UA-Pose: Uncertainty-Aware 6D Object Pose Estimation and Online Object Completion with Partial References

Supplementary Materials

1. Implementation Details

The object images and uncertainty maps for pose estimation are rendered using the graphics pipeline [2] and Kornia [7], and mesh rasterization for uncertainty modeling is performed with PyTorch3D [1]. For the pose refinement and selection modules described in Sec. 3.3 of the main paper, we utilize the publicly available checkpoints from [10]. Note that this version was not trained on diffusion-augmented data, which may lead to performance degradation compared to the expected results reported in [10]. In this work, the neural Signed Distance Field (SDF) is trained using a simplified version of multi-resolution hash encoding [4], leveraging the CUDA implementation from [9]. The encoding is structured with 4 levels, each having feature vectors ranging from 16 to 128 in size and a feature dimension of 2. The hash table size is set to 2^{22}. Each training iteration processes a ray batch of 2048, with a truncation distance λ of 1 cm. Every SDF is trained for 500 steps, which completes within seconds. The geometry network Ω is a two-layer MLP with a hidden size of 64 and ReLU activation for all layers except the last. It outputs a geometric feature $f_{\Omega(\cdot)}$ with a size of 16. The appearance network Φ is a three-layer MLP, also with a hidden size of 64. ReLU activations are used for all the layers except the last layer, where a sigmoid function maps the predicted colors to the range $[0, 1]$.

1.1. Computation Time

We evaluate the computation time of each module on a single NVIDIA RTX 3090 GPU. The full pose estimation procedure, which includes pose initialization, pose refinement, and pose selection, is completed within 2 seconds. Rendering uncertainty maps is highly efficient, requiring less than 0.01 seconds. In practice, assuming sequential input test frames, the full pose estimation procedure is applied only to the first frame of the test image sequence. For subsequent frames, the estimated pose from the previous frame serves as the sole pose hypothesis, and only pose refinement is performed to update the pose for the new frame. This refinement process takes less than 0.02 seconds per frame, ensuring real-time performance. When object completion is required, the full SDF training and Marching Cubes process is executed, taking approximately 30 seconds to generate the updated 3D model.

1.2. Hyperparameters

The hyperparameters and thresholds used in the main paper are set as follows. For pose initialization in both our method and the baseline implementation [10], the sampled viewpoints (N_v) and in-plane rotations (N_{in}) are set to 42 and 12, respectively, yielding a total of 504 pose hypotheses during the hypothesis generation step. The number of pose refinement iterations is set to 5 for the first frame and 2 for subsequent frames. In the pose selection process, a pose hypothesis is discarded if its *uncertainty rate* exceeds the threshold T_u or if its *seen IoU* falls below the threshold T_s, both set to 0.5. For storing frames and estimated poses for object completion, a frame is included in the memory pool $\mathcal{P}$ (with a maximum of 30 frames) only if its geodesic distance exceeds the threshold T_{geo}, set to 10 degrees. This ensures that each frame provides a unique viewpoint to enhance model completeness while keeping $\mathcal{P}$ compact. Additionally, a new frame can be appended to the memory pool only if its *seen IoU* exceeds the confidence threshold T_{conf}, set to 0.5. Object completion is triggered when the *seen IoU* falls below the threshold $T_{complete}$, set to 0.7. This ensures that object completion is performed only when the model lacks sufficient completeness. For SDF training, the training weights are set as follows: $w_c = 100$ for the color loss, $w_e = 1$ for the empty space loss, and $w_s = 1000$ for the surface loss.

1.3. Segmentation Methods

In this work, the object mask for each test image is required to calculate *seen IoU* and identify the region of the object for object completion. To obtain per-frame 2D object masks in test images, given the segmentation mask m_0 of the object of interest in the first-frame image I_0, the object masks of the following frames $\{m_1, m_2, \ldots, m_{k-1}\}$ are determined by off-the-shelf segmentation methods [13].

2. Datasets and Metrics

2.1. Preparing Partial References

Two RGBD references. In our work, all experiments and baselines are conducted under the *model-free* setting, where external RGBD images, along with their corresponding camera poses, are used as references for pose estimation. In [10], 16 reference images per object are sampled from the YCB-Video training set [11] to form the subset **S**, ensuring sufficient observations from diverse viewpoints. To ensure that the first frame of the test images is covered by the selected references for initial pose estimation, we manually select 2 reference images per object from **S** for the test sequences.

Single unposed RGB reference. In scenarios utilizing single-image-to-3D methods [12] for pose estimation, we use a single unposed RGB image as input to generate an initial 3D object model. Specifically, we manually select one RGB image per object from the reference set **S** for each test sequence. This selected image serves as the sole external reference for generating the 3D model, without any additional depth or pose information.

Subset for the YCB-Video Dataset. In this work, we leverage the pose refinement and selection models from [10] to estimate object poses using 3D object models generated by image-to-3D approaches [4]. However, we observed that current image-to-3D methods have difficulty generating accurate object models for certain objects. These poorly generated models result in failed pose estimations, introducing outliers into the average metrics. To ensure a fair evaluation, for the YCB-Video dataset [11], we focus on a subset of the evaluation set for the "single RGB + Image-to-3D" experiments, including commonly seen objects "004_sugar_box," "005_tomato_soup_can," "006_mustard_bottle," and "019_pitcher_base." This selection ensures consistent and stable evaluation results across all baselines. For further discussion on the limitations of current image-to-3D methods and pose estimation models, please refer to Sec. 6.

3. Leveraging Single-Image-to-3D Methods

Generate an object 3D mesh. To generate an initial 3D model of the object, we utilize InstantMesh [12], a single-image-to-3D method that generates a coarse 3D mesh from a single RGB image. InstantMesh combines learned geometric priors with image features to infer the object's shape and produce a 3D mesh representation based solely on the input RGB image. Given a single unposed RGB reference, we employ InstantMesh to generate an initial 3D object mesh, denoted as $\hat{\mathcal{E}}_g^i = (V_g^i, C_g^i, F_g^i)$ where the vertices V_g^i, vertex colors C_g^i, and faces F_g^i.

Uncertainty map for the generated mesh. An uncertainty map, $\hat{\mathcal{U}}_g^i$, is integrated into the mesh representation

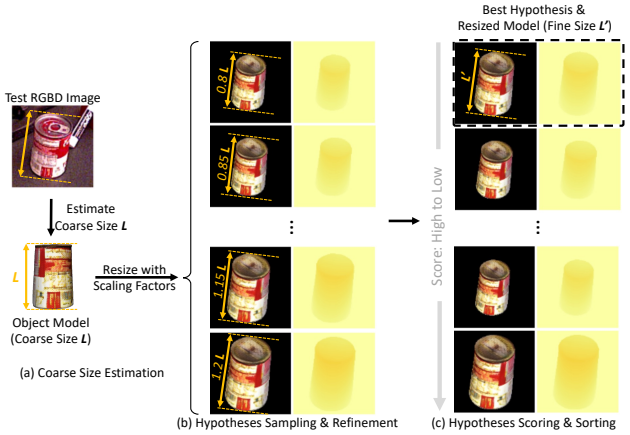

Figure 1. **Pipeline for rescaling the generated model.** The generated object model must be rescaled to match the actual object size in the test images for accurate pose estimation. (a) In the first stage, the coarse size of the model is estimated using the first frame of the testing RGBD images. The maximum distance, L, between the two farthest 3D points is computed to rescale the model to a rough size. (b) In the second stage, multiple scaling factors are uniformly sampled to slightly adjust the model size further, resulting in several resized models. For each resized model, pose hypotheses are generated and refined using the pose refinement model. (c) Finally, the selection strategy is applied to score the hypotheses and identify the optimal pose hypothesis and the fine size of the model.

$\hat{\mathcal{E}}_g^i$ to create a hybrid representation, $\hat{\mathcal{M}}_g^i = (\hat{\mathcal{E}}_g^i, \hat{\mathcal{U}}_g^i)$, for uncertainty-aware pose estimation. The uncertainty map $\hat{\mathcal{U}}_g^i$ is generated by marking the viewpoint corresponding to the initial RGB image as "certain" and labeling areas inferred by the image-to-3D method as "uncertain." This process involves projecting $\hat{\mathcal{E}}_g^i$ onto the viewpoint of the reference image using mesh rasterization, which maps 3D vertices onto 2D image pixels. For each vertex, an uncertainty score $u(v_i)$ is calculated based on its visibility in the reference image. The uncertainty score $u(v_i) \in \{0, 1\}$ is defined as follows: $u(v_i) = 0$ if the vertex v_i is visible in the reference image, and $u(v_i) = 1$ if the vertex is not visible. The resulting uncertainty map, $\hat{\mathcal{U}}_g^i = \{u(v_i) \mid v_i \in V_g^i\}$ is incorporated with $\hat{\mathcal{E}}_g^i$ to form the generated model $\hat{\mathcal{M}}_g^i = (\hat{\mathcal{E}}_g^i, \hat{\mathcal{U}}_g^i)$.

Rescale the generated model. The generated model $\hat{\mathcal{M}}_g^i$ requires rescaling to match the object's actual size in the test images for accurate pose estimation. To adjust the model to a size closer to the real object, we propose a two-stage coarse-to-fine process, as shown in the Fig. 1. In the first stage, the coarse size is estimated using the depth map from the first frame of the testing RGBD images. Specifically, given the depth map and the object mask, 2D points within the object mask are projected into 3D space using the corresponding depth values. The maximum distance, L, between the two farthest 3D points is then calculated. The generated model (represented as a mesh) is scaled such that its

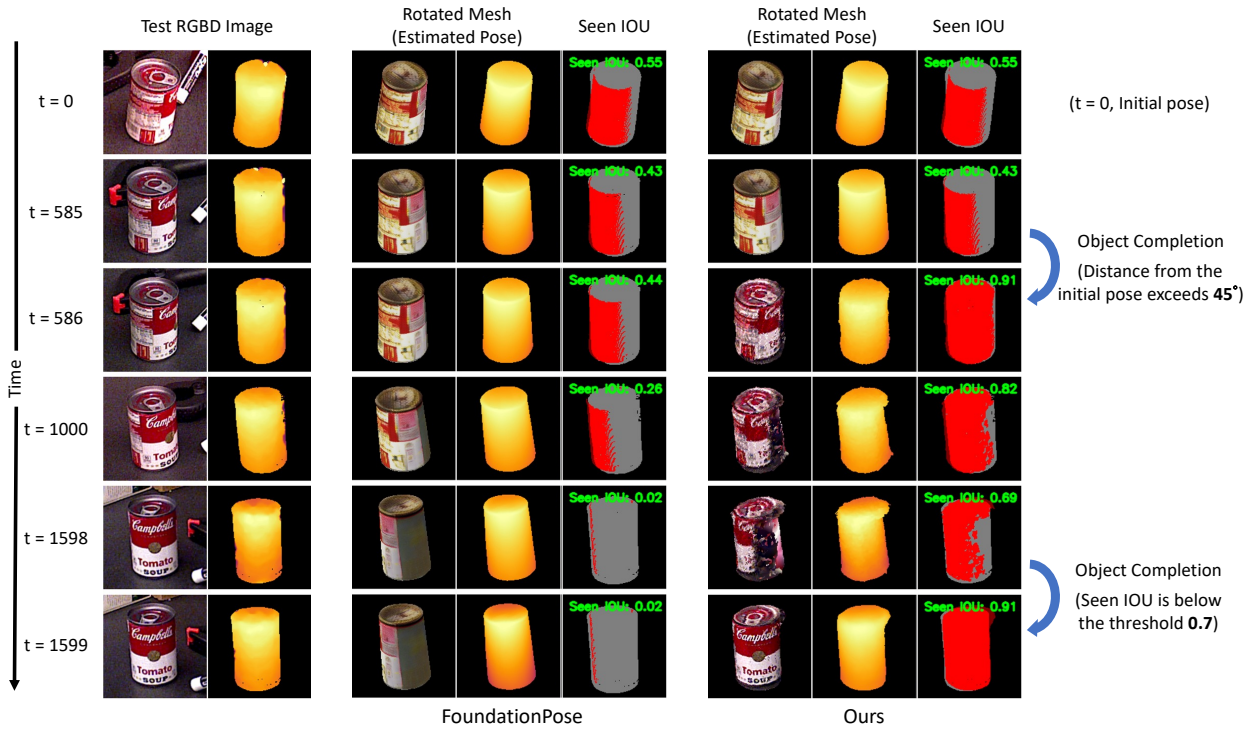

Figure 2. **Qualitative results on the YCB-Video dataset.** We compare our method with FoundationPose [10] using an object 3D model generated from a single RGB image by the image-to-3D approach [12]. The columns, from left to right, display the test RGBD images (left: RGB, right: depth), results from FoundationPose, and results from our method. The object 3D models are visualized by rotating them based on the estimated poses. Additionally, the uncertainty map (red: seen region of the object 3D model; gray: object mask of the test image) and the *Seen IoU* metric (indicating the overlap between the seen region of the object 3D model and the object mask) are shown. In the final column, we demonstrate our iterative process of pose estimation and online object completion in our method, highlighting how an initial generated object 3D model is refined into a more complete and accurate object 3D model that better represents the real object's appearance and geometry. In contrast to directly using FoundationPose with the initial generated object 3D model (second column), which may not closely resemble the real object captured in the test images, our method maintains the completeness and correctness of the object 3D model while enhancing pose estimation accuracy.

diameter, which is defined as the distance between its two farthest vertices, matches this coarse object length L, resulting in the rescaled model $\hat{\mathcal{M}}_g^c$. In the second stage, we refine the scaling to determine the fine object length, L'. Multiple scaling factors, $\mathcal{S} = \{0.8, ..., 1.2\}$ ($|\mathcal{S}| = 11$), are uniformly sampled and multiplied by the coarse length L to produce $|\mathcal{S}|$ resized models. For each resized model, $N'_v \cdot N'_{in}$ pose hypotheses are generated, where $N'_v = 5$ represents the number of sampled viewpoints, and $N'_{in} = 24$ denotes the number of sampled in-plane rotations. This process results in a total of $|\mathcal{S}| \cdot N'_v \cdot N'_{in}$ pose hypotheses. We then apply the pose refinement and selection strategy described in Sec. 3.3 of the main paper to identify the optimal pose hypothesis. The corresponding resized model, $\hat{\mathcal{M}}_g^f$, is scaled to the fine object length $L' = s \cdot L$, where $s \in \mathcal{S}$. This refinement process is repeated iteratively three times, resulting in the final model $\hat{\mathcal{M}}' = (\hat{\mathcal{E}}', \hat{\mathcal{U}}')$ used for pose estimation.

Generate augmented data for object completion. To further leverage the generated 3D model, we render RGBD images from the image-to-3D generated mesh $\hat{\mathcal{E}}'$ as augmented data for SDF training. This supervision helps enhance the object geometry in unseen regions, providing more complete information for pose estimation. Specifically, we uniformly sample 24 viewpoints from an icosphere centered on the generated object mesh $\hat{\mathcal{E}}'$, rendering synthesized object images $\hat{\mathcal{I}} = \{\hat{I}_0, \hat{I}_1, \ldots, \hat{I}_{23}\}$, along with their associated poses and 2D object masks, as augmented data for object completion. During testing, the initial generated model $\hat{\mathcal{M}}'$ is used to estimate poses for the first few test frames. When the rotational geodesic distance between the initial pose and a newly estimated pose exceeds a threshold T_{gen}(set to 45 degrees), a "refined object model" is trained to replace the initial generated model, resulting in a more complete and accurate representation. During this refinement, synthesized object images in $\hat{\mathcal{I}}$ are used to enhance the geometry in unseen regions of the object. However, synthesized images may negatively impact SDF training if the corresponding regions have already been accurately captured by real images. Such overlap can introduce inconsistent supervision

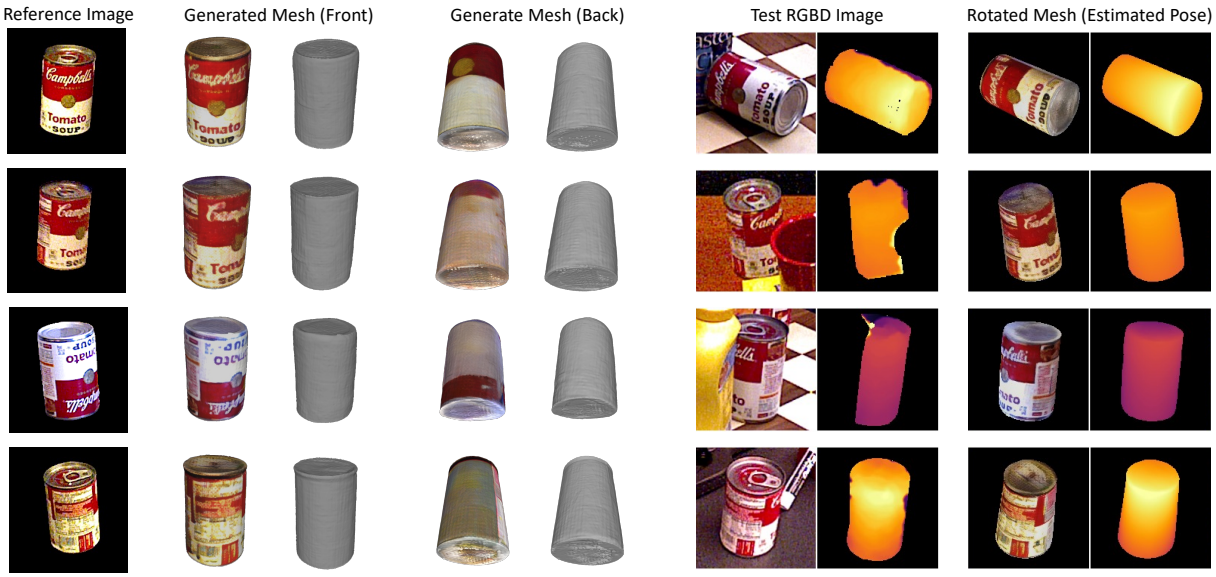

Figure 3. **Leveraging image-to-3D approaches for pose estimation across diverse reference images and test sequences.** We demonstrates the diversity and effectiveness of our method in utilizing the image-to-3D approach [12] to generate object 3D models for various test sequences. Each row represents a test sequence and includes the reference image, the generated object 3D model (front and back views), the test RGBD image, and the rotated mesh based on the estimated pose. The results highlight our method's ability to successfully perform pose estimation across different test sequences, showcasing the potential of leveraging image-to-3D approaches for pose estimation using various single RGB images.

and degrade the final SDF quality. To address this, we employ an uncertainty map $\mathcal{U}$ to filter out unnecessary synthesized images. Before each SDF training iteration, we render the uncertainty map for the "refined object model" using the pose of each synthesized object image $\hat{I}_i$ and compare the visible region with the 2D object mask of $\hat{I}_i$. If more than 30% of the pixels in $\hat{I}_i$'s 2D mask overlap with regions already marked as "seen" in the "refined object model", the synthesized image $\hat{I}_i$ is removed from $\hat{\mathcal{I}}$. By filtering out redundant or conflicting synthesized data, this process ensures that synthesized images are used exclusively to supervise unseen regions.

Difference with Gigapose. The prior work, GigaPose [5], investigated using object models generated by the single-image-to-3D method [3] for pose estimation. In GigaPose, a 3D object model is created from the first frame of the "RGBD" test images, with the depth map used to accurately scale the model. This generated 3D model is then directly applied for pose estimation across the entire test sequence. However, as discussed in Sec. 4 of the main paper, we demonstrate that such generated models often fail to represent real objects accurately, especially when objects rotate into unseen regions not covered by the initial partial references. Moreover, if the object is occluded in the first frame of the test image, GigaPose may struggle to generate a reasonably complete 3D model based on the cropped object image. In contrast, we focus on a more challenging scenario where only a single external "RGB" reference image is provided, without any depth information. "RGB" images are more applicable to everyday use cases, such as using internet-sourced images as references. Moreover, these external reference images can carry valuable meta-information for robotic tasks, such as grasping patterns and detection markers, making them highly practical for real-world applications.

4. Qualitative Results

4.1. Pose Estimation and Online Shape Completion

In Fig. 2, we demonstrate the effectiveness of our method for pose estimation and online shape completion in the scenario where a single image is used as input and we leverage the image-to-3D approach [12] to generate the initial object 3D model for pose estimation. The experimental results show that directly using the generated model with FoundationPose [10] often fails to produce optimal results, as these models may not accurately represent real objects. In contrast, UA-Pose utilizes the generated model solely for initial pose estimation and for rendering RGBD images as augmented data to support object completion, as detailed in Sec. 3. This iterative approach reconstructs an object 3D model that is more closely aligned with the real object.

4.2. Diverse generated objects

We showcases the ability of our method to utilize image-to-3D approaches for pose estimation with diverse refer-

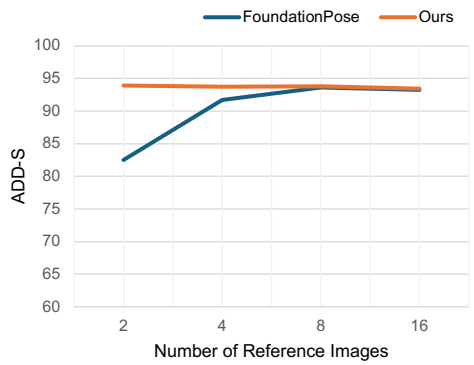

Figure 4. **Effects of the number of reference images.** The ADD-S scores in the YCBInEOAT dataset with 2, 4, 8, and 16 reference images are reported. Our method demonstrates stable performance across different numbers of reference images, while FoundationPose [6] shows significant performance drops when fewer reference images (e.g., 2 and 4 views) are used.

ence images across different test sequences. As illustrated in Fig. 3, each row corresponds to a unique test sequence. By leveraging image-to-3D techniques, our method achieves successful pose estimation with object 3D models generated based on single RGB images. This demonstrates the potential of leveraging image-to-3D approaches in enhancing pose estimation for real-world applications.

5. Additional Analysis

Effects of the viewpoints of input reference images. We conduct an experiment on the YCBInEOAT dataset [8] to evaluate the effect of varying the viewpoints of input reference images, specifically in the scenario with two input reference images. For each test sequence, we select three distinct pairs of reference images captured from different viewpoints. The results show that FoundationPose [10] achieves average ADD-S, ADD, and CD scores of 75.61, 65.37, and 0.905 cm, respectively. In comparison, our method achieves 93.47, 85.26, and 0.691 cm, with an average of approximately 6 SDF reconstructions per test sequence. These results demonstrate that our approach is robust to variations in reference viewpoints and outperforms the baseline.

Effects of the number of reference images. To examine the effect of the number of reference views, we plot the ADD-S scores on the YCBInEOAT dataset using different numbers of reference images in Fig. 4. The ADD-S scores show that our method maintains consistent performance across different numbers of reference images, while FoundationPose [6] experiences significant performance drops when fewer reference images (e.g., 2 and 4 views) are used. Note that this experiment excludes the test sequence "tomato_soup_can" due to occasional pose estimation failures, as discussed in Sec. 6, which introduce outliers for comparison.

Figure 5. **Incorrectly selected hypothesis.** Given a test RGBD image (left: RGB, right: depth), the pose selection module [10] scores all pose hypotheses and selects the one with the highest score as the pose estimation result. However, the selection module may occasionally assign a higher score to an obviously incorrect pose hypothesis (selected hypothesis, score: 138.906) while assigning a lower score to a more reasonable estimation (unselected hypothesis, score: 138.781). Such incorrect selections result in unstable pose estimation.

Figure 6. **Generated models from cropped object images.** We illustrate the limitations of single-image-to-3D approaches when relying on cropped or occluded reference images. The left column shows the RGB reference images and the right column displays the corresponding generated object 3D models. The results highlight that when the reference image is cropped or the object is partially occluded, the resulting 3D model is incomplete and fails to accurately represent the full object geometry.

6. Limitations

Foundation models for pose estimation. As shown in Fig. 5, the pose selection module of [10] may occasionally select an obviously incorrect pose hypothesis as the best. This issue likely arises because the pose selection model [10] was primarily trained on well-reconstructed object models that closely resemble real objects, rather than on incomplete, poorly reconstructed, or generated 3D models. To improve reliability, finetuning the pose selection model on synthesized data derived from incomplete object models could enhance its robustness and accuracy, reducing the occurrence of selecting clearly incorrect pose hypotheses.

Image-to-3D methods. Image-to-3D approaches generally rely on high-quality object reference images. When the reference image is cropped or the object is partially oc-

cluded, the generated object 3D model is often incomplete, as shown in Fig. 6. Additionally, as highlighted in the experiments of the main paper, using a single unposed RGB image that captures an object from only a specific viewpoint often results in object 3D models that fail to accurately resemble real objects, particularly in their uncaptured regions. Future work could address these limitations by integrating our pipeline with multi-view image-to-3D approaches to capture more comprehensive object information from different viewpoints. Alternatively, exploring image-to-3D methods that utilize RGBD inputs could better leverage depth information to infer more accurate and detailed object 3D geometry.

References

[1] Justin Johnson, Nikhila Ravi, Jeremy Reizenstein, David Novotny, Shubham Tulsiani, Christoph Lassner, and Steve Branson. Accelerating 3d deep learning with pytorch3d. In *SIGGRAPH*. 2020. 1

[2] Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. *TOG*, 2020. 1

[3] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, and Wenping Wang. Wonder3d: Single image to 3d using cross-domain diffusion. *CVPR*, 2024. 4

[4] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *TOG*, 2022. 1, 2

[5] Van Nguyen Nguyen, Thibault Groueix, Mathieu Salzmann, and Vincent Lepetit. Gigapose: Fast and robust novel object pose estimation via one correspondence. *CVPR*, 2024. 4

[6] Evin Pınar Örnek, Yann Labbé, Bugra Tekin, Lingni Ma, Cem Keskin, Christian Forster, and Tomas Hodan. Foundpose: Unseen object pose estimation with foundation features. *ECCV*, 2024. 5

[7] Edgar Riba, Dmytro Mishkin, Daniel Ponsa, Ethan Rublee, and Gary Bradski. Kornia: an open source differentiable computer vision library for pytorch. In *WACV*, 2020. 1

[8] Bowen Wen, Chaitanya Mitash, Baozhang Ren, and Kostas E Bekris. se (3)-tracknet: Data-driven 6d pose tracking by calibrating image residuals in synthetic domains. In *IROS*, 2020. 5

[9] Bowen Wen, Jonathan Tremblay, Valts Blukis, Stephen Tyree, Thomas Müller, Alex Evans, Dieter Fox, Jan Kautz, and Stan Birchfield. Bundlesdf: Neural 6-dof tracking and 3d reconstruction of unknown objects. In *CVPR*, 2023. 1

[10] Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. *CVPR*, 2024. 1, 2, 3, 4, 5

[11] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *RSS*, 2018. 2

[12] Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191*, 2024. 2, 3, 4

[13] Jinyu Yang, Mingqi Gao, Zhe Li, Shang Gao, Fangjing Wang, and Feng Zheng. Track anything: Segment anything meets videos. *arXiv preprint arXiv:2304.11968*, 2023. 1