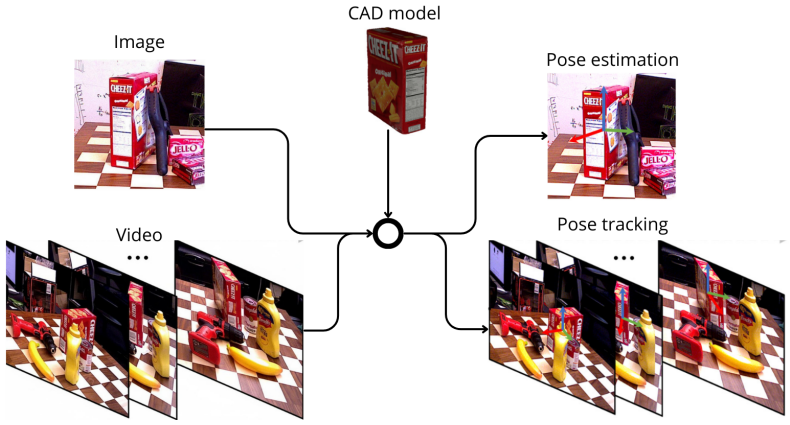

Figure 2.3: Visual representation of pose estimation on images and videos, from [57].

2.2.1. 6D Pose Sequence

For each frame i in the video sequence, we estimate a pose $\{\mathbf{R}_i, \mathbf{t}_i\}$ that describes the object's position and orientation relative to the camera (see Figure 2.4). A pose has 6 degrees of freedom: 3 for translation and 3 for rotation.

Translation

The translation vector $\mathbf{t}$ specifies where the object is located:

$$\mathbf{t} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \in \mathbb{R}^3,$$

where (x, y, z) are the coordinates of the object's origin in the camera frame.

Rotation

The orientation is captured by a rotation matrix $\mathbf{R} \in \mathrm{SO}(3)$, which is a 3×3 orthonormal matrix. One way to represent $\mathbf{R}$ is through three rotation angles $(\theta_x, \theta_y, \theta_z)$, known as Euler angles. The full rotation is obtained by composing rotations around each axis:

$$\mathbf{R} = \mathbf{R}_z(\theta_z)\, \mathbf{R}_y(\theta_y)\, \mathbf{R}_x(\theta_x),$$

where the elementary rotation matrices are:

$$\mathbf{R}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}, \quad \mathbf{R}_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}, \quad \mathbf{R}_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

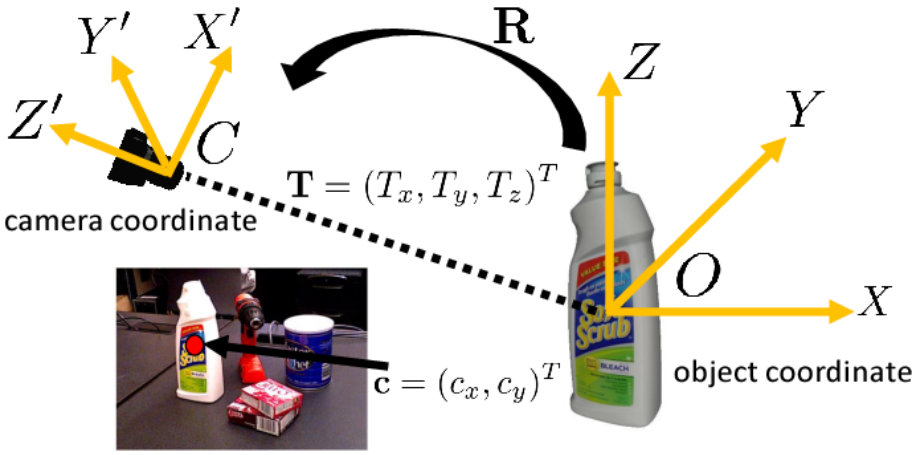

Figure 2.4: 6D object pose representation in the camera coordinate system, from [58]. The rotation $\mathbf{R}$ and translation $\mathbf{T} = (T_x, T_y, T_z)^\top$ map the object's local coordinate frame $(O; X, Y, Z)$ into the camera frame $(C; X', Y', Z')$. $\mathbf{c} = (c_x, c_y)^\top$ is the camera origin C projected into image-space.

Homogeneous Representation

It is often convenient to combine rotation and translation into a single 4×4 transformation matrix using homogeneous coordinates:

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & x \\ r_{21} & r_{22} & r_{23} & y \\ r_{31} & r_{32} & r_{33} & z \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

This representation allows us to transform a 3D point $\mathbf{p}_{\text{obj}} = [x_o, y_o, z_o]^T$ from the object's local coordinate system into the camera frame with a single matrix multiplication:

$$\begin{bmatrix} \mathbf{p}_{\text{cam}} \\ 1 \end{bmatrix} = \mathbf{T} \begin{bmatrix} \mathbf{p}_{\text{obj}} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}\,\mathbf{p}_{\text{obj}} + \mathbf{t} \\ 1 \end{bmatrix}.$$

2.2.2. Reconstructed 3D Mesh

The second output is a single, complete 3D mesh model $\mathcal{M}$ of the object surface that represents the full geometry, including regions not visible in any single frame.

A 3D mesh is a geometric representation of an object's surface that provides a compact and structured way to describe complex shapes. A mesh $\mathcal{M}$ consists of the following components (see Figure 2.5 for a visual example):

- **Vertices**: A set of 3D points $\mathbf{V} = \{\mathbf{v}_i \in \mathbb{R}^3 \mid i = 1, \ldots, N_v\}$;
- **Faces**: A set of triangles $\mathcal{F} = \{f_j \mid j = 1, \ldots, N_f\}$, where each face $f_j = (i_1, i_2, i_3)$ references three vertices that form a triangle;
- **Appearance**: Color or texture information, either stored per vertex or per face, that defines the visual appearance of the surface.

Meshes are typically defined in the object's local coordinate system (canonical frame). To place the mesh in camera space at a given frame, we apply the pose transformation $\mathbf{T}$ to all vertices:

$$\mathbf{v}_i^{\text{cam}} = \mathbf{R}\,\mathbf{v}_i + \mathbf{t}.$$

Figure 2.5: Three representations of the Stanford Bunny [48]. *Left*: a dense point cloud $\{\mathbf{v}_i \in \mathbb{R}^3\}$ capturing the raw surface geometry. *Center*: a coarse triangle mesh $\mathcal{M}$ with a reduced number of faces N_f, highlighting the underlying polygonal structure. *Right*: the same coarse mesh with a diffuse texture applied per face, illustrating how appearance information is encoded on the surface.

2.3. Standard Approach

The standard pipeline for 6D object pose tracking and mesh completion, shown schematically in Figure 2.6, consists of the following stages:

- **Object segmentation**: Detect and segment the target object to obtain a pixel-level mask isolating it from background clutter

- **Initial mesh generation**: Generate a complete or partial 3D mesh representation from the first frame or reference images
- **Initial pose estimation**: Estimate the 6D pose for the first frame. The mesh must be sufficiently accurate, as errors propagate through subsequent observations and can degrade the reconstruction
- **Pose tracking**: Track the object's 6D pose across subsequent frames by refining pose estimates using the current mesh
- **Mesh refinement**: Iteratively improve the mesh geometry using accumulated posed RGB-D observations from tracking

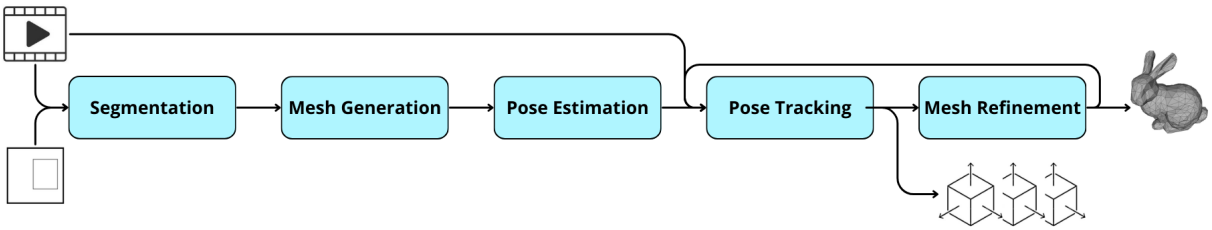

Figure 2.6: Mesh completion pipeline. RGB-D video sequence and a segmentation prompt [17] as input. A refined mesh and 6D poses for all frames as output.

2.4. Assumptions and Challenges

The problem formulation relies on the following constraints and assumptions:

- **Rigid body assumption**: The tracked object is assumed to be rigid, with no deformation over time
- **Temporal continuity**: The object motion between consecutive frames is assumed to be smooth and continuous
- **Single object tracking**: The system focuses on tracking a single primary object.

This problem presents several fundamental challenges:

- **Partial observability**: At any given frame, only a portion of the object surface may be visible due to self-occlusion and viewpoint limitations
- **Unknown object geometry**: No prior 3D model or CAD representation is available and the specific object has not been seen during any training phase
- **Robustness**: The system must handle occlusions, lighting variations and challenging viewpoints

3 | Related Work

The problem of simultaneous 6D object pose estimation and 3D mesh reconstruction requires detecting objects in video sequences, estimating their 6-degrees-of-freedom pose (translation + rotation) and reconstructing their 3D geometry.

The problem we address, presented in depth in Chapter 2, represents the convergence of multiple research areas in computer vision and robotics. It is rooted in two classical paradigms: pose estimation given a known 3D model and 3D reconstruction given known camera-object relative pose. Classical methods operated under a critical assumption: either the complete object geometry was available *a priori* (enabling pose estimation), or the camera-object relative trajectory was known (enabling reconstruction).

In the following sections, we review these research areas, examining their strengths, limitations and convergence towards methods that generalize to novel objects.

3.1. Object Segmentation

Before pose estimation and tracking can begin, the target object must be identified. The goal is to obtain the object mask, a boolean array indicating for each pixel whether it belongs to the object (Figure 3.1).

Figure 3.1: Example object masks from the HO3D dataset [9].

Segmentation errors directly propagate to pose estimation by contaminating depth observations with background geometry or excluding valid object regions. Some methods [18] avoid this problem by using ground-truth masks for evaluation.

3.1.1. Visual Recognition Tasks

Object segmentation builds upon a hierarchy of visual recognition tasks in computer vision. Object recognition encompasses three related tasks of increasing complexity. **Localization** [43] identifies a single object's position within an image using a bounding box. **Detection** [8] extends this to multiple objects, combining classification and localization to identify all instances with their corresponding class labels and bounding boxes (Figure 3.2). **Segmentation** [29] provides pixel-level precision by assigning each pixel to an object class, producing exact boundaries rather than rectangular approximations. For pose estimation applications, segmentation is essential as it provides the precise object masks required to isolate target geometry from background clutter.

Figure 3.2: Object detection with bounding boxes and class labels using YOLO [38].

3.1.2. From Traditional Methods to Zero-Shot Segmentation

Traditional object detection methods extracted hand-crafted features followed by classification or localization; however, they required object-specific training and could not generalize to unseen categories.

Deep learning fundamentally transformed the field. R-CNN [8] demonstrated that convolutional neural networks could dramatically improve detection accuracy. YOLO [38] introduced real-time detection by framing object detection as a single regression problem, enabling end-to-end optimization and significantly faster inference. For segmentation, Fully Convolutional Networks [29] revolutionized the field by enabling end-to-end pixel-wise prediction through fully convolutional architectures with upsampling layers. Mask R-CNN [10] extended this by combining object detection with instance segmentation, simultaneously predicting bounding boxes, class labels and pixel-precise masks for each object. These approaches still required fine-tuning on specific object categories and could not segment novel objects without retraining.

The introduction of foundation models, large AI systems trained on vast datasets that adapt to diverse tasks, brought true zero-shot capabilities to segmentation. Unlike earlier approaches, these models can segment entirely novel objects without needing example images or task-specific training. The Segment Anything Model (SAM) [17] exemplifies this shift. Trained on over 1 billion masks across 11 million images, SAM works through promptable segmentation: users simply provide cues like foreground/background points, bounding boxes, rough masks, or text descriptions to indicate what should be segmented. This flexibility is especially valuable because objects vary widely and collecting training data for every item would be impractical.

3.1.3. Video Object Segmentation

Video Object Segmentation (VOS) extends segmentation to temporal data by propagating a given first-frame mask across frames to maintain temporal consistency. Unlike frame-by-frame segmentation, VOS leverages temporal coherence to handle challenges such as occlusions, appearance variations, and motion blur. XMem [5] presents a long-term VOS architecture inspired by the Atkinson-Shiffrin memory model, incorporating three independent yet deeply-connected feature memory stores: a sensory memory updated every frame, a working memory updated every r-th frame, and a compact long-term memory. Given the mask of the first frame, XMem reads from all three stores to produce per-frame segmentation masks throughout the video. Track Anything [61] integrates SAM [17] with XMem to enable interactive zero-shot video segmentation: users provide point or bounding box prompts on the first frame, SAM generates an initial mask, and XMem propagates it across the sequence without category-specific training.

3.2. 6D Object Pose Estimation

6D pose estimation determines an object's position and orientation in 3D space from image observations. The field has evolved from traditional methods requiring known 3D models and hand-crafted features to learning-based approaches that generalize to novel objects with minimal or no prior information. This progression toward generalization is essential for practical applications where objects cannot be pre-scanned or annotated.

3.2.1. Traditional Methods

Traditional 6D pose estimation methods relied on explicit feature extraction and matching. These approaches followed a common pipeline: extract discriminative features from both

the target object model and the input data, establish correspondences between them, and solve for the rigid transformation using geometric constraints. Feature representations varied from gradient and normal information in template matching [12], to geometric relationships in Point Pair Features [6], to learned coordinates via Random Forests [3].

Methods diverged based on input modality. RGB methods solved the correspondence problem through Perspective-n-Point (PnP) algorithms [19], which estimate pose by minimizing reprojection error between point correspondences, within RANSAC frameworks [7] to filter outliers. RGB-D methods performed direct 3D-to-3D alignment using techniques like Iterative Closest Point (ICP) [2], which iteratively establishes nearest-neighbor correspondences between point clouds and minimizes alignment error.

The fundamental limitation across all traditional methods was their inability to generalize. They required either manual template creation, or accurate 3D models before estimation could begin. These constraints, combined with high computational complexity, motivated the paradigm shift toward learning-based approaches.

3.2.2. Learning-Based Methods

Deep learning fundamentally transformed pose estimation by replacing hand-crafted features with learned representations, progressively relaxing the constraints that limited traditional methods. This evolution spans three phases: instance-level methods trained on specific objects, novel object approaches requiring reference data, and zero-shot methods that generalize without prior examples.

Early learning-based approaches operated in the *instance-level* regime, training models on specific objects with known CAD models and extensive pose annotations. PoseCNN [58] pioneered end-to-end learning by predicting 2D object centers and 3D translations, then refining estimates through geometric reasoning. DenseFusion [49] advanced this paradigm by fusing pixel-wise RGB and depth features within a dense correspondence framework.

Methods for novel objects diverged into *model-based* approaches using CAD models and *model-free* approaches relying only on reference images. OnePose [46] and its extension OnePose++ [11] use structure-from-motion techniques to reconstruct 3D point clouds from a simple RGB video scan of the object and employ pre-trained 2D-3D matching networks to estimate pose. Gen6D [27] addresses novel objects through a two-stage approach: generating a 3D model from multiple posed reference images using learned priors, then estimating pose via dense correspondence prediction.

Recent research has reduced reference requirements, achieving *single-image* operation.

UNOPose [26] constructs an SE(3)-invariant reference frame to estimate pose from a single unposed RGB-D reference image. Similarly, methods leveraging image-to-3D generation, such as Any6D [18] and UA-Pose [21] (in single-image mode), generate a 3D model from an unposed reference image in a canonical frame, then estimate the relative pose to query images. The current frontier targets *zero-shot* generalization, estimating pose for completely unseen objects without CAD models or reference images [15, 22, 28].

FoundationPose

FoundationPose [57] employs a render-and-compare strategy where pose hypotheses are evaluated by rendering synthetic views of the 3D model and comparing them against observed images. The architecture consists of two core networks operating on multiple pose hypotheses initialized with translation set to the median depth of masked points and rotations uniformly sampled from an icosphere.

For each hypothesis, the input image is cropped according to the pose, and the model is rendered from that viewpoint. A refiner network based on CNNs and transformer encoders processes pairs of observed and rendered images to predict translation and rotation updates, which can be applied iteratively for increased precision. A hierarchical pose ranking network then evaluates all refined hypotheses simultaneously, comparing rendered images to observations through feature extraction and multi-head self-attention to produce a ranked list of poses.

FoundationPose provides both pose estimation and tracking modes. The estimation mode employs the full multi-hypothesis initialization and refinement pipeline, while tracking mode uses only the previous frame's pose as the initial hypothesis, enabling efficient real-time operation.

FoundationPose represents the current state-of-the-art open-source method for pose estimation and tracking, and recent mesh completion methods rely on it for these tasks.

3.3. 3D Reconstruction

3D reconstruction recovers the complete surface geometry of objects or scenes from visual observations, providing the geometric foundation necessary for pose estimation.

3.3.1. Traditional Methods

Classical approaches evolved from offline multi-view methods to real-time systems, establishing representations and fusion strategies that remain foundational to modern techniques.

Offline multi-view reconstruction methods like Structure-from-Motion (SfM) [41] and Multi-View Stereo (MVS) [42] estimate camera poses and dense geometry from many images captured from diverse viewpoints. SfM recovers sparse 3D structure through feature matching and bundle adjustment, while MVS densifies this reconstruction by computing per-pixel depth through photometric consistency across views. These methods produce high-quality results but require static scenes and extensive computation.

Simultaneous Localization and Mapping (SLAM) [35] enabled online reconstruction during camera motion by jointly estimating sensor poses and environment geometry. SLAM systems process observations incrementally, making them suitable for real-time applications. However, traditional SLAM remained scene-centric rather than object-focused, reconstructing entire environments without distinguishing individual objects.

A foundational representation for volumetric fusion is the Truncated Signed Distance Field (TSDF). TSDFs discretize space into voxel grids where each voxel stores the truncated signed distance to the nearest surface. This representation enables incremental fusion of noisy depth measurements through weighted averaging and supports efficient mesh extraction via Marching Cubes. KinectFusion [36] demonstrated efficient dense reconstruction from RGB-D cameras, establishing TSDF fusion as a standard approach for online 3D reconstruction.

For mesh generation from point clouds, Poisson surface reconstruction [16] solves a spatial Poisson problem to produce smooth, watertight surfaces from oriented points. Unlike explicit surface representations, Poisson reconstruction formulates surface generation as an implicit function fitting problem, generating complete meshes even from incomplete observations. This technique remains widely used for post-processing and refinement in modern reconstruction pipelines.

3.3.2. Neural Implicit Representations

Recent advancements have shifted from discrete voxel grids to continuous neural implicit representations, where a neural network approximates the scene geometry.

Density-Based Fields (NeRF). Neural Radiance Fields (NeRF) [34] represent a scene as a continuous volumetric field. A multilayer perceptron (MLP) F_Θ maps a 3D coordinate $\mathbf{x}$ and viewing direction $\mathbf{d}$ to a volume density σ and emitted color $\mathbf{c}$:

$$F_\Theta(\mathbf{x}, \mathbf{d}) = (\mathbf{c}, \sigma) \tag{3.1}$$

Image synthesis is performed via differentiable volume rendering, where the color of a pixel is computed by integrating radiance along a ray $\mathbf{r}(t)$. While NeRF excels at novel view synthesis, the extracted meshes often result noisy without a solid surface. Despite this limitation, NeRF has been adapted for pose estimation tasks by training scene-specific radiance fields and generating synthetic training data for pose regression networks, though this requires known poses during NeRF training.

Surface-Based Fields (Neural SDFs). To address the geometric limitations of NeRF, methods such as NeuS [51] and VolSDF [62] propose representing the geometry as a Signed Distance Field (SDF). The surface $\mathcal{S}$ is implicitly defined as the zero level-set of the function f:

$$\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^3 \mid f(\mathbf{x}) = 0\} \tag{3.2}$$

This formulation imposes solid geometry, allowing for the extraction of meshes via Marching Cubes. FoundationPose [57] applies this approach to obtain a mesh used as a drop-in replacement for CAD models, using 4-16 reference images with ground-truth pose annotations to learn the object 3D representation. The object's geometry is represented using a signed distance function $\Omega : \mathbf{x} \to s$ that maps 3D coordinates to signed distance values and an appearance function $\Phi : (f_\Omega(\mathbf{x}), \mathbf{n}, \mathbf{d}) \to \mathbf{c}$ that predicts color using the intermediate feature vectors $f_{\Omega(x)}$ from the geometry network, surface normals, and view directions.

3.3.3. Image-to-3D methods

Recent learning-based approaches generate complete 3D geometry from single RGB images by leveraging generative models. Zero-1-to-3 [24] pioneered this capability through fine-tuning Stable Diffusion [39] for zero-shot novel view synthesis. Zero123++ [44] extended these multi-view capabilities, enabling InstantMesh [60] to directly produce 3D representations through feed-forward networks trained on large-scale datasets.

Two fundamental limitations prevent direct application to tracking. First, generated meshes lack physical scale grounding. Pose estimation degrades severely under scale er-

rors, where even coarse approximations produce poor tracking performance. Second, these approaches *hallucinate* unseen regions from learned priors rather than actual observations. When tracking reveals these hallucinated regions, geometric inconsistencies emerge and compromise pose estimation accuracy.

Methods like Any6D [18] address the scale problem through coarse-to-fine scaling. They obtain a coarse approximation through oriented bounding box alignment between the generated mesh and observed depth, then refine scale through multiple FoundationPose estimations with different scale hypotheses. SAM-3D [4] introduces a foundation model for 3D that predicts object shape, texture, and pose from a single image. By developing a well-annotated dataset for 3D generation, it achieves significant improvements over prior works and performs well in occluded and cluttered scenes.

3.4. Online Mesh Reconstruction

Reconstruction methods build 3D object representations incrementally from RGB-D observations without relying on pre-existing CAD models or foundation model priors. These approaches formulate pose tracking as estimating the relative transformation from the initial frame to each subsequent frame $T_{0\to\tau} \in SE(3)$ with $\tau \in \{0, 1, ..., t\}$. Given an RGB-D sequence $\{I_\tau\}$ and an initial object mask M_0 on the first frame, they jointly optimize camera-object poses while reconstructing geometry from scratch.

These methods face a fundamental interdependency challenge: accurate pose estimation requires high-quality geometry, yet geometric refinement requires precise pose estimates. Moreover, they share a fundamental limitation: unobserved object regions remain incomplete, producing partial meshes with missing geometry.

3.4.1. SLAM-like Methods

Early object-level SLAM approaches [40, 59] extended neural SLAM techniques to dynamic objects but suffered from cumulative pose estimation errors due to their strategy of directly fusing RGB-D data using pose estimations. BundleTrack [53] addressed this limitation by maintaining representative historical observations as keyframes in a memory pool, enabling 3D representation learning through pose graph optimization that corrects erroneous frame estimates.

BundleSDF [56] advanced this paradigm by integrating neural implicit reconstruction with 6D object tracking. The method co-designs two simultaneous threads: an online pose graph optimization process and a Neural Object Field representation modeling both ge-

ometry via signed distance fields and visual texture. A dynamic memory pool of keyframes bridges these threads, automatically selecting the most informative historical observations to maintain multi-view consistency while preventing catastrophic forgetting.

3.4.2. 6DOPE-GS

6DOPE-GS [15] presents a zero-shot model-free approach for joint 6D pose estimation and reconstruction that replaces neural implicit representations with 2D Gaussian Splatting for computational efficiency [14]. The pipeline begins with object segmentation using SAM2 [37] and feature matching via LoFTR [45], followed by coarse pose initialization through RANSAC-based bundle adjustment [7]. The core "Gaussian Object Field" incrementally constructs oriented planar Gaussian disks jointly optimized with keyframe poses. A dynamic keyframe selection mechanism clusters views around icosahedron anchor points to ensure spatial coverage, while opacity-based pruning removes Gaussians in the bottom 5th percentile opacity to stabilize training.

After the Gaussian Object Field converges, all keyframe poses undergo refinement by temporarily freezing the Gaussians and optimizing poses using the reconstruction of the RGB, depth, and normals. The refined keyframes subsequently guide online pose graph optimization that selects overlapping frames from a memory pool based on view frustum visibility, computing dense pixel-wise reprojection errors for real-time tracking.

3.5. Online Mesh Completion

Mesh completion methods leverage image-to-3D foundation models to initialize a complete and potentially inaccurate 3D mesh from minimal input, then progressively refine it through posed RGB-D observations. Unlike reconstruction approaches that build geometry incrementally from scratch, these methods start with full object coverage generated by diffusion models, combining their strong generalization capabilities with geometric consistency refinement for robust pose tracking.

The typical pipeline (Figure 3.3) consists of three stages: mesh generation, scale recovery, and iterative refinement. First, a complete mesh is generated from either a single reference image or the first observed frame. Second, since diffusion models produce scale-agnostic output, the mesh is rescaled to match the actual object dimensions using depth information. Third, the scaled mesh enables accurate pose estimation, after which the system iteratively refines the mesh geometry using accumulated posed RGB-D observations from pose tracking. For pose estimation and tracking, these methods typically leverage

FoundationPose's networks [57].

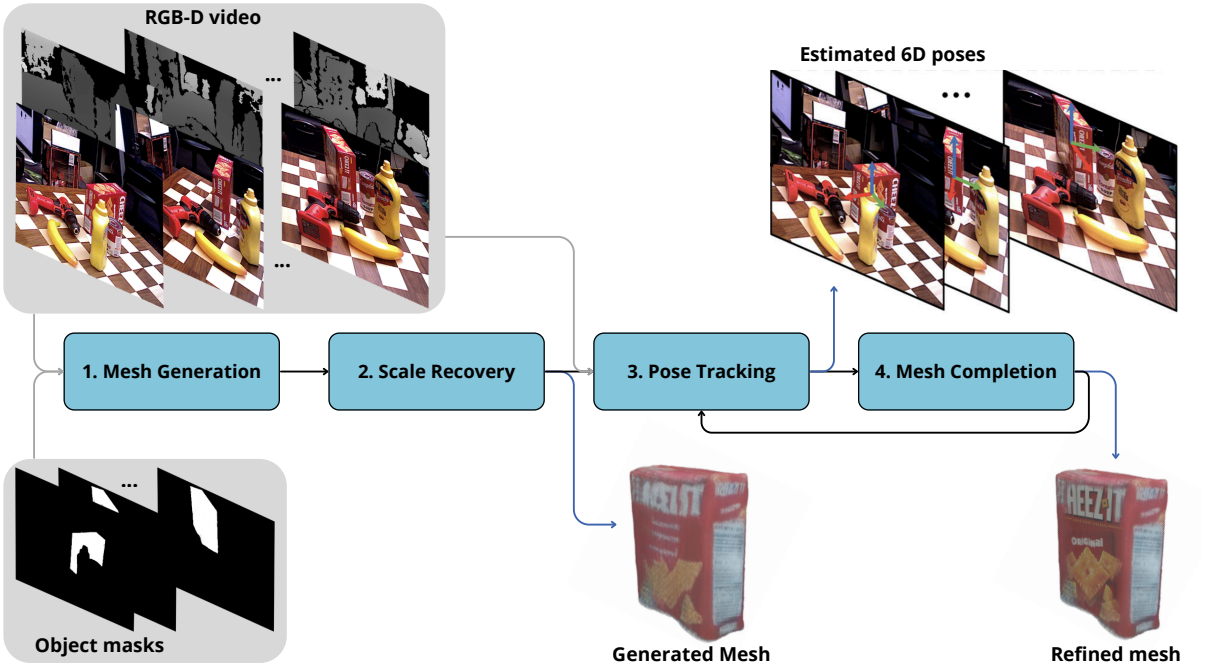

Figure 3.3: Pipeline overview of online mesh completion methods. Starting from an RGB-D video and object masks, the system (1) generates an initial complete mesh via an image-to-3D foundation model, (2) recovers real-world scale using depth data, (3) estimates 6D object poses using FoundationPose, and (4) iteratively refines the mesh geometry from accumulated posed RGB-D observations, feeding the improved mesh back into the tracking stage. Object masks are usually obtained through an off-the-shelf segmentation method. Images from the YCB-Video dataset [58].

3.5.1. UA-Pose

UA-Pose [21] introduces an uncertainty-aware framework for 6D object pose estimation and online mesh completion from partial references. The method constructs a hybrid object representation that integrates texture, geometry, and uncertainty by training a neural Signed Distance Field (SDF) from a limited set of reference images. Here we focus on the single-reference scenario for pose estimation and tracking.

Mesh generation. Given a single unposed RGB image, the method employs InstantMesh's image-to-3D technique [60] to generate an initial object 3D model. The viewpoint associated with the initial image is labeled as "certain," while other areas, inferred by the diffusion method, are marked "uncertain".

Scale recovery. The generated model is rescaled to align with the object size in the test images. Specifically, the system uses the depth map and object mask from the first frame of the test image to approximate an initial model size. Then, multiple scale candidates (slightly larger and smaller) are sampled, and for each, pose hypotheses are generated. The pose selection module [57] identifies the optimal pose hypothesis and corresponding model size.

Mesh refinement. The generated mesh is used to synthesize 24 viewpoints, uniformly sampled from an icosphere and rendered for SDF training. During tracking, newly captured frames with confident pose estimates are added to the training set, gradually replacing synthetic viewpoints. Confidence assessment relies on the seen IoU metric, which measures the overlap between the seen regions of the 3D model and the 2D object mask in test images, high seen IoU indicates reliable pose estimates, while low values trigger online object completion. The system maintains a memory pool of informative test frames selected via uncertainty-aware sampling, prioritizing viewpoints that reveal the most unseen object regions for iterative mesh refinement. As real observations replace synthetic data, the corresponding mesh vertices' labels transition from "uncertain" to "certain", maintaining an uncertainty map that distinguishes between observed and synthetic regions.

3.5.2. HIPPo

HIPPo [28] extends the mesh completion paradigm by eliminating the need for any reference image, instead generating the initial mesh directly from the first RGB-D test frame in a fully zero-shot manner.

Mesh generation. The HIPPo Dreamer module generates a complete 3D mesh from the first RGB-D frame through a two-stage process. First, Wonder3D's multiview diffusion model [30] synthesizes novel viewpoints of the object. Second, a modified MASt3R [20] performs joint depth estimation and 3D reconstruction to produce the initial mesh.

Scale recovery. The scaling factor is determined by computing oriented bounding boxes (OBB) from two sources: the partial point cloud generated via estimated depth and the measured depth from the RGB-D sensor. Aligning these bounding boxes establishes the correct object scale for pose estimation.

Mesh refinement. A viewpoint sphere mechanism monitors camera pose trajectories and identifies keyframes at significant viewpoint shifts to trigger mesh updates. Each

update aligns the measured point cloud with the diffusion-generated mesh in the object coordinate frame, substitutes synthetic geometry and appearance with actual observations, and performs Poisson surface reconstruction [16] to maintain mesh consistency. This strategy progressively replaces the initially complete but potentially inaccurate diffusion prior with geometrically accurate real-world measurements.

3.6. The Gap in Zero-shot Model-free Methods

Despite progress in joint pose estimation and reconstruction, existing methods face a critical trade-off for object tracking applications. SLAM-like methods [56] reconstruct from scratch without priors, providing online refinement but requiring extensive observation before achieving tracking-quality geometry, making them unsuitable for immediate tracking initialization. Single-image reconstruction [60] provides instant complete meshes enabling immediate tracking, but produces static outputs that propagate hallucination errors throughout tracking. Neural field methods (NeRF [34], SDF [51, 62]) support online optimization but require minutes of computation per refinement step, making them impractical for real-time tracking.

Our approach must satisfy four requirements. First, immediate initialization from a single frame using learned priors to obtain an initial complete mesh. Second, online refinement progressively improving mesh geometry by incorporating posed RGB-D observations without batch reprocessing. Third, novel object generalization handling arbitrary objects without category-specific training or CAD models. Fourth, geometric consistency ensuring refined geometry improves rather than degrading pose estimation.

Our contribution combines FoundationPose's [57] tracking capabilities with explicit mesh refinement using confidence-weighted vertex updates. By maintaining an explicit mesh representation rather than implicit neural fields, we avoid expensive retraining while supporting incremental refinement. The key insight is that tracking provides naturally posed observations, and leveraging these for geometric correction requires only local vertex position updates based on nearest-neighbor correspondence, not global reoptimization of neural network weights or latent shape codes.

4 | Zero-shot Model-free Methods

Object pose estimation has applications across robotic manipulation [23, 55], augmented reality [33], and other related fields. In many scenarios, the object of interest cannot be assumed to have been seen before, nor can a textured CAD model be prepared in advance, making model-based [58] approaches impractical for these open-world deployments. Model-free methods reduce this requirement to a set of posed reference images [27, 57], but even this assumption becomes limiting for more general applications, when only a single unposed image is available or when objects are encountered for the first time without any prior capture.

As formalized in Chapter 2, we address the setting where a novel object is observed as an RGB-D video sequence with no prior knowledge of its geometry or appearance. Within this zero-shot model-free setting, two main paradigms have emerged, both reviewed in Chapter 3. The first, exemplified by BundleSDF [56] and 6DOPE-GS [15], follows an online reconstruction strategy: pose and geometry are estimated jointly from scratch as the video unfolds, without any prior object knowledge. The second leverages image-to-3D generative models [4, 30, 60] to produce a complete initial mesh from a single image or first frame, providing immediate tracking capability; methods such as HIPPo [28] and UA-Pose [21] then refine this mesh online as new observations arrive. The complete initial mesh is a key advantage over online reconstruction approaches, which can only represent geometry that has already been observed.

Despite this progress, both paradigms carry significant limitations that affect pose estimation accuracy and robustness. This chapter analyses these challenges in Section 4.1, then presents the proposed method in Section 4.2, and concludes with a comparative evaluation in Section 4.3.

4.1. Challenges

4.1.1. General Challenges

Pose-geometry coupling. Pose estimation and online reconstruction are tightly coupled: poor geometry leads to poor pose estimates, and poor pose estimates prevent correct geometry acquisition from new views [21, 56]. Pose accuracy determines where new RGB-D observations are projected onto the mesh, so errors in either component compound over time.

Symmetry ambiguity. Symmetric or near-symmetric objects introduce rotational ambiguity that pose estimation cannot resolve without additional cues [57], often resulting in tracking failures or flipped poses. Online reconstruction can mitigate this by tracking relative to the first frame and capturing geometry differences on the symmetric side, with SDF-based approaches being particularly effective as they rely less on the initial mesh quality.

Depth sensor reliability. All methods are sensitive to depth noise or missing values [32], which arise on specular and transparent surfaces, near object edges, in cast shadows, and at increasing sensor range. Invalid depth values that are not carefully filtered corrupt both scale estimation and mesh reconstruction.

Recovery from failure. Pose tracking methods have difficulties recovering after complete occlusion or tracking failure, since pose estimates are obtained by refining the previous pose [57]. Any prolonged disruption, such as the object moving partially out of frame, leaves estimation unreliable for a span of frames, making long-term robustness a persistent weakness.

Illumination dependency. Meshes reconstructed from video observations accumulate texture under the lighting conditions of the recording [32]. This causes the render-and-compare loop to degrade under lighting changes, since the rendered reference no longer matches the current frame appearance, and makes the reconstructed mesh possibly inconsistent with a ground-truth model.

4.1.2. Dependence on Segmentation Quality

First-frame mask. The initial object mask seeds all downstream processing, including 3D initialization, scale estimation, and subsequent frame mask propagation, so an error in the first frame propagates irrecoverably, corrupting both geometry and pose from the start [18].

Per-frame mask quality. While pose tracking can proceed without per-frame masks [57], providing them generally improves results. Online mesh updates depend on masks more strictly: false positives inject background depth into the reconstructed model, while false negatives discard valid object geometry, degrading reconstruction fidelity in either case.

4.1.3. Limitations of Online Reconstruction

Incomplete coverage. SLAM-based methods [15, 56] build geometry exclusively from observed viewpoints, so occluded or never-rotated parts remain permanently absent from the model. This affects both pose accuracy in the render-and-compare loop [57] and applications such as robotic manipulation and AR visualization that require a complete mesh [33].

Failure on newly revealed geometry. A typical failure case arises when a previously occluded part becomes visible — for instance, when a hand is removed from an object — and the SLAM method, treating that viewpoint as already observed, fails to update it with the newly revealed geometry, resulting in an incomplete representation and degraded pose estimates.

SDF retraining cost. SDF-based methods [21, 56] incur significant cost when retraining the implicit representation as new frames are accumulated, a process that can take minutes per update. This restricts how frequently geometry can be refined during tracking, creating a fundamental tension between reconstruction quality and responsiveness [28].

4.1.4. Limitations of Image-to-3D Initialization

Scale ambiguity. Diffusion-based image-to-3D methods produce scale-agnostic outputs [30, 60], which is problematic because pose estimation requires the correct physical scale. Additional recovery steps such as oriented bounding box alignment and multi-scale hypothesis testing [18] are therefore necessary, adding complexity to the pipeline.

Geometry discrepancies. For unseen regions, the diffusion prior must hallucinate geometry and appearance, often producing inaccurate shape proportions for asymmetric or textured objects [21, 60]. These hallucinated regions degrade pose estimation when tracking subsequently reveals them. The recent SAM-3D model [4] partially addresses this through better-curated training data.

Sensitivity to initialization viewpoint. Mesh quality depends entirely on what is visible at initialization. Methods that generate the mesh from the first test frame [28] are particularly exposed: partial occlusion or an unfavorable viewpoint compromises the entire subsequent tracking, and online refinement cannot fully compensate for a fundamentally flawed starting mesh. Methods that use a separate reference image [18, 21] reduce this sensitivity, but at the cost of sacrificing full zero-shot generality.

4.2. Proposed Method

The proposed pipeline is a model-free, zero-shot approach that requires only an indication of the target object in the first frame. It follows the image-to-3D initialization paradigm and chains five stages, illustrated in Figure 4.1: mask estimation, mesh generation, scale recovery, pose estimation, and mesh completion. Our main contributions lie in the scale recovery and mesh completion stages, while the remaining components build upon and adapt existing methods from prior work.

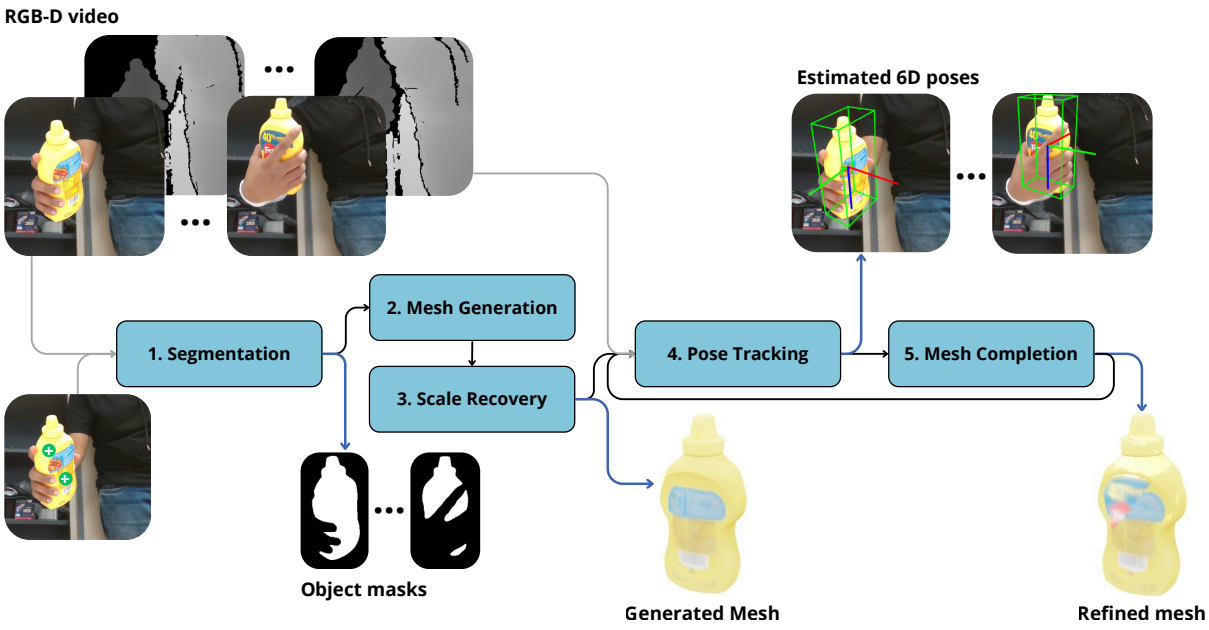

Figure 4.1: Overview of the proposed pipeline. Given an RGB-D video sequence and a segmentation prompt (two points in this example), the method proceeds through five stages: (1) object segmentation, propagating masks across all frames; (2) single-view mesh generation from the first frame; (3) scale recovery to align the mesh to the correct physical size; (4) 6D pose tracking throughout the video; and (5) mesh completion using RGB-D observations, while tracking the object. Images from the HO3D dataset [9].

4.2.1. Object Segmentation

Object segmentation is performed using an off-the-shelf method [61] that extends SAM [17] to video. Given a prompt on the first frame, SAM segments the target object, after which XMem [5] propagates the segmentation across the remaining frames. The prompt can be a set of foreground/background points, a bounding box, or a mask.

4.2.2. Mesh Generation

SAM-3D [4] is applied to the first frame for img-to-3D generation, to obtain a complete 3D mesh of the object. Unlike most single-view reconstruction methods, SAM-3D handles occlusion robustly by inferring plausible geometry for unobserved regions, providing a complete mesh from the very first frame.

4.2.3. Scale Recovery

While SAM-3D provides an initial scale estimate, it is too coarse for accurate pose estimation. We refine it by running the same scale search procedure twice, increasingly refining the scale estimate.

In each round, we first obtain a coarse pose by running a single FoundationPose [57] registration step on the current estimate. Then scale hypotheses s, sampled uniformly from $[0.8, 1.2]$, are applied to this coarse pose to generate a set of pose hypotheses, which are evaluated using the FoundationPose refiner and scorer networks, and the highest-scoring scale is retained. The final scale is the product of the two per-round factors.

This procedure is inspired by Any6D [18] and adapted to SAM-3D meshes. Because SAM-3D produces more accurate object proportions than InstantMesh [60], we restrict the search to a single uniform scale factor rather than independent per-axis scaling, simplifying the procedure.

4.2.4. Pose Estimation

With the correct scale applied to the mesh, FoundationPose [57] estimates the 6D pose at every frame. On the first frame, registration mode samples a large set of pose hypotheses, renders them against the RGB-D observation, and selects the highest-scoring one as the initial pose. From the second frame onward, tracking mode refines the previous pose rather than sampling from scratch, yielding temporally consistent estimates at lower computational cost.

4.2.5. Mesh Completion

This stage represents our main contribution: a geometric confidence-based vertex update scheme. All referenced parameters and their values are listed in Table 4.1.

Every n_{comp} frames, the mesh geometry undergoes a refinement step using the current posed RGB-D observation. For each pixel inside the object mask, the depth value is unprojected into 3D and transformed into the object frame, yielding a set of observed 3D point and color estimates. Since both the pose and the mask are imperfect estimates, and depth sensors introduce additional noise, each observed point is assigned a confidence scalar $\hat{c}$ reflecting observation reliability. Rather than directly updating the mesh, these observations are accumulated into a confidence-weighted point cloud that serves as an intermediate per-vertex buffer, where each entry aggregates observations assigned to a nearby mesh vertex across frames. A vertex is committed to the mesh only once its accumulated confidence reaches c_{commit}, ensuring that only positions supported by consistent evidence are transferred to the mesh geometry, as illustrated in Figure 4.2. Vertices that never reach this threshold retain their generative prior positions, preserving the complete initial shape on unobserved parts. Before tracking begins, the mesh is resampled to a fixed count of N_{vert} vertices to support this vertex-level scheme. An overview of the procedure is given in Algorithm 4.1.

The following paragraphs describe each step in detail. For the definitions of the problem and variables, such as the calibration matrix $\mathbf{K}$ and the object pose $\mathbf{T}$, refer to Chapter 2.

Depth preprocessing. The depth map is denoised following the implementation of Warp [31], first is eroded to remove noise at object boundaries, then smoothed with a bilateral filter. Each valid pixel (u, v) in the object mask is unprojected to a 3D point in the camera frame using the pinhole model defined in Chapter 2, and transformed into the object frame by $\mathbf{T}_{\text{cam}}^{-1}$. Statistical outliers are then removed from the resulting point cloud [63].

Pose quality checks. Before processing, two checks assess whether the pose estimate is accurate enough to proceed. The first computes the IoU between the mesh silhouette $\hat{\mathcal{S}}$, obtained by projecting all vertices into the image via $\mathbf{T}_{\text{cam}}$ and $\mathbf{K}$, and the observed mask $\mathcal{S}$:

$$\text{IoU}(\hat{\mathcal{S}}, \mathcal{S}) = \frac{|\hat{\mathcal{S}} \cap \mathcal{S}|}{|\hat{\mathcal{S}} \cup \mathcal{S}|}.$$

If IoU $< \tau_{\text{skip}}$ the frame is discarded entirely; if $\tau_{\text{skip}} \leq$ IoU $< \tau_{\text{half}}$, all confidence values are halved. τ_{skip} shouldn't be too high, as hallucinated or missing mesh regions reduce

IoU even under a correct pose. The second check assigns each observed point $\mathbf{p}_i$ to its nearest mesh vertex $\mathbf{v}_i^*$ and computes the median point-to-mesh distance:

$$d_{\text{med}} = \text{median}_i \, \|\mathbf{p}_i - \mathbf{v}_i^*\|_2.$$

If $d_{\text{med}} > \tau_{\text{med}}$, the frame is discarded, as the pose is too inaccurate for a reliable update.

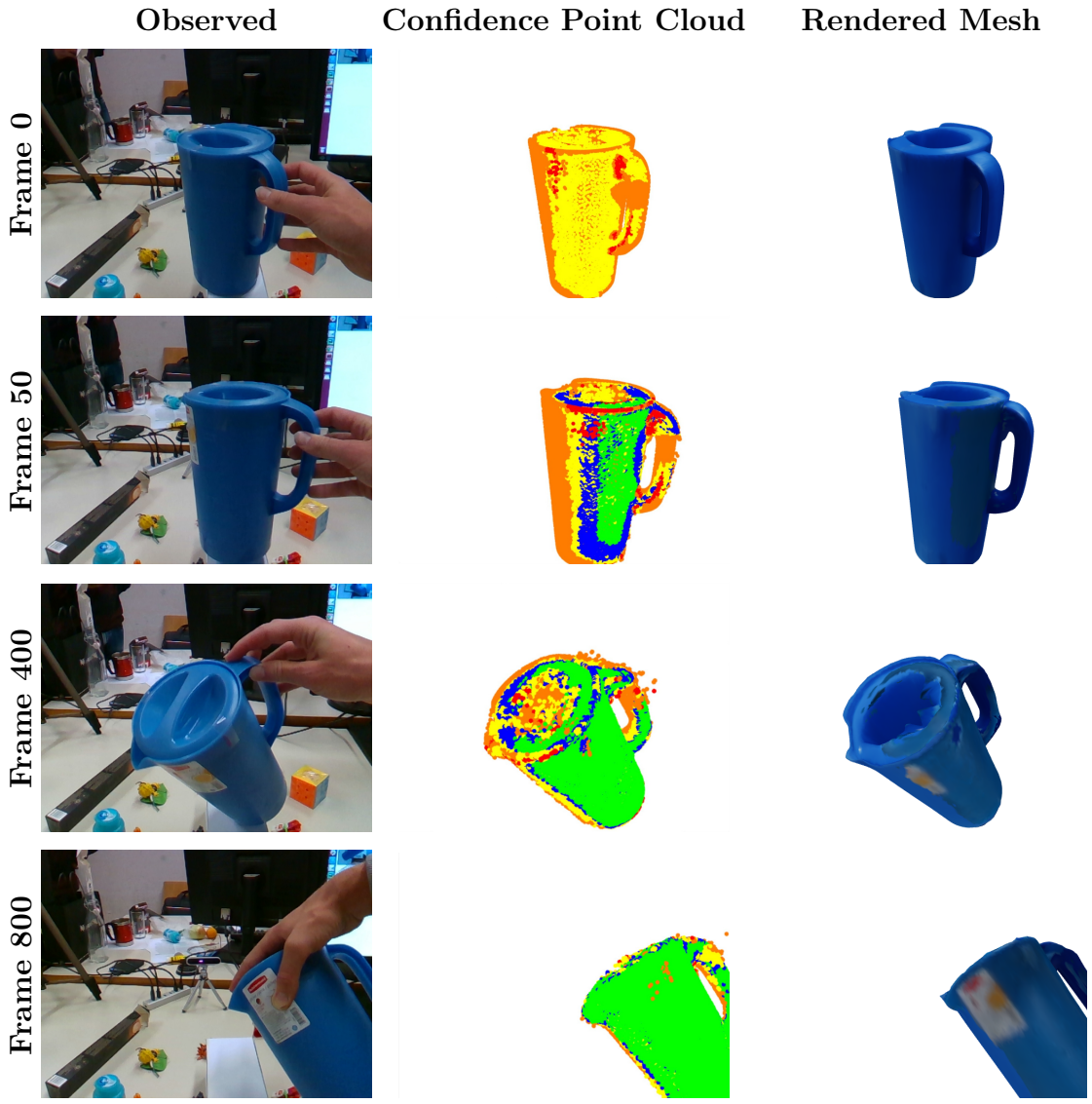

Figure 4.2: Evolution of the confidence-weighted point cloud and the resulting mesh texture refinement across frames. Point colors encode confidence level, from lowest to highest: red, orange, yellow, blue, green.

Confidence and color extraction. Each observed point is assigned a confidence based on its pixel distance $\delta(u, v)$ from the mask boundary, computed via a distance transform:

$$c(u,v) = \begin{cases} 0 & \text{if } \delta(u,v) < \delta_{\text{near}}, \\ c_{\text{low}} & \text{if } \delta_{\text{near}} \le \delta(u,v) < \delta_{\text{far}}, \\ c_{\text{high}} & \text{if } \delta(u,v) \ge \delta_{\text{far}}, \end{cases}$$

halved when $\tau_{\text{skip}} \le \text{IoU} < \tau_{\text{half}}$. Points near the boundary ($c = 0$) are discarded, since mask inaccuracies make those observations unreliable. RGB colors are extracted at the remaining valid pixels alongside their 3D positions.

Stochastic vertex assignment. For each observed point, the K nearest mesh vertices are retrieved and one is selected at random with probabilities $\mathbf{p}_{\text{assign}} = [p_1, \ldots, p_K]$, spreading updates more evenly across the mesh rather than always targeting only the closest vertex. Points whose distance to the assigned vertex exceeds d_{out} are discarded as gross outliers, typically caused by depth sensor or mask failures.

Confidence-weighted vertex update. Let $\hat{\mathbf{q}}$, $\hat{\mathbf{c}}_{\text{rgb}}$, $\hat{c}$ denote the stored position, color, and confidence of a vertex, and $\bar{\mathbf{q}}$, $\bar{\mathbf{c}}_{\text{rgb}}$, $\bar{c}$ the mean position, color, and confidence of all observations assigned to it in the current frame. Vertices with $\hat{c} \le 0$ store the observation directly. For vertices with $\hat{c} > 0$, if $\|\bar{\mathbf{q}} - \hat{\mathbf{q}}\|_2 \le d_{\text{thresh}}$, position and color are updated by confidence-weighted averaging and confidence is accumulated:

$$\hat{\mathbf{q}} \leftarrow \frac{\hat{c}\,\hat{\mathbf{q}} + \bar{c}\,\bar{\mathbf{q}}}{\hat{c} + \bar{c}}, \qquad \hat{\mathbf{c}}_{\text{rgb}} \leftarrow \frac{\hat{c}\,\hat{\mathbf{c}}_{\text{rgb}} + \bar{c}\,\bar{\mathbf{c}}_{\text{rgb}}}{\hat{c} + \bar{c}}, \qquad \hat{c} \leftarrow \min(\hat{c} + \bar{c},\, c_{\max}).$$

Otherwise the observation is rejected and $\hat{c} \leftarrow \hat{c} - \bar{c}$. Vertices reaching $\hat{c} \ge c_{\text{commit}}$ are committed to the mesh.

Wrong point correction. Vertices with $\hat{c} \le 0$ have never been positively observed and may carry incorrect positions from mesh generation. Because point assignment favors the nearest vertex, such vertices are rarely reached through normal updates. Here, vertices that received no observation in the current frame are projected into the image and compared against the observed depth at the corresponding pixel. Letting $\Delta_d = d_{\text{vertex}} - d_{\text{obs}}$, invisible protrusions are identified as vertices that project confidently inside the mask ($\delta(u, v) > \delta_{\text{near}}$) and are closer to the camera than the observed depth ($\Delta_d < -d_{\text{thresh}}$), yet have never been observed, pointing to spurious protrusions from the image-to-3D method. These are penalized by $\hat{c} \leftarrow \hat{c} - c_{\text{high}}$. Once $\hat{c}$ drops below c_{reset}, the vertex is

relocated to the observed 3D position at the projected pixel.

Taubin smoothing. The updated mesh undergoes n_{smooth} iterations of Taubin smoothing [47], alternating a forward Laplacian step with weight λ_{T} and a backward step with weight μ_{T}. This reduces depth noise without the volume shrinkage of plain Laplacian smoothing. FoundationPose is then reinitialized with the updated mesh so that subsequent tracking uses the refined geometry.

In Table 4.1 are listed the parameters of our approach. All values are tuned for the experimental setup described in Section 5 and depend on factors such as image resolution, frame rate, object size, and the accuracy of the mask estimator, mesh generator, and pose estimator.

Table 4.1: Parameters of the mesh completion procedure.

Parameter	Description	Value
n_{comp}	Attachment frequency (frames)	10
N_{vert}	Target mesh vertex count	60,000
τ_{skip}	IoU threshold for frame rejection	0.5
τ_{half}	IoU threshold for confidence halving	0.8
τ_{med}	Median point-to-mesh threshold for frame rejection	4 mm
δ_{near}	Inner boundary margin	12 px
δ_{far}	Interior distance threshold	35 px
c_{low}	Confidence for near-interior observations	0.12
c_{high}	Confidence for interior observations	0.35
K	Nearest vertices queried per point	3
$\mathbf{p}_{\text{assign}}$	Stochastic assignment probabilities	$[0.70,\ 0.20,\ 0.10]$
d_{thresh}	Dissimilarity threshold	2 mm
d_{out}	Gross outlier distance threshold	$15 \times d_{\text{thresh}}$
c_{max}	Maximum confidence clamp	2
c_{commit}	Minimum confidence for mesh commit	1
c_{reset}	Confidence threshold triggering vertex reset	-1
n_{smooth}	Taubin smoothing iterations	3
λ_{T}	Taubin forward step weight	0.5
μ_{T}	Taubin backward step weight	-0.53

Algorithm 4.1 Confidence-based Mesh Completion

Preprocess depth: erode, filter, unproject to object frame, remove outliers $\rightarrow \mathcal{P}$
Compute IoU$(\hat{\mathcal{S}}, \mathcal{S})$; **skip frame** if IoU $< \tau_{\text{skip}}$; halve confidence if IoU $< \tau_{\text{half}}$
Compute d_{med}; **skip frame** if $d_{\text{med}} > \tau_{\text{med}}$
for each point $\mathbf{p}_i \in \mathcal{P}$ **do**
    Assign confidence $c(u, v)$ via distance transform $\delta(u, v)$ from mask boundary
    Assign $\mathbf{p}_i$ to one of the K nearest mesh vertices $\mathbf{v}^*$; discard if $\|\mathbf{p}_i - \mathbf{v}^*\| > d_{\text{out}}$
end for
for each vertex $\mathbf{v}$ with stored state $(\hat{\mathbf{q}}, \hat{\mathbf{c}}_{\text{rgb}}, \hat{c})$ that received observations **do**
    Compute mean observation $(\bar{\mathbf{q}}, \bar{\mathbf{c}}_{\text{rgb}}, \bar{c})$ over all points assigned to $\mathbf{v}$
    if $\hat{c} \leq 0$ **then**
        $(\hat{\mathbf{q}}, \hat{\mathbf{c}}_{\text{rgb}}, \hat{c}) \leftarrow (\bar{\mathbf{q}}, \bar{\mathbf{c}}_{\text{rgb}}, \bar{c})$
    else if $\|\bar{\mathbf{q}} - \hat{\mathbf{q}}\|_2 \leq d_{\text{thresh}}$ **then**
        Update $\hat{\mathbf{q}}$, $\hat{\mathbf{c}}_{\text{rgb}}$ by confidence-weighted average; $\hat{c} \leftarrow \max(\hat{c} + \bar{c}, c_{\max})$
        if $\hat{c} \geq c_{\text{commit}}$ **then**
            Commit vertex $(\hat{\mathbf{q}}, \hat{\mathbf{c}}_{\text{rgb}}, \hat{c})$ to $\mathcal{M}$
        end if
    else
        $\hat{c} \leftarrow \hat{c} - \bar{c}$
    end if
end for
for each unobserved vertex $\mathbf{v}$ with $\hat{c} \leq 0$ **do**
    Project $\mathbf{v}$ onto the image; penalize $\hat{c}$ if invisible protrusion
    if $\hat{c} < c_{\text{reset}}$ **then**
        Relocate $\mathbf{v}$ to observed 3D point at projected pixel
    end if
end for
Apply Taubin smoothing; reinitialize FoundationPose with updated $\mathcal{M}$

4.3. Comparative Analysis

We compare the proposed approach against five methods spanning the two zero-shot paradigms described in the introduction of this chapter. This section provides a conceptual comparison structured stage by stage; an experimental evaluation is presented in Chapter 5. Any6D [18], UA-Pose [21], and HIPPo [28] follow the image-to-3D initialization paradigm, generating a complete mesh from a single image before tracking.

BundleSDF [56] and 6DOPE-GS [15] instead build the object representation purely from observations, without any generative prior, jointly estimating pose and geometry from scratch and therefore requiring neither mesh generation nor scale recovery.

These methods are not fully comparable: they differ in input requirements, problem scope, and pipeline stages covered, as summarized in Tables 4.2 and 4.3. The two most similar methods to ours are also compared in the synthetic Figure 4.3, which provides a visual overview of the stage options of the three image-to-3D pipelines that do completion, the possibilities of each stage are better analyzed in the discussion that follows.

BundleSDF, Any6D, and UA-Pose are open source and directly evaluated in Section 5.3. 6DOPE-GS is not publicly available, but its authors evaluate on the same test set used in this work, including a run of BundleSDF from its open-source implementation, and a direct numerical comparison is provided in Section 5.4. HIPPo, despite sharing the same paradigm and problem setup as the proposed method, is not publicly available and is therefore compared only at the algorithmic level. The comparison is structured stage by stage, discussing only the methods relevant to each step. For a detailed description of each method, refer to Chapter 3.

Table 4.2: Inputs required by each method. **K** is the camera calibration matrix. The prompt for the first-frame mask can be points, bounding boxes, or masks [17].

Method	RGB-D video	**K**	Object reference	First-frame mask
Proposed	✓	✓	none (zero-shot)	prompt
Any6D	✓	✓	single RGB-D image	GT mask
UA-Pose	✓	✓	single RGB image	prompt
HIPPo	✓	✓	none (zero-shot)	text prompt
BundleSDF	✓	✓	none (zero-shot)	GT mask
6DOPE-GS	✓	✓	none (zero-shot)	prompt

Table 4.3: Problem formulation of each method.

Method	Paradigm	Goal
Proposed	Image-to-3D	6D pose tracking, mesh completion
Any6D	Image-to-3D	6D pose estimation, metric scale recovery
UA-Pose	Image-to-3D	6D pose tracking, mesh completion
HIPPo	Image-to-3D	6D pose tracking, mesh completion
BundleSDF	Online recon.	6D pose tracking, 3D reconstruction
6DOPE-GS	Online recon.	6D pose tracking, 3D reconstruction

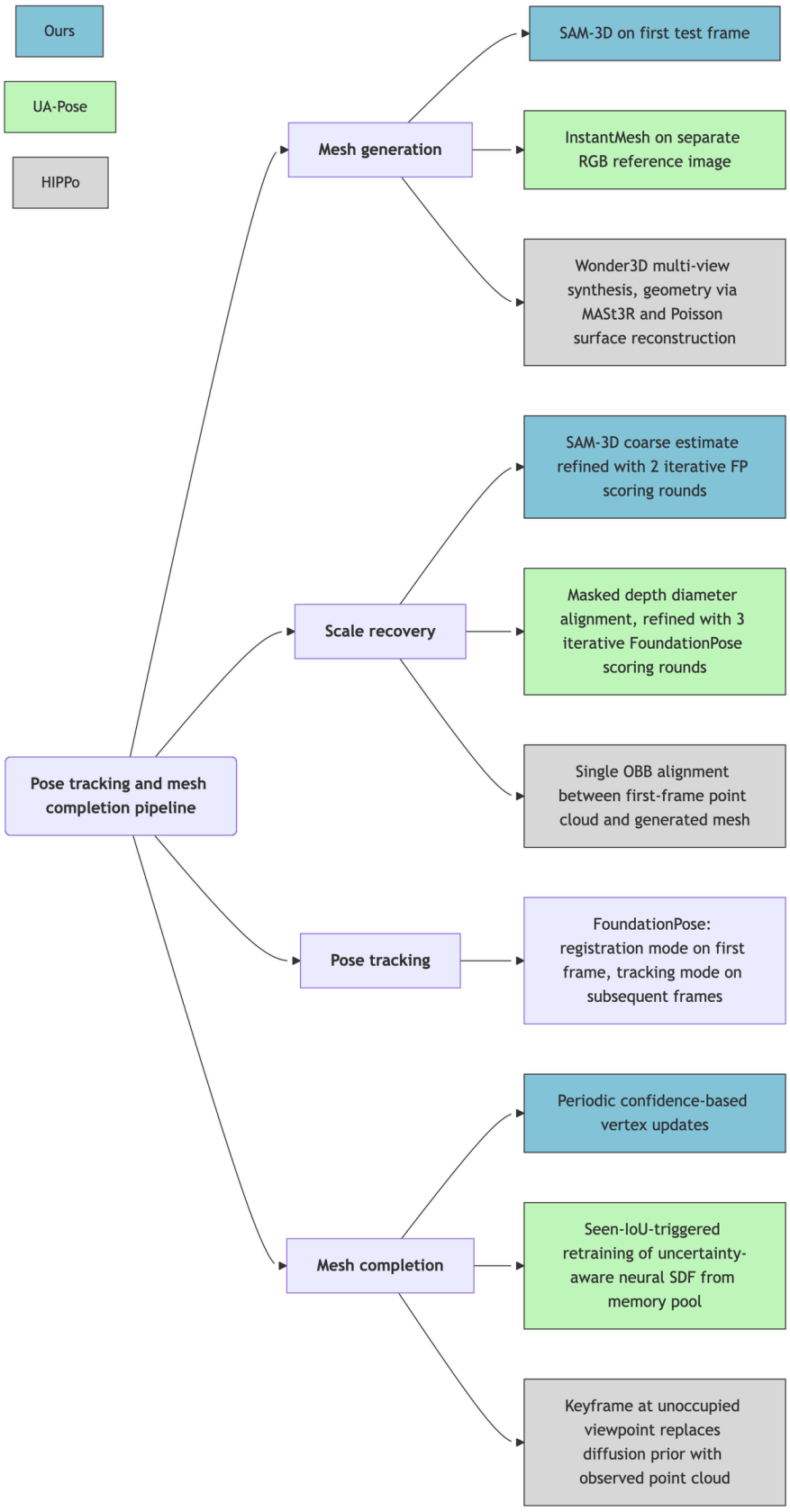

Figure 4.3: Stage-by-stage pipeline comparison of the three image-to-3D completion methods: the proposed approach, UA-Pose [21], and HIPPo [28].

4.3.1. Object Segmentation

The proposed approach and UA-Pose use SAM [17] for the initial mask, then propagate it across all frames with XMem [5]. BundleSDF follows the same XMem propagation strategy, starting instead from a user-annotated mask. 6DOPE-GS relies on SAM2 [37], which handles both initial segmentation and temporal tracking within a unified model. HIPPo applies Grounding DINO [25] with a text prompt on every frame independently, with no temporal propagation.

Table 4.4: Object segmentation tools and temporal propagation strategy.

Method	Prompt Type	Initial Segmentation	Temporal Propagation
Proposed	Points / boxes / mask	SAM	XMem
UA-Pose	Points / boxes / mask	SAM	XMem
BundleSDF	GT mask	Manual annotation	XMem
6DOPE-GS	Points / boxes / mask	SAM2	SAM2 (unified)
HIPPo	Text	Grounding DINO	None (per-frame)

4.3.2. Mesh Generation

The image-to-3D methods generate a complete object mesh prior to tracking. Any6D and UA-Pose both apply InstantMesh [60] to a separate unposed RGB reference image. For UA-Pose, regions visible in the reference are labeled certain while generative hallucinations are labeled uncertain, directly populating the uncertainty map that drives its completion module. HIPPo and the proposed approach instead derive the mesh from the first test frame, removing the need for a separate reference capture: HIPPo synthesizes multi-view images with Wonder3D [30], reconstructs geometry via a modified MASt3R [20], and applies Poisson surface reconstruction [16]. The proposed approach uses SAM-3D [4] directly.

Table 4.5: Mesh generation strategy for image-to-3D methods.

Method	Reference Source	Generator
Proposed	First test frame	SAM-3D
Any6D	Separate RGB-D image	InstantMesh
UA-Pose	Separate RGB image	InstantMesh
HIPPo	First test frame	Wonder3D $\rightarrow$ MASt3R $\rightarrow$ Poisson

4.3.3. Scale Recovery

All image-to-3D methods must recover the physical scale of the generated mesh before tracking, since generative models produce scale-agnostic outputs.

HIPPo performs a single oriented bounding box (OBB) alignment between the object point cloud from the first-frame depth map and the generated mesh, yielding a uniform scaling factor.

The proposed approach applies the coarse scale estimated from SAM-3D and refines it with two successive rounds of FoundationPose [57] hypothesis scoring over $[0.8, 1.2]$, with the final scale being the product of both per-round factors.

UA-Pose initializes the scale from the maximum depth extent of the masked first-frame point cloud, then runs three iterative rounds of FoundationPose scoring over 11 scale candidates in $[0.8, 1.2]$ relative to the current estimate, progressively narrowing the search.

Any6D is the only method that recovers non-uniform, per-axis scale, making it uniquely capable of correcting shape distortions introduced by the mesh generator. It proceeds in four stages. A coarse OBB alignment with axis-permutation search initializes the y-axis scale. An initial FoundationPose pose is then estimated on the coarse mesh. The observed point cloud is transformed into the object frame via this pose, where per-axis OBB ratios are computed and applied. Finally, 252 scale hypotheses in $[0.6, 1.4]$ are scored by FoundationPose and the best is selected.

Table 4.6: Scale recovery strategies. FP denotes FoundationPose [57].

Method	Initialization	Refinement
Proposed	SAM-3D coarse estimate	2 iterative FP scoring rounds
Any6D	Axis-permutation on OBB alignment	FP $\rightarrow$ per-axis OBB ratios $\rightarrow$ FP scoring
UA-Pose	Masked depth diameter alignment	3 iterative FP scoring rounds
HIPPo	Single OBB alignment	None

4.3.4. Pose Estimation

The proposed approach, Any6D, UA-Pose, and HIPPo all use FoundationPose [57]: the first frame is handled in registration mode, which samples and ranks a large set of pose hypotheses via render-and-compare, while subsequent frames use tracking mode to refine the previous estimate.

BundleSDF and 6DOPE-GS do not rely on an external pose estimator and instead derive

poses jointly with the object representation built online. Both initialize a coarse per-frame pose via feature matching and RANSAC, then refine it through online pose graph optimization over a keyframe memory pool, with pose updates informed by the learned object representation.

Table 4.7: Pose estimation strategy for each method. FP denotes FoundationPose [57].

Method	Estimator	First Frame	Subsequent Frames
Proposed Any6D UA-Pose HIPPo	FoundationPose	Registration: hypothesis sampling + render-and-compare	Tracking: refinement of previous estimate
BundleSDF 6DOPE-GS	Online joint estimation	Coarse init via feature matching + RANSAC	Pose graph optimization on keyframe memory pool

4.3.5. Online Mesh Reconstruction

UA-Pose. The mesh is represented as a neural SDF trained jointly on real observations stored in a memory pool and on synthetic RGB-D views rendered from the diffusion-generated mesh. Each vertex visible from any real viewpoint is labeled as certain, while the rest are labeled uncertain, forming an uncertainty map. During tracking, a seen-IoU metric measures the overlap between the certain regions of the rendered mesh and the current frame mask. When seen-IoU falls below a threshold, online completion is triggered: an uncertainty-aware sampling strategy selects the most informative frames from the memory pool prioritizing views that reveal the most unseen geometry, the SDF is retrained from scratch for a fixed number of steps, and the mesh is re-extracted via marching cubes.

HIPPo. A viewpoint sphere with uniformly distributed viewpoints monitors relative pose changes between frames. When the current pose aligns with an unoccupied viewpoint, a keyframe is recognized and a mesh update is triggered. The measured point cloud is used to update the diffusion prior point cloud and the result is converted to a mesh via Poisson surface reconstruction [16].

BundleSDF. A Neural Object Field, encoding both geometry and appearance, is trained in a background thread concurrently with pose graph optimization. It consumes all memory frames and their current poses, applying a hybrid SDF formulation that distinguishes

uncertain free space, empty space, and near-surface regions to handle noisy segmentation and occlusions. Once training converges, updated poses are returned to the pose graph, and the process repeats as new frames are added to the memory pool. The final mesh is extracted via marching cubes.

6DOPE-GS. A Gaussian Object Field based on 2D Gaussian Splatting is jointly optimized with keyframe poses. A dynamic keyframe selection mechanism clusters poses around an icosahedron to maximize spatial coverage and rejects outlier frames via the median absolute deviation of the reconstruction loss. The fast differentiable rasterization of Gaussian Splatting allows the field to converge significantly faster than a neural SDF.

BundleSDF and 6DOPE-GS build their representations purely from observations, producing geometry faithful to measurements but incomplete until sufficient viewpoints have been observed. If a region of the object is permanently occluded throughout the sequence, it remains absent from the model indefinitely. This is the fundamental trade-off between the two paradigms: image-to-3D methods provide full object coverage from the first frame at the cost of hallucinated unseen regions, whereas online-reconstruction methods converge to accurate geometry incrementally.

Table 4.8: Online mesh reconstruction strategies.

Method	Representation	Trigger	Keyframe Selection	Mesh Extraction
Proposed	Confidence-informed mesh	Periodic	Pose quality checks. IoU and point-to-mesh distance	Confidence-weighted vertex update
UA-Pose	Uncertainty-aware mesh	Seen-IoU	Uncertainty-aware sampling; prioritizes unseen geometry	Learned neural SDF + Marching cubes
HIPPo	Point cloud	Unoccupied viewpoint slot	One keyframe per new unoccupied viewpoint	Poisson reconstruction [16]
BundleSDF	Neural SDF	Continuous	All memory frames with current poses	Marching cubes
6DOPE-GS	Gaussian Object Field (2DGS)	Continuous	Icosahedron-clustered poses	Differentiable rasterization

5 | Experiments and Results

This chapter demonstrates that zero-shot model-free 6D pose tracking can approach model-based performance. On the HO3D benchmark [9], our approach achieves an 82.0% on ADD-S pose metric, closing to within 2.5 percentage points of the model-based baseline (84.5%) that uses the ground-truth CAD model, while substantially outperforming previous model-free methods that still require an external object reference [18, 21]. Beyond pose, the proposed scale recovery consistently yields higher 3D IoU than both competing model-free methods, and the completion stage further improves mesh geometry surpassing the baseline on appearance (PSNR), as the mesh texture is learned directly from the video under the actual scene lighting rather than from a fixed CAD model texture. Finally, a direct comparison against zero-shot online reconstruction methods confirms that the proposed pipeline outperforms both BundleSDF and 6DOPE-GS on reported metrics under a compatible evaluation protocol.

The chapter is organized as follows. Section 5.1 defines the evaluation metrics, covering pose accuracy (ADD, ADD-S), mesh geometry (Chamfer Distance, 3D IoU), appearance fidelity (PSNR), and the joint pose-and-mesh metric ADI. Section 5.2 describes the HO3D dataset and its relevance to the zero-shot setting. Section 5.3 presents the evaluated configurations and reports quantitative and qualitative results across all metrics and videos. Section 5.4 extends this comparison to BundleSDF [56] and 6DOPE-GS [15].

5.1. Metrics

Evaluation is organized along three axes: pose accuracy (ADD, ADD-S), mesh quality (Chamfer Distance, 3D IoU, PSNR), and joint pose-mesh quality (ADI). Pose can be assessed independently of the estimated mesh, while mesh quality and joint metrics require posed comparisons since meshes exist in different coordinate systems. The pose metrics are standard in this literature [56, 57] and follow the BOP (Benchmark for 6D Object Pose Estimation) protocol [13], which defines unified dataset formats, standardized pose-error functions, and an open evaluation system enabling reproducible comparison. The remaining metrics are adopted from relevant literature [34, 50] and extend the evaluation

aspects of mesh quality.

ADD and ADD-S. The primary pose evaluation metric is the Average Distance of Model Points (ADD) [12], which measures the mean point-to-point distance between the ground-truth mesh $\mathcal{M}$ transformed by the estimated and ground-truth poses over n vertices $\{p_i\}$:

$$\text{ADD} = \frac{1}{n}\sum_{i=1}^{n} \|(R_{\text{est}}\, p_i + t_{\text{est}}) - (R_{\text{gt}}\, p_i + t_{\text{gt}})\|$$

ADD-S, its symmetric variant, replaces exact point correspondence with nearest-neighbor distance:

$$\text{ADD-S} = \frac{1}{n}\sum_{i=1}^{n} \min_j \|(R_{\text{est}}\, p_i + t_{\text{est}}) - (R_{\text{gt}}\, p_j + t_{\text{gt}})\|$$

ADD is the stricter measure, penalizing rotation errors under exact correspondence, while ADD-S is invariant to point ordering and always satisfies ADD-S $\leq$ ADD. Since the objects in HO3D are symmetric or near-symmetric, both metrics are applied to all objects, following standard practice [18, 21, 56]. As both variants transform the same ground-truth mesh under two poses, they isolate pose error from mesh quality. Rather than a fixed threshold, which cannot characterize performance on near-threshold poses [58], we report the Area Under the Curve (AUC) of the cumulative success rate for both ADD and ADD-S, sweeping thresholds from 0 to $0.1d$ in increments of $0.001d$, where d is the object diameter, yielding a percentage score.

Chamfer Distance. CD evaluates the geometric quality of the estimated mesh independently of tracking drift, comparing $\hat{\mathcal{M}}$ and $\mathcal{M}$ on the first frame using T_0^{est} and T_0^{gt}:

$$\text{CD} = \frac{1}{2}\left(\frac{1}{n}\sum_i \min_j \|p_i^{\hat{\mathcal{M}}} - p_j^{\mathcal{M}}\| + \frac{1}{m}\sum_j \min_i \|p_j^{\mathcal{M}} - p_i^{\hat{\mathcal{M}}}\|\right)$$

The first frame is chosen because its pose estimate benefits from multiple refinement iterations and is therefore more accurate than subsequent tracking poses. The bidirectional formulation penalizes both missing geometry and spurious reconstructions [21, 56].

3D IoU. Intersection over Union of 3D bounding boxes is a standard metric for joint detection and size evaluation in category-level 6D pose and size estimation [50], where no instance CAD model is available and scale must be recovered. Here we adapt it to evaluate scale accuracy of the estimated mesh independently of point-level geometry, complementing Chamfer Distance. Using T_0^{est} and T_0^{gt}, it is computed as the volumetric overlap of the axis-aligned bounding boxes, or oriented bounding box (OBB), of the two

transformed point clouds:

$$\text{3D IoU} = \frac{|\text{OBB}(\hat{\mathcal{M}}_{T_0^{\text{est}}}) \cap \text{OBB}(\mathcal{M}_{T_0^{\text{gt}}})|}{|\text{OBB}(\hat{\mathcal{M}}_{T_0^{\text{est}}}) \cup \text{OBB}(\mathcal{M}_{T_0^{\text{gt}}})|}$$

A high 3D IoU indicates that the scale recovery stage has correctly aligned the estimated mesh to the physical object dimensions.

PSNR. Peak Signal-to-Noise Ratio (PSNR) evaluates the photometric fidelity of the mesh texture by comparing a rendering of the estimated mesh against the reference RGB image. The mesh is rendered at the estimated pose and PSNR is computed over the pixels Ω covered by the object mask:

$$\text{MSE} = \frac{1}{|\Omega|} \sum_{(u,v)\in\Omega} \left\| I_{\text{render}}(u,v) - I_{\text{ref}}(u,v) \right\|_2^2, \quad \text{PSNR} = 10 \cdot \log_{10}\left(\frac{255^2}{\text{MSE}}\right)$$

Higher values in decibels indicate closer agreement with the reference. PSNR is the standard pixel-level quality metric in image quality assessment and novel-view synthesis evaluation [34, 52], complementing the geometry-only Chamfer Distance by measuring appearance fidelity.

ADI. ADI, or average distance of model points for objects with indistinguishable views [12], assesses pose and mesh quality jointly over the full sequence. Unlike ADD and ADD-S, it compares *different* point clouds: $\hat{\mathcal{M}}$ transformed by the estimated pose against $\mathcal{M}$ transformed by the ground-truth pose. It computes the mean nearest-neighbor distance from each transformed ground-truth point to the transformed estimated cloud:

$$\text{ADI} = \frac{1}{m} \sum_{i=1}^{m} \min_j \left\| (R_{\text{gt}}\, p_i^{\mathcal{M}} + t_{\text{gt}}) - (R_{\text{est}}\, p_j^{\hat{\mathcal{M}}} + t_{\text{est}}) \right\|$$

By capturing the combined effect of mesh inaccuracies and pose errors, ADI is suitable for monitoring the overall progression of mesh completion along the sequence.

5.2. Dataset

Pose estimation has given rise to a large number of datasets, each targeting a specific sub-problem or scenario. Among these, YCB-Video [58] stands as a widely adopted benchmark, providing RGB-D sequences of YCB objects in challenging multi-object scenes. For zero-shot and single-reference methods specifically, the HO3D dataset [9], which fea-

tures the same YCB objects, has become the standard dataset, having been adopted by BundleSDF [56] and subsequently by 6DOPE-GS [15] and UA-Pose [21].

We evaluate on the test set of HO3D v3, an RGB-D dataset designed to study hand-object interaction, consisting of 13 video sequences of 4 YCB objects shown in Figure 5.1 and listed in Table 5.1. The dataset is well suited to our setting for several reasons: objects are frequently occluded by the manipulating hand or move partially out of frame, challenging both pose estimation and mask propagation; objects are actively rotated and repositioned throughout each sequence, progressively revealing new surface regions that are informative for mesh completion; and the first frame often provides limited, partially occluded visibility of the object, making initialization challenging for zero-shot methods.

Table 5.1: Objects and sequences in HO3D [9].

Object	Sequences	Total frames
Pitcher base	AP10, AP11, AP12, AP13, AP14	1616
Potted meat can	MPM10, MPM11, MPM12, MPM13, MPM14	1618
Bleach cleanser	SB11, SB13	1680
Mustard bottle	SM1	898

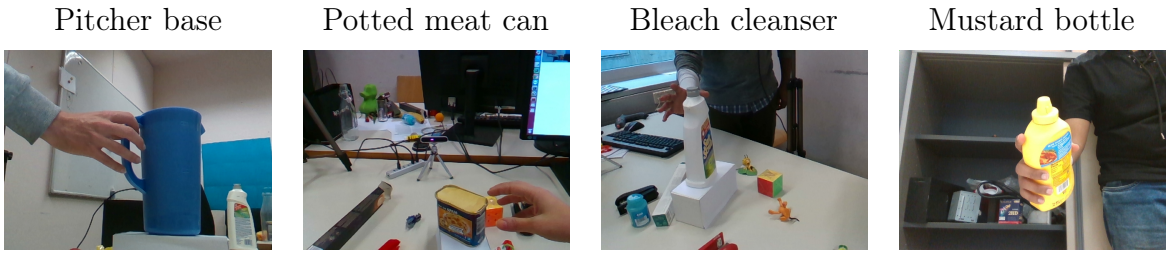

Figure 5.1: First frames of one representative sequence per object in the HO3D dataset [9].

The HO3D dataset is organized hierarchically following the structure below. For each object, it provides a high-resolution triangular mesh, used both as ground truth for mesh quality evaluation and as model input for model-based baselines. For each sequence, the camera intrinsic matrix $\mathbf{K}$ is provided. Each frame supplies an RGB image capturing visible appearance, a depth map encoding per-pixel metric distances to the scene, a segmentation mask pre-computed with XMem [5] following the protocol of BundleSDF [56], and a ground-truth 6D object pose annotation for evaluation.

```
HO3D
|--evaluation/
|--|--{video_id}       (e.g., AP10, MPM11, SB13, SM1)
|--|--|--rgb/
|--|--|--|--{frame_id}.jpg
|--|--|--depth/
|--|--|--|--{frame_id}.png
|--|--|--mask_obj/
|--|--|--|--{frame_id}.png
|--|--|--meta/
|--|--|--|--{frame_id}.pkl
|--models/
|--|--{object_id}/
|--|--|--textured_simple.obj
```

5.3. Results

Five configurations are evaluated on HO3D, designed to isolate the contribution of each pipeline component and to quantify the gap between zero-shot model-free operation and model-based tracking. To ensure a fair comparison, all configurations share the same segmentation masks estimated with XMem [5] and use FoundationPose [57] as the common pose estimator, so that differences in pose metrics reflect mesh quality rather than the choice of estimator. We first introduce the evaluated configurations, then report quantitative results for each metric, and finally provide qualitative per-video analyses relating metric values to observed reconstruction and tracking behavior.

5.3.1. Evaluated Methods

Table 5.2 summarizes the key differences between the five configurations.

The **Baseline** runs FoundationPose with the ground-truth CAD model, making it the only model-based configuration and the upper bound on performance when object geometry is fully known. **Any6D** [18] and **UA-Pose** [21] are model-free but not fully zero-shot: they remove the need for a CAD model but still require an object reference captured outside the test sequence. **Ours** and **OursComp** are the proposed methods, operating without any object-specific prior beyond the segmentation mask, respectively without and with the mesh completion step. A full description of all methods is provided in Chapters 3 and 4; we summarize only the aspects most relevant to interpreting the results.

Table 5.2: Overview of the evaluated configurations. All methods share the same XMem [5] masks and use FoundationPose [57] for tracking. *Object reference* denotes any input beyond the RGB-D test video, camera intrinsics **K** and segmentation mask.

Method	Object reference	Mesh generation	Online completion
Baseline [57]	ground-truth CAD model	none	none
Any6D [18]	RGB-D anchor image	InstantMesh [60]	none
UA-Pose [21]	RGB reference image	InstantMesh [60]	neural SDF [56]
Ours	none (zero-shot)	SAM-3D [4]	none
OursComp	none (zero-shot)	SAM-3D [4]	geometric vertex update

Any6D generates the object mesh from a single RGB-D anchor image using InstantMesh [60] and recovers non-uniform per-axis scale through oriented bounding box alignment followed by FoundationPose hypothesis scoring over 252 scale candidates, with no online mesh update during tracking. UA-Pose uses a single RGB reference image and the same InstantMesh pipeline as Any6D, enabling a direct comparison of the completion stage on an identical starting mesh. It applies online completion via an uncertainty-aware neural SDF, retrained from scratch whenever the seen-IoU between the rendered mesh and the current observation falls below a threshold; its scale recovery is however less accurate than Any6D's, which penalizes initial mesh quality. Since single-reference operation was not the primary focus of the original work, the code was not open-source, and the reference images used in the authors' own evaluation were unavailable, UA-Pose results reflect our best reproduction effort.

Ours generates the mesh directly from the first frame of the test sequence using SAM-3D [4] and refines the coarse SAM-3D scale estimate through two consecutive rounds of FoundationPose hypothesis scoring, with no online mesh update. OursComp extends Ours with the confidence-based completion module of Section 4.2.5, which incrementally updates vertex positions and colors every n_{comp} frames, and constitutes the full proposed pipeline.

5.3.2. Pose Tracking

Since estimated poses are expressed in the generated mesh coordinate system, direct comparison against ground-truth poses requires a normalization step. We anchor each trajectory to the first-frame estimate via $\hat{T}_i = T_i^{\text{est}} \, (T_0^{\text{est}})^{-1} \, T_0^{\text{gt}}$, following a similar approach to [18]. This is the only way to evaluate pose independently of mesh quality; the alternative, ADI, foregoes normalization by applying each pose to its respective mesh and

is discussed in Section 5.3.4. All ADD and ADD-S values are reported as AUC over the per-frame scores.

Table 5.3 shows the overall averages. OursComp reaches 60.5% ADD and 82.0% ADD-S, closing to within 8.5 and 2.5 percentage points of the model-based Baseline, despite operating with no prior knowledge of the object. Ours alone already outperforms Any6D by 18 points on ADD and 10 on ADD-S, even though Any6D uses an external reference image and a more elaborate scale recovery, confirming that the quality of the underlying mesh generation [4] drives most of the performance. UA-Pose remains roughly on par with Any6D despite applying online completion, though its less accurate scale recovery results in a worse initial mesh, as shown by the following metrics in this section.

The large gap between ADD and ADD-S scores for UA-Pose and Any6D reflects the near-symmetry of the HO3D objects. Because these methods rely on lower-quality mesh generation that tend to produce symmetric geometry, they cannot resolve symmetric ambiguities, causing strict point-to-point ADD scores to drop while the ADD-S remains comparatively high. Our methods are less affected thanks to the more accurate geometry and texture provided by SAM-3D. Per-video results in Tables 5.4 and 5.5 also reveal that the averages of both competitors are pulled down by sequences where they fail completely (e.g. MPM10 UA-Pose ADD = 0.21%, SM1 Any6D ADD-S = 29.0%), whereas our methods remain consistently strong across most sequences.

Comparing OursComp against Ours, completion provides an overall gain (+4.1 ADD, +1.8 ADD-S) and is especially effective on challenging sequences: on AP12 it triples ADD (11.2 to 37.6%) and on AP13 nearly doubles it (35.3 to 68.6%). On sequences where the initial SAM-3D mesh is already accurate, the benefit is smaller and occasionally negative (e.g. AP11, SB11), suggesting that completion brings the largest return when there is geometry left to recover. In a few cases, SM1 and AP13, our methods even surpass the Baseline, which we attribute to the mesh texture being learned directly from the video under the actual scene lighting, whereas the Baseline relies on a fixed CAD model texture.

Table 5.3: Average of ADD, ADD-S across all videos.

Method	ADD ↑ (%)	ADD-S ↑ (%)
UA-Pose	37.930	68.133
Any6D	38.158	70.638
Ours	56.397	80.273
OursComp	**60.516**	**82.031**
Baseline	68.976	84.490

Table 5.4: Pose estimation results on all videos. Metric: AUC of ADD ↑ (%)

Method	UA-Pose	Any6D	Ours	OursComp	Baseline
AP10	52.276	13.605	59.853	**61.864**	72.347
AP11	4.779	57.786	**67.716**	53.644	63.326
AP12	12.789	8.395	11.244	**37.624**	63.230
AP13	14.997	53.858	35.288	**68.637**	62.382
AP14	67.939	0.129	56.811	**68.415**	72.948
MPM10	0.213	24.206	**58.310**	49.577	63.278
MPM11	25.935	45.347	55.811	**58.198**	64.001
MPM12	0.624	40.922	53.478	**61.310**	72.769
MPM13	59.123	53.909	**64.667**	60.431	71.818
MPM14	**53.215**	49.549	45.048	52.860	55.237
SB11	76.800	67.670	**78.911**	67.613	80.136
SB13	63.857	69.163	**74.599**	72.656	86.227
SM1	60.545	11.519	71.426	**73.877**	68.993
Mean	**37.930**	**38.158**	**56.397**	**60.516**	**68.976**

Table 5.5: Pose estimation results on all videos. Metric: AUC of ADD-S ↑ (%)

Method	UA-Pose	Any6D	Ours	OursComp	Baseline
AP10	81.170	62.491	82.835	**84.652**	86.202
AP11	66.499	83.353	**85.382**	81.861	83.808
AP12	67.030	58.131	66.161	**76.902**	85.029
AP13	68.441	82.434	75.315	**84.512**	84.871
AP14	**84.506**	67.272	80.322	84.022	85.780
MPM10	2.003	64.069	**79.670**	79.316	80.885
MPM11	66.108	74.390	78.015	**78.926**	79.541
MPM12	40.852	72.439	77.564	**78.899**	84.457
MPM13	78.160	77.801	**81.368**	79.203	83.611
MPM14	77.133	74.472	74.324	**77.164**	77.582
SB11	88.203	85.047	**89.743**	86.618	89.463
SB13	83.257	**87.421**	85.493	86.065	92.366
SM1	82.365	28.978	87.354	**88.266**	84.774
Mean	**68.133**	**70.638**	**80.273**	**82.031**	**84.490**

5.3.3. Mesh Geometry and Scale Recovery

Both metrics are computed at the first-frame pose. CD uses the final mesh, which differs from the initial only for completion methods (UA-Pose, OursComp), thus capturing the full benefit of completion on top of the initial geometry. 3D IoU, on the other hand, uses the initial mesh, so it purely reflects scale recovery quality before any completion. OursComp is omitted from the scale table as it shares the exact same scale recovery as Ours. The Baseline registers the ground-truth mesh against itself, so its metrics reflect only the first-frame pose error of FoundationPose, providing a lower bound on geometry error.

Table 5.6 reports per-video CD results. OursComp (mean 0.352 cm) consistently outperforms Ours (0.400 cm), which in turn outperforms Any6D (0.559 cm) and UA-Pose (0.832 cm). Completion degrades CD in only two cases, MPM10 and MPM13, which are also sequences where SAM-3D produces a close-to-baseline initial mesh, leaving little geometry to recover. On SB11, both Ours and OursComp actually surpass the Baseline (0.254 and 0.250 vs. 0.264 cm), a result made possible by the accurate scale recovery: hypothesis-based scale tuning can yield a first-frame alignment that is slightly tighter than using the ground-truth mesh directly, since FoundationPose may favor a marginally non-physical scale. While Any6D occasionally produces a better initial mesh than Ours (AP11, AP13), completion closes and mostly reverses this gap. The elevated UA-Pose CD average is dominated by the MPM10 outlier (3.037 cm), a direct consequence of its failed scale initialization on that sequence, as shown in Table 5.7.

Table 5.6: Mesh geometry results on all videos. Metric: CD ↓ (cm)

Method	UA-Pose	Any6D	Ours	OursComp	Baseline
AP10	0.585	0.919	0.546	**0.428**	0.292
AP11	1.134	**0.404**	0.519	0.438	0.352
AP12	1.158	1.250	0.777	**0.686**	0.227
AP13	0.727	0.389	0.445	**0.367**	0.263
AP14	0.588	0.770	0.588	**0.542**	0.294
MPM10	3.037	0.385	**0.219**	0.224	0.217
MPM11	0.459	0.303	0.209	**0.183**	0.175
MPM12	0.721	0.324	0.303	**0.247**	0.142
MPM13	0.441	0.294	**0.166**	0.172	0.161
MPM14	0.418	0.312	0.261	**0.227**	0.153
SB11	0.516	0.646	0.254	**0.250**	0.264
SB13	**0.527**	0.616	0.662	0.584	0.227
SM1	0.505	0.648	0.250	**0.226**	0.164
Mean	**0.832**	**0.559**	**0.400**	**0.352**	**0.225**

Table 5.7 reports the per-video 3D IoU results. Ours achieves a mean 3D IoU of 86.2%, outperforming Any6D (78.8%) and UA-Pose (53.6%), and closing to within 5.4 points of the Baseline (91.6%). Ours leads on 11 of 13 sequences; the two exceptions, AP11 and MPM12, correspond to challenging first frames shown in Figure 5.2, where the pitcher handle is occluded and the object is partially out of frame, respectively. UA-Pose starts from the same InstantMesh mesh as Any6D but consistently achieves worse scale, confirming that its scale recovery is the primary bottleneck.

Table 5.7: Scale recovery results on all videos. Metric: 3D_IOU ↑ (%)

Method	UA-Pose	Any6D	Ours	Baseline
AP10	60.119	69.570	**90.860**	92.271
AP11	55.851	**84.142**	83.254	95.105
AP12	52.014	73.068	**74.456**	92.457
AP13	48.990	88.758	**89.649**	95.405
AP14	78.525	79.389	**89.703**	92.631
MPM10	17.181	78.206	**90.545**	88.939
MPM11	30.073	83.089	**87.169**	90.244
MPM12	39.119	**83.996**	81.369	93.368
MPM13	79.319	77.778	**88.755**	92.155
MPM14	63.863	78.466	**86.770**	90.661
SB11	73.131	71.199	**85.234**	89.388
SB13	54.918	74.394	**87.317**	87.312
SM1	44.065	82.257	**85.908**	90.694
Mean	**53.628**	**78.793**	**86.230**	**91.587**

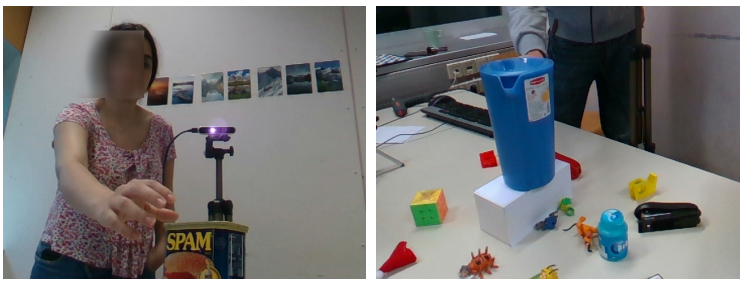

Figure 5.2: Challenging first frames for mesh generation. Respectively MPM12, with the object partially out of frame, and AP11, with the pitcher handle not visible.

5.3.4. Mesh Completion

ADI measures pose and mesh quality jointly: computed like ADD-S but applying the estimated pose to the estimated mesh rather than to the ground-truth, it accumulates errors from both geometry and tracking. For methods with active completion (UA-Pose and OursComp), the mesh is updated at each frame, so ADI captures the progressive improvement as new observations are incorporated. Since OursComp and Ours share the same initial mesh, their ADI values coincide in early frames and diverge as completion takes effect, as visible in Figure 5.3.

Table 5.8 reports per-video ADI results. OursComp achieves a mean ADI of 81.2%, compared to 76.7% for Ours, 70.2% for Any6D, and 69.0% for UA-Pose, closing to within 4.2 points of the Baseline (85.4%). The gain from completion is consistent across all sequences except SB11, where Ours already reaches 90.6% (above the Baseline), leaving little geometry to recover.

Table 5.8: Combined pose tracking and mesh reconstruction results on all videos. Metric: AUC of ADI ↑ (%)

Method	UA-Pose	Any6D	Ours	OursComp	Baseline
AP10	70.968	71.417	76.310	**81.787**	88.107
AP11	64.655	79.870	79.176	**82.150**	86.918
AP12	57.679	51.208	46.321	**59.297**	86.976
AP13	75.728	80.561	69.609	**82.769**	85.795
AP14	82.263	75.565	81.833	**86.242**	86.662
MPM10	66.049	72.056	80.456	**81.411**	81.330
MPM11	66.983	74.766	82.131	**83.599**	81.421
MPM12	68.543	72.998	73.909	**80.254**	83.721
MPM13	73.319	76.141	85.001	**85.006**	84.070
MPM14	69.078	73.350	75.609	**81.069**	80.565
SB11	77.111	76.924	**90.591**	89.758	87.557
SB13	60.017	**80.196**	69.984	73.892	91.180
SM1	64.986	27.891	86.806	**88.415**	86.303
Mean	**69.029**	**70.226**	**76.749**	**81.204**	**85.431**

SB13 is the only sequence where our methods do not lead: Any6D's better initial mesh gives it a higher ADI in early frames, which its AUC advantage ultimately reflects. However, Figure 5.3 shows that once OursComp's completion module engages, the initial

deficit is progressively recovered and ADI converges to the level of competing methods by the end of the sequence. This is further corroborated by the CD values: OursComp's final mesh (0.584 cm) surpasses Any6D's initial mesh (0.616 cm), even though the AUC does not fully capture this late recovery due to the early-frame disadvantage. SB13 is also UA-Pose's strongest sequence: despite starting from a worse scale estimate, its completion module compensates effectively, reaching the highest ADI values among all methods by the end of the video and achieving the best final CD overall.

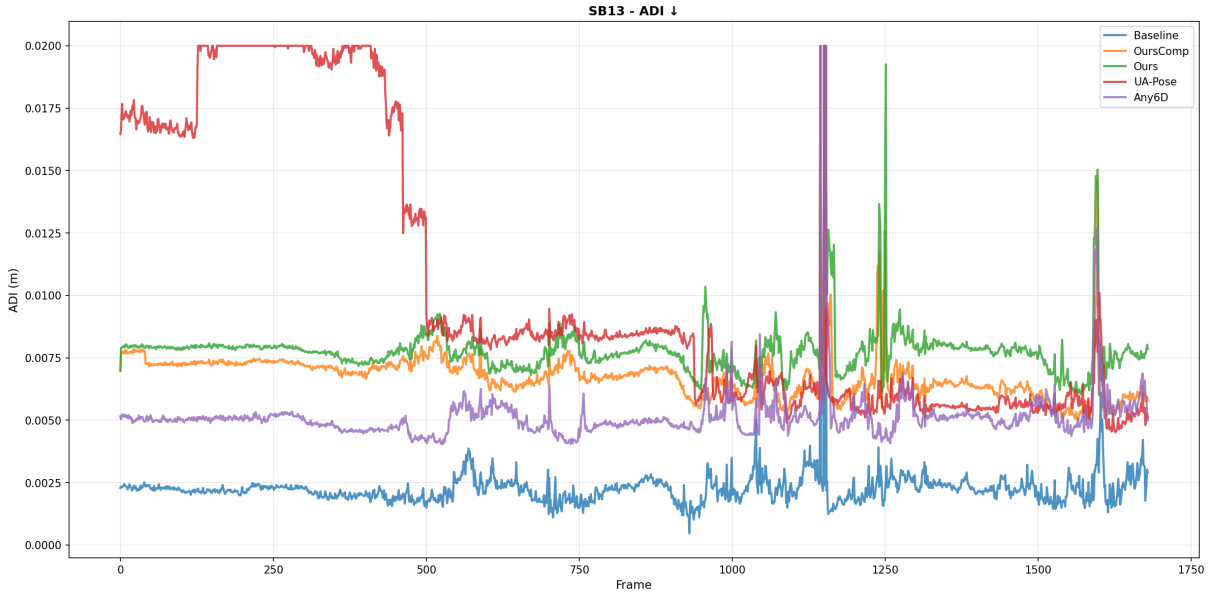

Figure 5.3: ADI metric over time on video SB13. Values are computed with the current mesh for each method and cropped at 0.02 m for visibility.

5.3.5. Mesh Texture

Since prior model-free methods struggled with geometry, appearance quality was not previously evaluated in this setting. As our CD results show a competitive mesh geometry, we introduce PSNR [34, 52] to assess texture fidelity. For each method with active completion (UA-Pose and OursComp), PSNR is evaluated twice at each sampled frame: once with the initial mesh and once with the current (completion-updated) mesh, isolating the contribution of completion to appearance. The third subtable of table 5.9 reports the upper bound: the ground-truth mesh rendered at the ground-truth pose, unaffected by any pose or mesh error. Due to rendering cost, PSNR is sampled once every 10 frames.

The OursComp initial mesh achieves a mean PSNR of 14.7 dB, comparable to the ground-truth baseline (15.3 dB), despite relying solely on the first frame for texture. After completion, OursComp reaches 18.6 dB, surpassing the baseline on 12 of 13 sequences and

by 3.3 dB on average. The exception is SB11 (16.4 vs. 17.1 dB), where the baseline texture matches the scene lighting well. The most striking case is SM1, where OursComp scores 15.6 dB against a baseline of only 6.3 dB, because the fixed CAD texture is severely mismatched to the actual illumination, as shown in Figure 5.6. Vertex colors are never frozen: more recent confident observations carry higher weight in the running average, so the texture continuously adapts to the current lighting as tracking progresses.

Despite the poor accuracy on other metrics, UA-Pose achieves good PSNR values on its current mesh, which is expected since PSNR is evaluated only over the visible regions at each frame and the neural SDF approach recovers accurate texture on the parts of the object observed throughout the sequence. However, its current-mesh PSNR (mean 15.4 dB on the evaluated sequences) remains below OursComp (18.6 dB). This is partly because the neural SDF retrains only when the seen-IoU falls below a threshold, making texture updates less frequent than OursComp's every-n_{comp}-frame geometric update.

Table 5.9: Mesh appearance results. Metric: PSNR ↑ (dB). For UA-Pose and OursComp, *Initial* uses the mesh before completion and *Current* uses the completion-updated mesh at each frame. The GT subtable renders the ground-truth mesh at the ground-truth pose.

	UA-Pose		**OursComp**		**GT Poses**
Video	**Initial**	**Current**	**Initial**	**Current**	**GT Mesh**
AP10	14.058	17.462	17.835	**20.866**	18.867
AP11	13.339	17.892	15.934	**19.784**	17.561
AP12	14.814	17.989	17.410	**23.006**	18.386
AP13	15.694	15.344	15.554	**17.970**	14.578
AP14	12.247	17.534	16.090	**20.006**	17.313
MPM10	11.911	14.824	14.078	**16.900**	14.719
MPM11	10.615	15.550	14.815	**17.600**	14.175
MPM12	11.517	16.769	14.226	**19.808**	15.885
MPM13	11.871	14.836	13.874	**17.228**	13.726
MPM14	11.881	16.519	12.776	**17.261**	15.097
SB11	12.036	13.222	11.557	**16.387**	17.140
SB13	10.633	12.064	13.776	**18.799**	14.898
SM1	11.519	10.398	12.937	**15.568**	6.290
Mean	**11.472**	**15.416**	**12.472**	**15.416**	**15.280**

5.3.6. Video MPM12

Figure 5.4 shows the tracked and rendered meshes at frames 0, 800 and 1400 for all methods. Each mesh is rendered at the estimated pose and masked for direct comparison with the observed frame (last row).

- **Mesh generation.** InstantMesh recovers the object shape but not the texture, since the reference image shows the opposite side from the viewpoints used for generation, making the mesh near-symmetric. This is the same underlying cause as for the large ADD/ADD-S gap of UA-Pose and Any6D in Table 5.3. SAM-3D avoids this by generating directly from the first frame. Notably, SAM-3D achieves CD= 0.303 cm even though roughly half the object is occluded in that frame.
- **Scale recovery.** At frame 0, all methods except UA-Pose recover the correct scale, consistent with Table 5.7. The UA-Pose mesh is visibly clipped, with its boundary falling inside the masked region.
- **Mesh completion.** OursComp clearly improves texture and geometry over Ours, in line with Table 5.9. UA-Pose's neural SDF retrieves detailed texture on this video, but its overall metric is limited by poor geometry on the regions not seen during the initial reference view, as discussed further in Section 5.3.8.
- **Mesh texture.** The Baseline illustrates the lighting mismatch of a fixed CAD texture: the rendered appearance diverges from the observed image despite correct geometry, confirming the advantage of learning texture directly from the video.

Figure 5.5 shows the ADI metric over time for all methods on MPM12. OursComp starts from the same level as Ours and progressively approaches the Baseline as the mesh is refined, confirming that completion drives the improvement rather than pose estimation alone. UA-Pose starts from a low ADI due to its poor scale, but completion allows it to recover and approach the level of the non-completion image-to-3D methods by the end of the sequence. Frames with ADI above 0.02 m are cropped for visibility and correspond to highly challenging viewpoints where all methods fail.

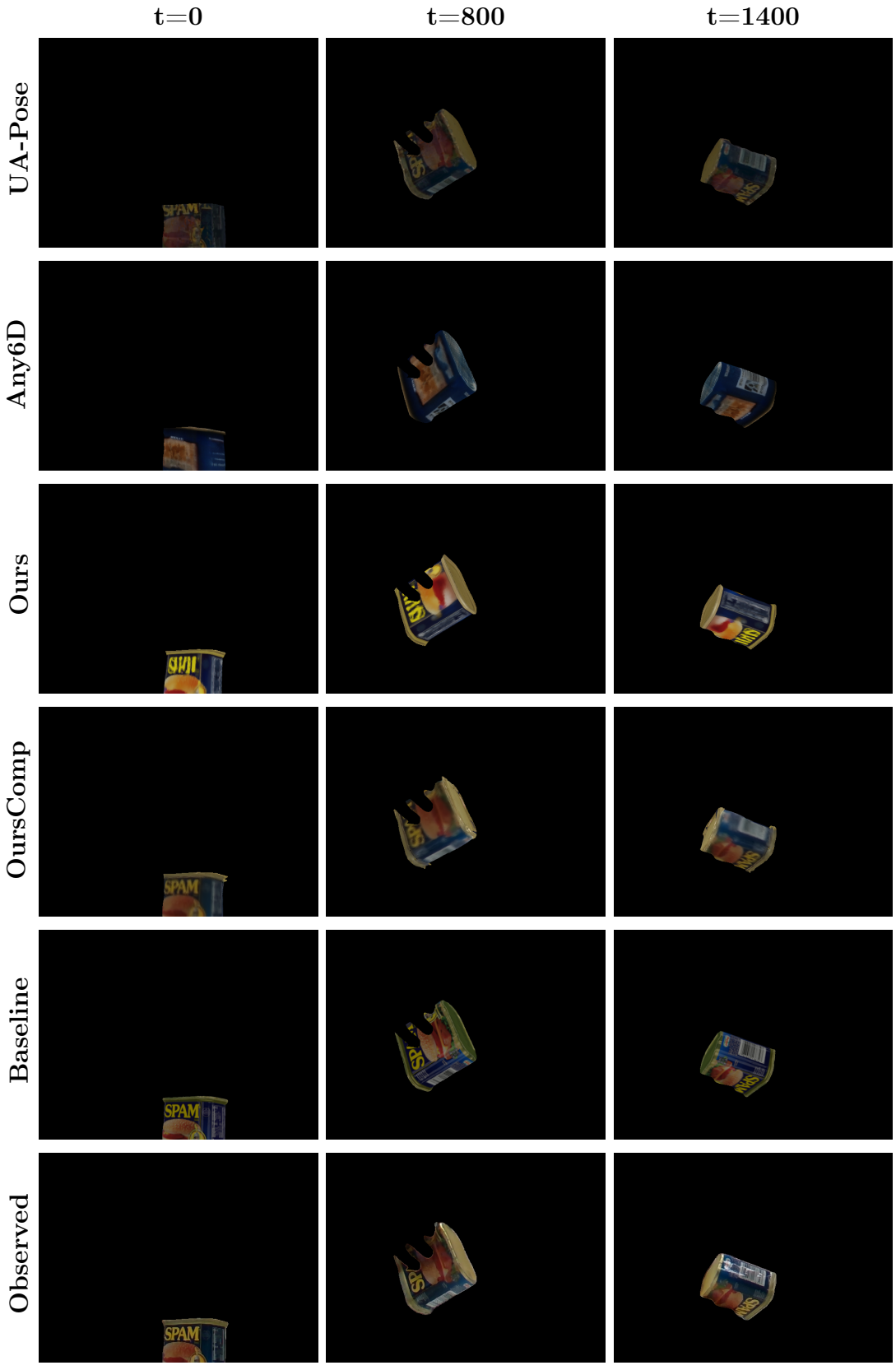

Figure 5.4: Qualitative comparison of pose and mesh on video MPM12.

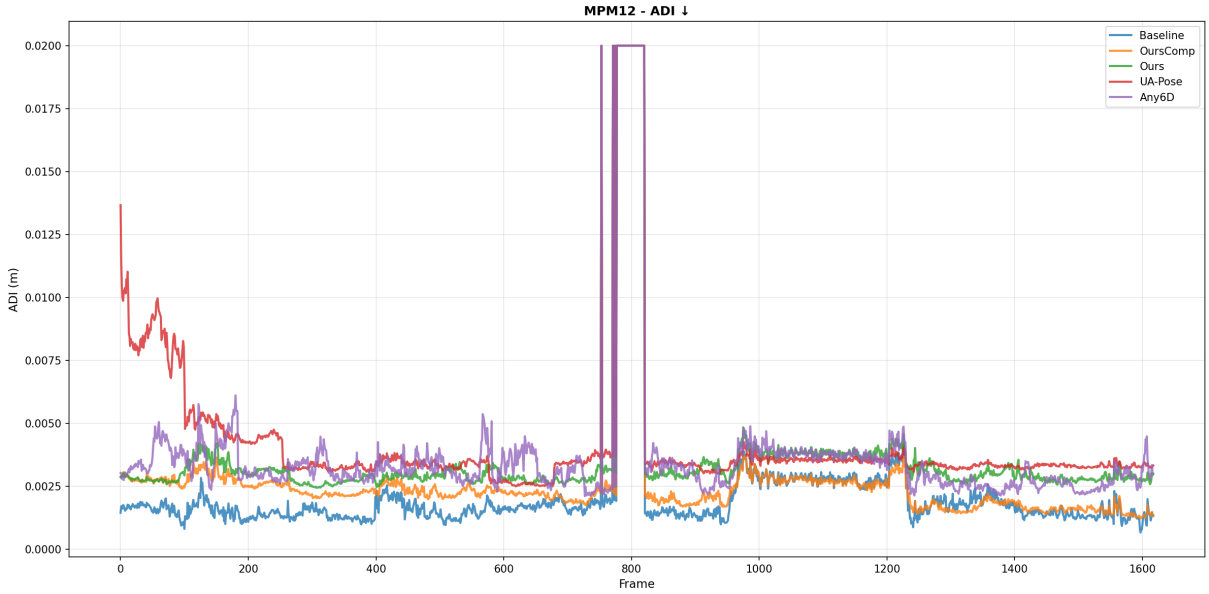

Figure 5.5: ADI metric over time on video MPM12. Values are computed with the current mesh for each method and cropped at 0.02 m for visibility.

5.3.7. Video SM1

Figure 5.6 shows the tracked and rendered meshes at frames 0, 300 and 800 for all methods on the 898-frame SM1 sequence.

- **Mesh Generation.** Any6D produces a visibly deformed mesh, so SAM-3D clearly outperforms InstantMesh in this case - even though the first frame is partially occluded - which is consistent with Ours higher CD reported in Table 5.6. Both image-to-3D methods exhibit the texture symmetry issue identified in MPM12, though on opposite sides of the object. In Ou, SAM-3D is initialized on the first frame, resulting in good texture quality at frame 0 but degraded texture at frame 300 (the opposite side of the bottle). In Any6D, InstantMesh uses an anchor image from the opposite side of the bottle, which leads to poor texture on the first frame but better quality at frame 300.
- **Mesh completion.** UA-Pose recovers the observed surfaces reasonably well, as expected from a neural SDF approach [56], despite the poor scale initialization. OursComp progressively improves on Ours, bringing the rendered mesh closer to the observed frames.
- **Mesh texture.** The object is under strong directional lighting, which causes a severe mismatch for the Baseline's fixed CAD texture (PSNR 6.3 dB), while OursComp

learns the lighting directly from the video and achieves 15.6 dB, as reported in Table 5.9.

The ADI progression in Figure 5.7 reinforces these observations. The deformed Any6D mesh causes a tracking failure at around frame 400, after which the object becomes nearly invisible in the masked region at frame 800. UA-Pose recovers reasonably from its poor initialization via completion, but does not reach the level of the better-initialized methods. OursComp again starts at the same level as Ours and progressively converges toward the Baseline as the mesh is refined.

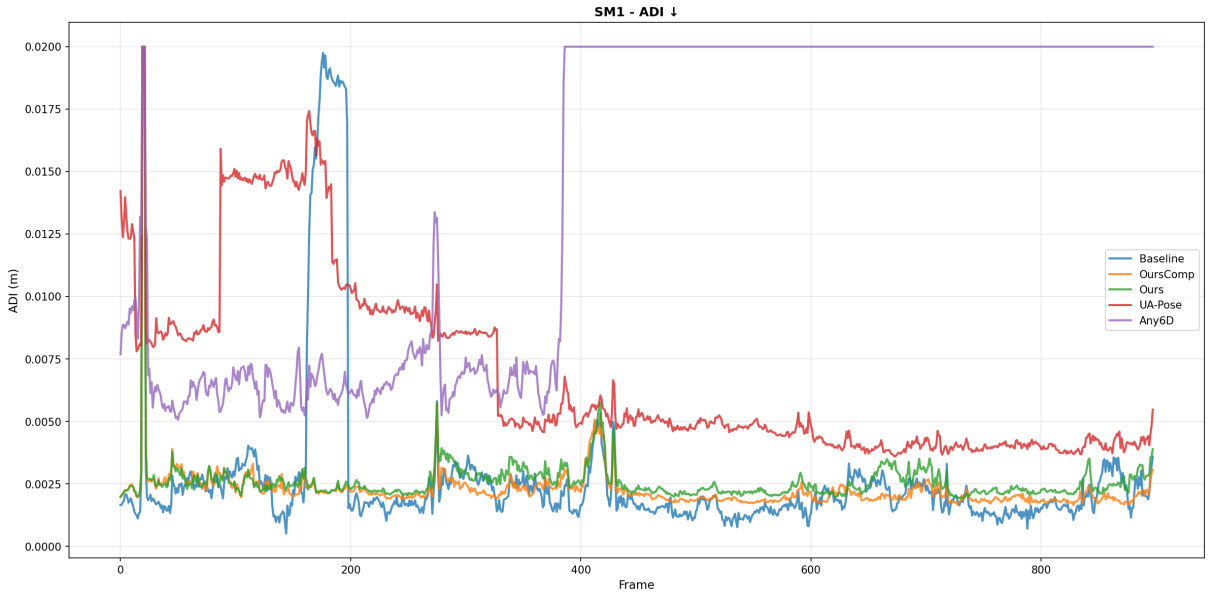

Figure 5.7: ADI metric over time on video SM1. Values are computed with the current mesh for each method and cropped at 0.02 m for visibility.

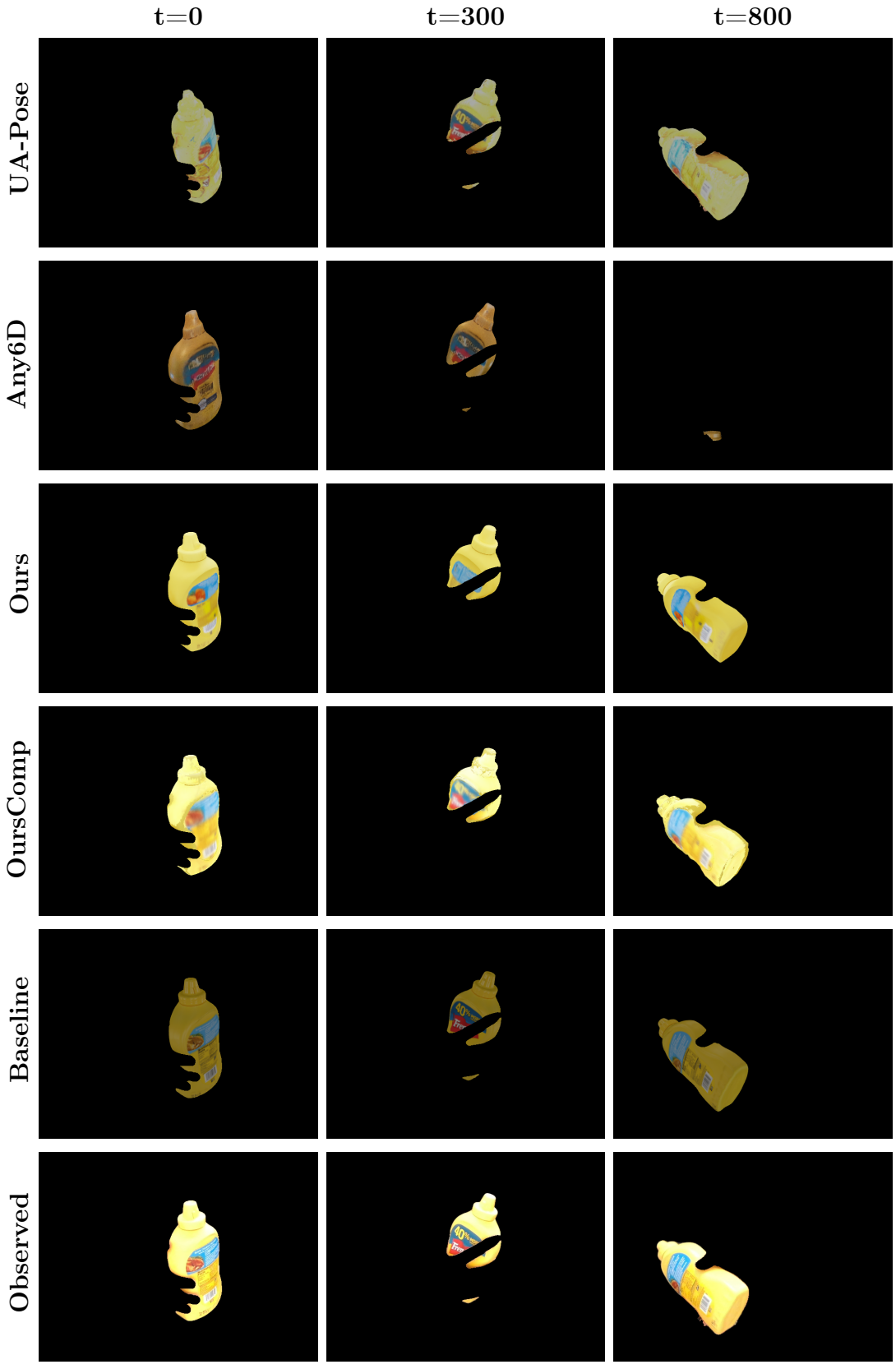

Figure 5.6: Qualitative comparison of pose and mesh on video SM1.

5.3.8. UA-Pose Incomplete Meshes

Neural SDF completion methods struggle with occlusion: since the mesh is not anchored to a complete prior shape, occluded regions are progressively removed during reconstruction, a known limitation shared with BundleSDF [56]. Figure 5.8 shows two cases where this affects UA-Pose. In SM1, occlusion by the hand causes the mustard bottle mesh to become incomplete over time; the good masked-rendering results seen earlier masked this issue, but the unmasked mesh reveals significant missing geometry. In AP13, inaccurate XMem masks cause the hand to be incorporated into the reconstructed object mesh. Ours and OursComp are shown for comparison: both maintain a complete mesh throughout, with OursComp refining geometry and appearance on the observed parts while preserving the SAM-3D prior on unseen regions.

5.3.9. Inference Time Performance

This work focuses on accuracy rather than efficiency, and a detailed profiling is outside its scope. Nevertheless, we briefly report the runtime of each pipeline stage to contextualize computational cost, noting that mesh generation and scale recovery are applied once at the start of the video and therefore do not affect directly online tracking latency. All timings were measured on an NVIDIA RTX A5000 GPU.

FoundationPose. As a reference, FoundationPose registration takes on average 4 s per video, while tracking runs at approximately 0.03 s per frame, corresponding to a rate of 33 frames per second after the first pose is established.

Mesh generation and scale recovery. Both SAM-3D [4] and InstantMesh [60] require approximately 40 s for the full generation step, with no significant performance difference between the two. For scale recovery, average runtimes are 20 s for Ours, 28 s for UA-Pose, and 42 s for Any6D, a ordering consistent with the complexity of each method. Our approach benefits from starting with a higher-quality mesh, which reduces the work required at this stage, while Any6D performs a more involved recovery process precisely because it is designed to be robust to lower-quality mesh inputs.

Mesh completion. The completion stage is the most computationally relevant to online deployment. UA-Pose's neural SDF retrains a network from scratch on every update, taking on average 235.9 s per update and accumulating over one hour of total completion time on the longer sequences. Our geometric vertex update completes in approximately 1.4 s on average per update and, despite being triggered far more frequently (150 updates

on average vs. 19 for UA-Pose), yields a mean total completion time of only 3.5 min. Table 5.10 reports the per-video breakdown. In OursComp, the per-update time is highly variable, as it depends on the number of pixels in the mask, which is determined by how close the object is to the camera and how occluded it is.

Our approach is not yet suitable for real-time deployment, but is substantially closer to that goal than UA-Pose. Reducing update frequency, tightening the confidence criteria to trigger completion less often, and improving code efficiency could all reduce runtime further without necessarily affecting accuracy, confirming that the geometric approach is competitive from both an accuracy and an efficiency standpoint.

Table 5.10: Per-video completion times. Mean (s) is the average time per update, #Upd is the number of completion updates triggered, and Tot (min) is the cumulative completion time. UA-Pose retrains a neural SDF from scratch on each update; OursComp applies the geometric vertex update.

	UA-Pose			**OursComp**		
Video	**Mean (s)**	**#Upd**	**Tot (min)**	**Mean (s)**	**#Upd**	**Tot (min)**
AP10	294.6	23	112.9	2.1	153	5.5
AP11	202.3	10	33.7	0.6	159	1.7
AP12	287.1	5	23.9	4.1	159	10.8
AP13	380.9	4	25.4	2.2	158	5.8
AP14	352.3	13	76.3	5.4	147	13.2
MPM10	172.9	26	74.9	0.4	127	0.7
MPM11	175.8	43	126.0	0.3	157	0.8
MPM12	149.6	6	15.0	0.5	162	1.4
MPM13	126.3	13	27.4	0.3	153	0.9
MPM14	187.5	10	31.2	0.3	155	0.9
SB11	234.5	25	97.7	0.4	167	1.1
SB13	254.7	14	59.4	0.8	166	2.1
SM1	248.6	54	223.7	0.5	90	0.7
Mean	**235.9**	**18.9**	**74.4**	**1.4**	**150.2**	**3.5**

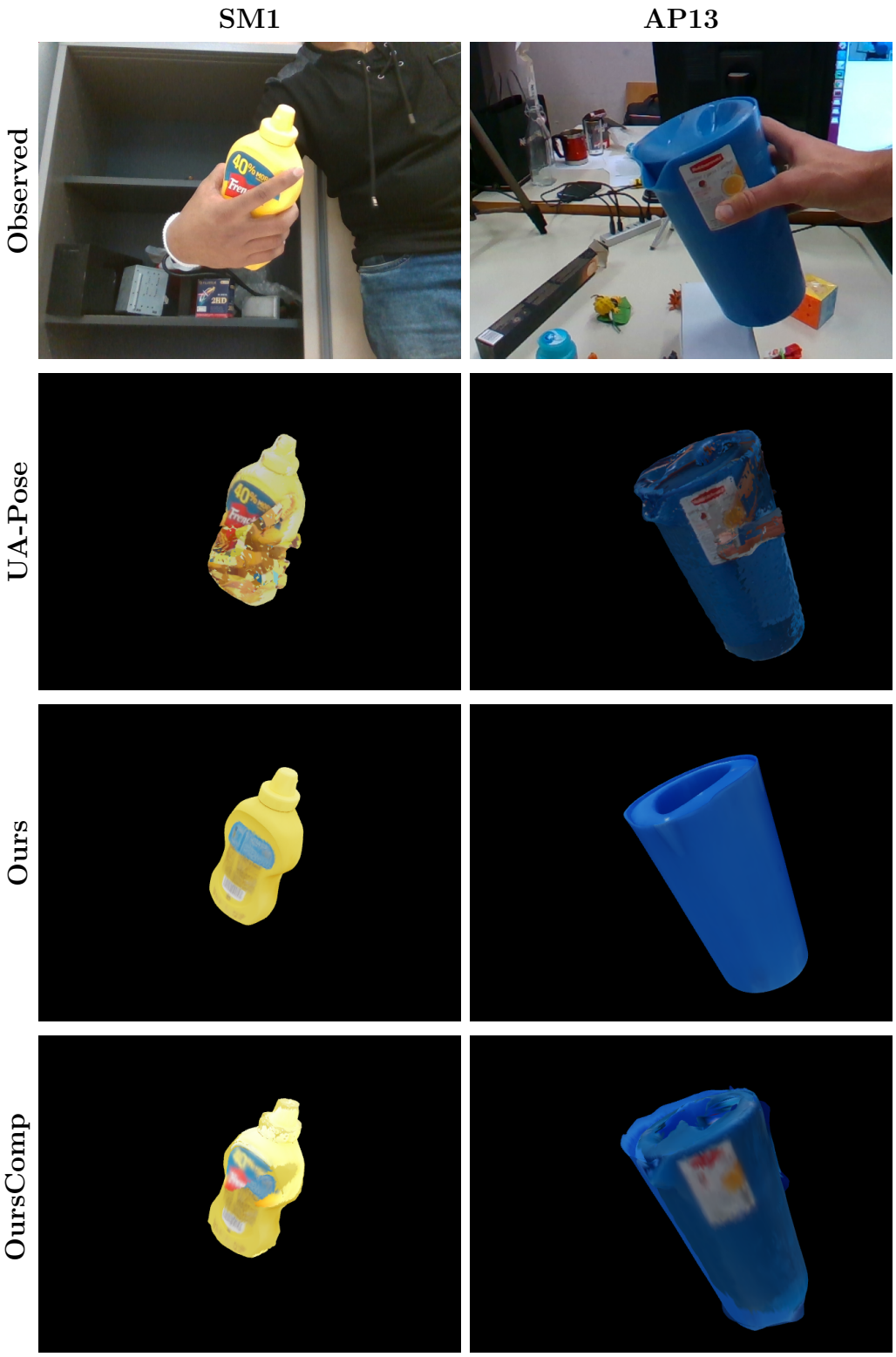

Figure 5.8: Mesh reconstruction failures of UA-Pose due to occlusion (SM1) and mask inaccuracies (AP13), compared to Ours and OursComp at the same frames.

5.3.10. Analysis

The experiments were designed to evaluate each pipeline stage independently while quantifying the overall gap between zero-shot model-free operation and model-based tracking. A fair comparison is enforced throughout: all methods share the same dataset, segmentation masks, and FoundationPose estimator, so performance differences isolate the contribution of mesh quality. The Baseline, running FoundationPose on the ground-truth CAD model, represents the upper bound achievable with a perfect mesh.

The core result is that SAM-3D combined with the proposed scale recovery achieves near-baseline pose performance with no object-specific prior, demonstrating that zero-shot model-free tracking is viable at minimal accuracy cost. The proposed scaling strategy proves well-suited to SAM-3D's coarse initial estimate, producing 3D IoU values that consistently exceed Any6D despite the latter having a more elaborate per-axis scale recovery and access to a reference image. The advantage traces directly to mesh generation quality: SAM-3D produces more geometrically accurate meshes than InstantMesh, and since FoundationPose operates by render-and-compare, texture and geometry fidelity directly affect pose quality.

Mesh completion yields a consistent improvement in mesh quality across all metrics, and a more modest but reliable gain in pose accuracy. The largest pose gains occur precisely on the sequences where the initial mesh is weakest, confirming that completion is most valuable when mesh generation alone is insufficient. If pose accuracy is the only concern and the initial mesh is already good, completion adds computational cost for marginal return. Its primary benefit is mesh reconstruction quality, most prominently in texture fidelity, where OursComp surpasses even the Baseline on most sequences by adapting to the actual scene lighting. UA-Pose's neural SDF likewise achieves competitive texture on observed regions, confirming that appearance quality is attainable for completion-based methods.

Any6D and UA-Pose share the same InstantMesh-generated mesh as starting point, yet diverge sharply in strategy. Any6D demonstrates that accurate scale recovery is the decisive step for image-to-3D methods: by applying per-axis scale correction, it reaches competitive pose and geometry results without any online completion. UA-Pose instead relies on a weaker scale initialization, which produces a poor starting mesh, though is able to recover from it through neural SDF completion. The reconstruction quality that results matches what is expected from BundleSDF [56], together with its known limitation: because the approach does not anchor reconstruction to a complete prior shape, occluded regions are progressively lost during tracking. UA-Pose's advantage over BundleSDF is

that it begins with a complete generated mesh, allowing convergence to a reasonable reconstruction when occlusion is not extreme. The overall behavior of UA-Pose suggests that neural SDF completion has potential, but it requires a more accurate mesh generation and scale estimation pipeline to be competitive.

5.4. Comparison with External Methods

6DOPE-GS[15] and BundleSDF[56] are zero-shot model-free online reconstruction methods that jointly estimate pose and geometry from scratch, without any generative prior, as described in Chapters 3 and 4. Both have been evaluated by the authors of 6DOPE-GS on the HO3D test set, with BundleSDF run from its open-source implementation, making a direct numerical comparison possible. The results are reported in Table 5.11. Unlike the metrics used in Section 5.3.2, ADD and ADD-S are here computed as AUC from 0 to 0.1 m (absolute threshold) rather than 0 to $0.1 \cdot d$ (diameter-relative), following the protocol adopted in their work; the values are therefore not directly comparable to those in Table 5.3.

OursComp outperforms both methods on all three metrics. The key structural difference is that OursComp initializes from a complete generative mesh on the first frame, whereas both competitors build geometry incrementally from observations only and therefore remain incomplete until sufficient viewpoints have been covered. The CD advantage confirms that explicit vertex-level completion consistently yields tighter geometry than either the neural SDF of BundleSDF or the Gaussian Object Field of 6DOPE-GS under the same evaluation conditions.

Table 5.11: Comparison on the HO3D test set against methods from the online reconstruction paradigm. ADD and ADD-S are AUC percentages with threshold range 0 to 0.1 m, following the protocol of [15]. CD is Chamfer Distance in cm.

Method	ADD-S ↑ (%)	ADD ↑ (%)	CD ↓ (cm)
BundleSDF [56]	94.86	89.56	0.58
6DOPE-GS [15]	95.07	84.33	0.41
OursComp	**95.28**	**89.92**	**0.35**

6 | Conclusions and Future Work

6.1. Conclusions

This thesis addressed the problem of 6D pose tracking and mesh completion of novel objects from a single RGB-D video, with no prior knowledge of object geometry or appearance. The proposed pipeline chains five stages: segmentation mask estimation and propagation, starting from a prompt of two points on the first frame [5, 17], single-view mesh generation with SAM-3D [4], two-round FoundationPose-based hypothesis scoring for scale recovery, FoundationPose tracking [57], and confidence-based geometric vertex update for mesh completion.

The central quantitative finding is that mesh generation quality is the dominant driver of performance across all metrics. SAM-3D, applied to the first test frame, consistently produces more geometrically accurate meshes than InstantMesh [60], even though Any6D [18] and UA-Pose [21] supply InstantMesh with a separately captured reference image. This quality difference propagates through the entire pipeline, since FoundationPose operates by render-and-compare: higher-fidelity geometry and texture directly translate into more accurate pose hypotheses.

The proposed scale recovery, which searches a single uniform scale factor over two consecutive FoundationPose scoring rounds, achieves a mean 3D IoU of 86.2%, outperforming Any6D (78.8%) and UA-Pose (53.6%). The result confirms that a uniform scale search is well-suited to SAM-3D meshes, whose proportions are more accurate than those of InstantMesh, making the more elaborate per-axis recovery of Any6D unnecessary. The two exceptions, AP11 and MPM12, correspond to sequences with a partially occluded or out-of-bound first frame, identifying initialization viewpoint quality as the primary remaining bottleneck for zero-shot scale recovery.

On pose tracking, the full proposed pipeline (SAM-3D mesh generation with confidence-based geometric completion) achieves 60.5 ADD / 82.0 ADD-S AUC, within 8.5 and 2.5 points of the model-based Baseline (68.9 / 84.5) that uses the ground-truth CAD model, while substantially outperforming Any6D (38.2 / 70.6) and UA-Pose (37.9 / 68.1).

The pipeline without the completion stage already exceeds Any6D by 18 ADD points, confirming that the SAM-3D mesh quality, rather than the completion stage, is responsible for the bulk of the performance gap on pose estimation.

The confidence-based geometric vertex update consistently improves both geometry and joint metrics: the full pipeline reaches a mean Chamfer Distance of 0.352 cm versus 0.400 cm without completion, and a mean ADI of 81.2 versus 76.7, closing to within 4.2 points of the Baseline (85.4). The largest gains occur precisely on the sequences where the initial SAM-3D mesh is weakest, confirming that completion is most valuable when mesh generation alone is insufficient. On texture fidelity, the full pipeline achieves a mean PSNR of 18.6 dB after completion, surpassing the Baseline (15.3 dB) on 12 of 13 sequences, because vertex colors are learned directly from the video under the actual scene lighting rather than from a fixed CAD model texture.

Two structural limitations remain. First, while the completion stage reliably refines the geometry of observed regions, it cannot synthesize entirely new parts of the object if the SAM-3D initial mesh is fundamentally incomplete: the confidence-based vertex update corrects and improves existing geometry but relies on the prior mesh as general structure, so a severely degraded first frame propagates an irrecoverable deficit through the pipeline. Adding entirely new surface regions would require regenerating the mesh from the accumulated point cloud, similarly to what HIPPo [28] does via Poisson surface reconstruction [16], at the cost of a more expensive and less frequent update. Second, neither the proposed completion stage nor any competing method currently runs at real-time frame rates, which prevents continuous online refinement during fast manipulation. Nevertheless, the geometric approach remains near-practical: although it is triggered more frequently than the neural SDF retraining of UA-Pose, each individual update completes considerably faster, resulting in a mean total completion time of only 3.5 min per video compared to over one hour for UA-Pose.

These constraints suggest a natural deployment scenario: the pipeline is run offline on an initial segment of the video to produce an accurate, scene-adapted mesh, after which standard pose tracking can be applied with that mesh as the object model. This is in fact a simpler and more scalable workflow than the classical model-based paradigm, which requires capturing a dedicated reference sequence with pose annotations and processing it through a separate reconstruction pipeline before tracking can begin. Within this framing, the results demonstrate that zero-shot model-free 6D pose tracking is viable at minimal accuracy cost relative to model-based tracking, without any object-specific prior.

6.2. Future Work

Several directions could extend the scope and robustness of the proposed pipeline.

Near-term quality improvements. The proposed geometric completion already delivers competitive geometry and texture on observed regions, yet targeted refinements remain viable without architectural changes. Tightening the confidence criteria for triggering updates, improving texture blending consistency at completion boundaries, and better handling of heavily occluded regions are the most promising directions. Since the geometric approach already strikes a favorable accuracy–efficiency trade-off, these improvements could yield meaningful gains while preserving online tracking capability.

Towards high-fidelity mesh reconstruction. Applications such as augmented reality demand complete, high-fidelity meshes that classical reconstruction pipelines may not always be able to provide. A natural path toward higher fidelity is to replace the geometric vertex update with a neural SDF anchored to the generative prior on unseen parts, preserving completeness under occlusion while recovering finer surface detail on observed regions. However, as the runtime analysis shows, retraining a neural SDF per update is computationally prohibitive for online use — UA-Pose already accumulates over one hour of completion time on longer sequences. This direction therefore sacrifices online tracking capability, but would constitute an effective zero-shot pipeline for high-quality mesh acquisition from a simple RGB-D camera in offline settings.

Towards real-time zero-shot pose tracking. When mesh reconstruction quality is not the primary objective and low latency is required, the results show that the pipeline without completion already achieves near model-based pose accuracy, making it a viable drop-in replacement for workflows that previously required a dedicated capture session and reconstruction pipeline. To close the remaining accuracy gap in failure cases, a promising direction would be to fine-tune the image-to-3D generation model on video observations: the pipeline would start as now, but monitor pose quality online and, when hallucinated or inconsistent geometry is detected, incorporate the new posed RGB-D observations directly into the generation model to refine only the affected regions, without rerunning the full generation from scratch.

Adaptive completion parameters. The completion module relies on fixed parameters tuned for HO3D's specific resolution, frame rate, and object scale. Replacing these with thresholds and update frequencies inferred from per-frame quality signals would improve

robustness across datasets and sensor configurations without manual retuning.

Broader evaluation. The current evaluation is limited to HO3D, a single dataset in hand-object interaction scenarios. Evaluating on additional benchmarks such as YCB-Video [58] and YCBInEOAT [54] would assess generalization across object categories, lighting conditions, and interaction types. It would also enable a more comprehensive comparison with 6DOPE-GS [15] and HIPPo [28], the latter of which is currently compared only at the algorithmic level due to its unavailable public implementation.

Multi-object scenes. The current pipeline tracks a single object. Extending it to simultaneous multi-object tracking and reconstruction is a natural next step, with all pipeline stages generalizing straightforwardly at the cost of increased computation per frame.

Bibliography

[1] E. Arnold, O. Y. Al-Jarrah, M. Dianati, S. Fallah, D. Oxtoby, and A. Mouzakitis. A survey on 3d object detection methods for autonomous driving applications. *IEEE Transactions on Intelligent Transportation Systems*, 20(10):3782–3795, 2019.

[2] P. J. Besl and N. 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.

[3] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother. Learning 6d object pose estimation using 3d object coordinates. In *European conference on computer vision*, pages 536–551. Springer, 2014.

[4] X. Chen, F.-J. Chu, P. Gleize, K. J. Liang, A. Sax, H. Tang, W. Wang, M. Guo, T. Hardin, X. Li, et al. Sam 3d: 3dfy anything in images. *arXiv preprint arXiv:2511.16624*, 2025.

[5] H. K. Cheng and A. G. Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *European conference on computer vision*, pages 640–658. Springer, 2022.

[6] B. Drost, M. Ulrich, N. Navab, and S. Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 998–1005. IEEE, 2010.

[7] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981. ISSN 0001-0782. doi: 10.1145/358669.358692. URL `https://doi.org/10.1145/358669.358692`.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 580–587, 2014.

[9] S. Hampali, M. Rad, M. Oberweger, and V. Lepetit. Honnotate: A method for 3d

annotation of hand and object poses. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3196–3206, 2020.

[10] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2961–2969, 2017.

[11] X. He, J. Sun, Y. Wang, D. Huang, H. Bao, and X. Zhou. Onepose++: Keypoint-free one-shot object pose estimation without CAD models. In *Advances in Neural Information Processing Systems*, 2022.

[12] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *Asian conference on computer vision*, pages 548–562. Springer, 2012.

[13] T. Hodan, F. Michel, E. Brachmann, W. Kehl, A. GlentBuch, D. Kraft, B. Drost, J. Vidal, S. Ihrke, X. Zabulis, et al. Bop: Benchmark for 6d object pose estimation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 19–34, 2018.

[14] B. Huang, Z. Yu, A. Chen, A. Geiger, and S. Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 conference papers*, pages 1–11, 2024.

[15] Y. Jin, V. Prasad, S. Jauhri, M. Franzius, and G. Chalvatzaki. 6dope-gs: Online 6d object pose estimation using gaussian splatting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8032–8043, 2025.

[16] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, 2006.

[17] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick. Segment anything. *arXiv:2304.02643*, 2023.

[18] T. Lee, B. Wen, M. Kang, G. Kang, I. S. Kweon, and K.-J. Yoon. Any6d: Model-free 6d pose estimation of novel objects. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 11633–11643, 2025.

[19] V. Lepetit, F. Moreno-Noguer, and P. Fua. Epnp: Efficient perspective-n-point camera pose estimation. *Int. J. Comput. Vis*, 81(2):155–166, 2009.

[20] V. Leroy, Y. Cabon, and J. Revaud. Grounding image matching in 3d with mast3r. In *European Conference on Computer Vision*, pages 71–91. Springer, 2024.

[21] M.-F. Li, X. Yang, F.-E. Wang, H. Basak, Y. Sun, S. Gayaka, M. Sun, and C.-H. Kuo. Ua-pose: Uncertainty-aware 6d object pose estimation and online object completion with partial references. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1180–1189, 2025.

[22] 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.

[23] J. Liu, W. Sun, C. Liu, X. Zhang, and Q. Fu. Robotic continuous grasping system by shape transformer-guided multiobject category-level 6-d pose estimation. *IEEE Transactions on Industrial Informatics*, 19(11):11171–11181, 2023.

[24] R. Liu, R. Wu, B. Van Hoorick, P. Tokmakov, S. Zakharov, and C. Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9298–9309, 2023.

[25] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. 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.

[26] X. Liu, G. Wang, R. Zhang, C. Zhang, F. Tombari, and X. Ji. UNOPose: Unseen object pose estimation with an unposed rgb-d reference image. In *CVPR*, June 2025.

[27] Y. Liu, Y. Wen, S. Peng, C. Lin, X. Long, T. Komura, and W. Wang. Gen6d: Generalizable model-free 6-dof object pose estimation from rgb images. In *European Conference on Computer Vision*, pages 298–315. Springer, 2022.

[28] Y. Liu, Z. Jiang, B. Xu, G. Wu, Y. Ren, T. Cao, B. Liu, R. H. Yang, A. Rasouli, and J. Shan. Hippo: Harnessing image-to-3d priors for model-free zero-shot 6d pose estimation. *IEEE Robotics and Automation Letters*, 2025.

[29] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[30] X. Long, Y.-C. Guo, C. Lin, Y. Liu, Z. Dou, L. Liu, Y. Ma, S.-H. Zhang, M. Habermann, C. Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9970–9980, 2024.

[31] M. Macklin. Warp: A high-performance python framework for gpu simulation and graphics. In *NVIDIA GPU Technology Conference (GTC)*, volume 3, 2022.

[32] T. Mallick, P. P. Das, and A. K. Majumdar. Characterizations of noise in kinect depth images: A review. *IEEE Sensors journal*, 14(6):1731–1740, 2014.

[33] E. Marchand, H. Uchiyama, and F. Spindler. Pose estimation for augmented reality: a hands-on survey. *IEEE transactions on visualization and computer graphics*, 22 (12):2633–2651, 2015.

[34] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[35] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. Orb-slam: A versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.

[36] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE international symposium on mixed and augmented reality*, pages 127–136. Ieee, 2011.

[37] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[38] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.

[39] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[40] M. Runz, M. Buffier, and L. Agapito. Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects. In *2018 IEEE international symposium on mixed and augmented reality (ISMAR)*, pages 10–20. IEEE, 2018.

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

[42] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 1, pages 519–528. IEEE, 2006.

[43] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[44] R. Shi, H. Chen, Z. Zhang, M. Liu, C. Xu, X. Wei, L. Chen, C. Zeng, and H. Su. Zero123++: a single image to consistent multi-view diffusion base model. *arXiv preprint arXiv:2310.15110*, 2023.

[45] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8922–8931, 2021.

[46] J. Sun, Z. Wang, S. Zhang, X. He, H. Zhao, G. Zhang, and X. Zhou. Onepose: One-shot object pose estimation without cad models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6825–6834, 2022.

[47] G. Taubin. A signal processing approach to fair surface design. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 351–358, 1995.

[48] G. Turk and M. Levoy. Zippered polygon meshes from range images. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 311–318, 1994.

[49] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3343–3352, 2019.

[50] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L. J. Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2642–2651, 2019.

[51] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021.

[52] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[53] B. Wen and K. Bekris. Bundletrack: 6d pose tracking for novel objects without instance or category-level 3d models. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8067–8074. IEEE, 2021.

[54] B. Wen, C. Mitash, B. Ren, and K. E. Bekris. se (3)-tracknet: Data-driven 6d pose tracking by calibrating image residuals in synthetic domains. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10367–10373. IEEE, 2020.

[55] B. Wen, W. Lian, K. Bekris, and S. Schaal. You only demonstrate once: Category-level manipulation from single visual demonstration. *arXiv preprint arXiv:2201.12716*, 2022.

[56] B. Wen, J. Tremblay, V. Blukis, S. Tyree, T. Müller, A. Evans, D. Fox, J. Kautz, and S. Birchfield. Bundlesdf: Neural 6-dof tracking and 3d reconstruction of unknown objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 606–617, 2023.

[57] B. Wen, W. Yang, J. Kautz, and S. Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17868–17879, June 2024.

[58] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox. Posecnn: A convolutional neural network for 6d object pose estimation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5738–5745. IEEE, 2018.

[59] B. Xu, W. Li, D. Tzoumanikas, M. Bloesch, A. Davison, and S. Leutenegger. Midfusion: Octree-based object-level multi-instance dynamic slam. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5231–5237. IEEE, 2019.

[60] J. Xu, W. Cheng, Y. Gao, X. Wang, S. Gao, and Y. Shan. Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191*, 2024.

[61] J. Yang, M. Gao, Z. Li, S. Gao, F. Wang, and F. Zheng. Track anything: Segment anything meets videos. *arXiv preprint arXiv:2304.11968*, 2023.

[62] L. Yariv, J. Gu, Y. Kasten, and Y. Lipman. Volume rendering of neural implicit surfaces. *Advances in neural information processing systems*, 34:4805–4815, 2021.

[63] Q.-Y. Zhou, J. Park, and V. Koltun. Open3d: A modern library for 3d data processing. *arXiv preprint arXiv:1801.09847*, 2018.

List of Figures

List of Tables