GigaPose: Fast and Robust Novel Object Pose Estimation via One Correspondence

Van Nguyen Nguyen[1]   Thibault Groueix[2]   Mathieu Salzmann[3]   Vincent Lepetit[1]

[1]LIGM, École des Ponts   [2]Adobe   [3]EPFL

| Ground-truth & Input segmentation | MegaPose [24] (1.68 s / detection) | GigaPose (0.048 s / detection) | Reconstruction & Input segmentation | MegaPose [24] (1.68 s / detection) | GigaPose (0.048 s / detection) |

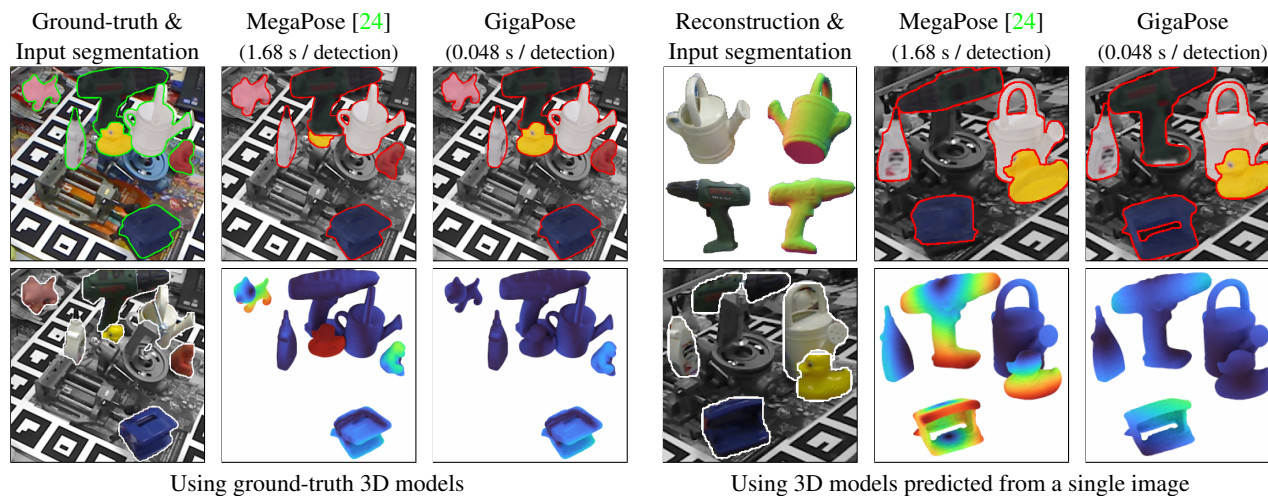Using ground-truth 3D models   Using 3D models predicted from a single image

Figure 1. **Comparison of our method GigaPose with MegaPose [24].** GigaPose is (i) more robust to noisy segmentation, often due to occlusions, (ii) more accurate with 3.5 % average precision improvement on the BOP benchmark [54], and (iii) significantly faster with a speed up factor of 35× per detection for coarse object pose estimation stage (0.048 s vs 1.68 s). Left example compares the results using accurate 3D models, while the right example shows the results with 3D models predicted from a single image by Wonder3D [34]. The bottom row shows the input segmentation, and the depth error heatmap of each detected object with respect to the ground truth pose, i.e the distance between each 3D point in the ground-truth depth map and its position with the predicted pose (legend: 0 cm ▬▬▬ 10 cm).

Abstract

We present GigaPose, a fast, robust, and accurate method for CAD-based novel object pose estimation in RGB images. GigaPose first leverages discriminative "templates", rendered images of the CAD models, to recover the out-of-plane rotation and then uses patch correspondences to estimate the four remaining parameters. Our approach samples templates in only a two-degrees-of-freedom space instead of the usual three and matches the input image to the templates using fast nearest-neighbor search in feature space, results in a speedup factor of 35× compared to the state of the art. Moreover, GigaPose is significantly more robust to segmentation errors. Our extensive evaluation on the seven core datasets of the BOP challenge demonstrates that it achieves state-of-the-art accuracy and can be seamlessly integrated with existing refinement methods. Additionally, we show the potential of GigaPose with 3D models predicted by recent work on 3D reconstruction from a single image, relaxing the need for CAD models and making 6D pose object estimation much more convenient. Our source code and trained models are publicly available at https://github.com/nv-nguyen/gigaPose.

1. Introduction

6D object pose estimation has significantly improved over the past decade [22,23,27,32,52,55,57,61]. However, supervised deep learning methods, despite remarkable accuracy, are cumbersome to deploy to an industrial setting. Indeed, for each novel object, the pose estimation model needs to be retrained using newly-acquired data, which is impractical: Retraining typically takes several hours or days [23,32], and the end users might not have the skills to retrain the model.

To fulfill the needs of such industrial settings, CAD-based novel object pose estimation, which focuses on estimating the 6D pose of *novel* objects (i.e., objects only available at inference time, not during training), has garnered attention and was introduced in the latest BOP challenge [16]. Current approaches involve three main steps: object detection and segmentation, coarse pose estimation, and refinement. While object detection and segmentation has been recently addressed by CNOS [41], refinement has been also addressed effectively with render-and-compare approaches [24, 56]. However, existing solutions to coarse pose estimation still

suffer from low inference speed and sensitivity to segmentation errors. We thus focus on this step in this paper.

The low inference speed stems from how existing coarse pose estimation methods rely on templates [24, 42, 62]. Among them, MegaPose [24] has been widely adopted and integrated into various pipelines, notably the BOP challenge-winner GenFlow [37]. However, the complexity of MegaPose is linear in the number of templates, as it matches the input images against the templates by running a network on each image-template pair. As a result, the methods based on MegaPose require more than 1.6 seconds *per detection.*

Sensitivity to detection and segmentation errors, often due to occlusions, is a common issue for template-based approaches [24, 42]. As illustrated in Figure 1, the segmentation of occluded objects such as the "duck" (left example), results in a scale and translation mismatch when cropping the test image and templates. Additionally, the erroneous segments may include noisy signal from the other objects or the background, which results in numerous outlier matches between the input image and the templates.

To address these two major limitations, we introduce GigaPose, a novel approach for CAD-based coarse object pose estimation. GigaPose makes several technical contributions towards speed and robustness and can be seamlessly integrated with any refinement method for CAD-based novel object pose estimation to achieve state-of-the-art accuracy.

The key idea in GigaPose is to find the right trade-off between the use of templates, which have been shown to be extremely useful for estimating the pose of novel objects, and patch correspondences, which lead to better robustness and more accurate pose estimates. More precisely, we propose to rely on templates to estimate two degrees of freedom (DoFs)—azimuth and elevation—as varying these angles changes the appearance of an object in complex ways, which templates excel at capturing effectively. Our templates are represented with local features that are trained to be robust to scaling and in-plane rotations. Matching the input image with the templates based on these local features yields robustness to segmentation errors.

To estimate the remaining 4 DoFs—in-plane rotation and 3D translation decomposed into 2D translation and 2D scale, we rely on patch correspondences between the input image and the template candidates. Given a template candidate, we match its local features with those of the input image, which gives us 2D-2D point correspondences. Instead of simply exploiting the matched point coordinates and use a PnP algorithm [38] to estimate the pose as done in previous works [17, 18, 52], we also exploit their appearances: We show that it is possible to predict the in-plane rotation and relative scale between the input image and the template from local features computed at the matched points. The remaining 2D translation is obtained from the positions of these matched points, allowing the estimation of the four DoFs

from a single correspondence. To robustify this estimate, we combine this process with RANSAC.

We experimentally demonstrate that our balance between the use of templates and patch correspondences effectively addresses the two issues in coarse pose estimation. Indeed, our method relies on a sublinear nearest-neighbor template search, successfully addressing the low inference speed issue with a speedup factor of $35\times$ per detection compared to to MegaPose [24]. Furthermore, the two steps of our method are particularly robust to segmentation errors.

We also demonstrate that GigaPose can exploit a 3D model reconstructed from a single image by a diffusion-based model [31, 33, 34, 50, 51, 58] instead of an accurate CAD model. Despite the inaccuracies of the predicted 3D models, our method can recover an accurate 6D pose as shown on Figure 1. This relaxes the need for CAD models and makes 6D pose object detection much more convenient.

In summary, our contribution is a novel RGB-based method for CAD-based novel object coarse pose estimation from a single correspondence that is significantly faster, more robust, and more accurate than existing methods. We demonstrate this through extensive experiments on the seven core datasets of the BOP challenge [54].

2. Related Work

Seen object pose estimation. Early works on 6D pose estimation have introduced diverse benchmarks to evaluate the performance of their approaches [2,8,10,13,14,21,54,60]. This data and its ground truth have powered many deep learning-based methods [17,22,23,27,28,32,47,52,55,61]. Some of them show remarkable performance in terms of run-time and accuracy [23, 32]. However, these approaches require long and expensive training, such as the state-of-the-art methods [23, 32] require several hours for training for a single object, making them too cumbersome for many practical applications in robotics and AR/VR.

To avoid the need for re-training when dealing with new object instances, one approach is to train on object categories by assuming that the testing objects belong to a known category [6, 19, 20, 26, 30, 35, 36, 59]. These category-level pose estimation methods, however, cannot generalize to objects beyond the scope of the training categories. By contrast, our method operates independently of any category-level information and seamlessly generalizes to novel categories.

Novel object pose estimation. Several techniques have been explored to improve the generalization of object pose estimation methods [1,4,24,29,39,40,42,43,46,49,53,56,62]. These can be roughly divided into feature-matching methods [1, 49] and template matching ones [1, 4, 24, 40, 42, 43, 46, 53, 62]. Feature-matching methods extract local features from the image, match them to the given 3D model and then use a variant of the PnP algorithm [38] to recover the 6D pose from the 3D-to-2D correspondences. By contrast, template
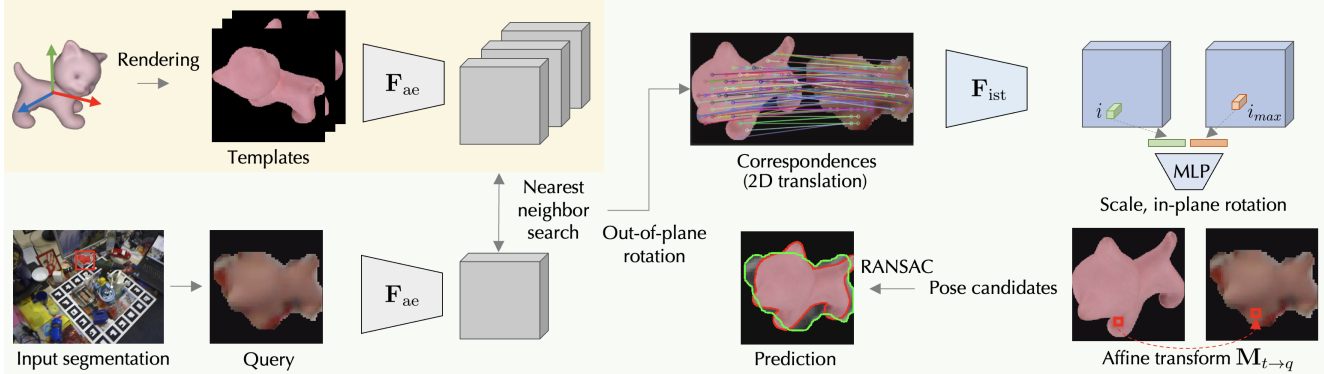
Figure 2. **Overview.** We first onboard each novel object by rendering 162 templates, spanning the spectrum of out-of-plane rotations. We also extract dense features using $\mathbf{F}_{ae}$ from each of the templates. At runtime, given a query image segmented with CNOS [41], we process it (by masking the background, cropping on the segment, adding padding then resizing), and extracting features with $\mathbf{F}_{ae}$. We retrieve the nearest template to the segment using the similarity metric detailed in Section 3.2. Further, 2D scale and in-plane rotation are computed from a single 2D-2D correspondence using $\mathbf{F}_{ist}$ and two lightweight MLPs. The 2D position of the correspondences also gives us the 2D translation which is used with 2D scale, in-plane rotation to create the affine transformation $\mathbf{M}_{t \to q}$, mapping the nearest template to the query image. This enables us to recover the complete 6D object pose from a single correspondence. Finally, we use RANSAC to robustly find the best pose candidate. Onboarding takes 11.5 seconds per object and inference takes 48 milliseconds per detection on average.

matching methods first render synthetic templates of the CAD models, and then use a deep network to compute a score for each input image-template pair, thus aiming to find the template with the most similar pose to the input image.

At the last BOP challenge [16], CAD-based *novel* object pose estimation was introduced as a new task, using CNOS [41] as the default detection method. MegaPose [24] and ZS6D [1] showed promising results for this new task. Nevertheless, MegaPose's run-time was highlighted as a significant limitation due to the need for a forward pass through a coarse pose estimator to compute a classification score for every (query, template) comparison. ZS6D relies on DINOv2 features [45] and the similarity metric of [42] to predict sparse 3D-to-2D correspondences. Unfortunately, their experiments do not evaluate the model's sensitivity to segmentation errors. In contrast, we present extensive evaluations and demonstrate that GigaPose outperforms both MegaPose and ZS6D. Our method can also be seamlessly integrated with any refinement method.

Correspondence-based object pose estimation. A classical approach to solving the 6D object pose estimation problem is to establish 3D-to-2D correspondences and compute the pose with a P*n*P algorithm [17, 18, 28, 48, 52, 55, 61], which requires at least four 3D-to-2D point correspondences. Because we first match the input image against templates and estimate the four remaining DoFs from a single 2D-to-2D match, we only need one correspondence to determine the 6D object pose. Our experimental evaluation shows that our method outperforms ZS6D, which, as stated above, is a state-of-the-art method relying on 3D-to-2D correspondences.

3. Method

Figure 3 provides an overview of GigaPose. Given the 3D model of an object of interest, we render templates and extract their dense features using a Vision-Transformer (ViT) model $\mathbf{F}_{ae}$. Then, given an input image, we detect the object of interest and segment it using an off-the-shelf method CNOS [41]. GigaPose extracts dense features from the input image at the object location using $\mathbf{F}_{ae}$ again. We select the template most similar to the input image using a similarity metric based on the dense features, detailed in Section 3.2. This gives us the azimuth and elevation of the camera.

To estimate the remaining DoFs, we look for corresponding patches between the input image and its most similar template. From one such pair of patches, we can directly predict two additional DoFs: the 2D scale s and the in-plane rotation α, by feeding two lightweight MLPs the features for the two patches extracted by another feature extractor denoted as $\mathbf{F}_{ist}$. Note that the features extracted by $\mathbf{F}_{ae}$ are not suitable here, as they discard information about scale and in-plane rotation by design. The image locations of the corresponding patches also give us directly the last two DoFs: the 2D translation $(\mathbf{t}_x, \mathbf{t}_y)$. From the scale and 2D translation, we can estimate the 3D translation. We use a RANSAC framework and iterate over different pairs of patches to find the optimal pose. We detail the training of $\mathbf{F}_{ist}$ and the MLPs, and the RANSAC scheme in Section 3.3.

3.1. Generating Templates

In contrast to other approaches [42, 62], we do not generate templates for both in-plane and out-of-plane rotations, as this yields thousands of templates. Instead, we decouple the 6 DoFs object pose into out-of-plane rotation, in-plane ro-
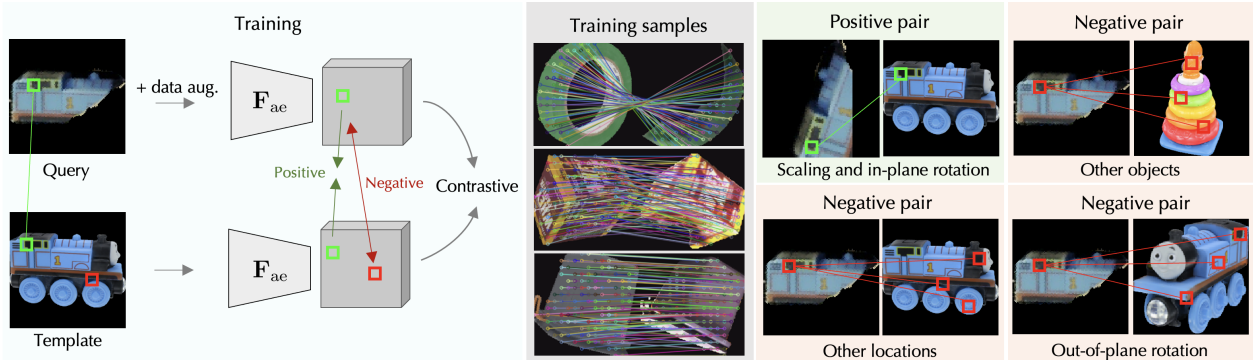
Figure 3. **Contrastive training of $\mathbf{F}_{ae}$.** We use pairs made of a query image and a template to train a network using local contrastive learning as detailed in Section 3.2. Middle: Training samples provided by [24], and the 2D-2D correspondences created from ground-truth 3D information used to generate positive and negative pairs. Right: We seek local features that vary with the out-of-plane rotation, but are invariant to in-plane rotation and scaling. Thus, positive pairs are made of corresponding patches under scaling and in-plane rotation changes, and negative pairs are made of corresponding patches under different out-of-plane rotations, patches that do not correspond, or that come from different objects.

tation, and 3D translation (2D translation and scale). Given the out-of-plane rotation, finding the scale and in-place rotation is indeed a 2D problem only. We thus create much less templates and push the estimation of the other DoFs to a later stage in the pipeline (see Section 3.3).

In practice, we use 162 templates. These are generated from viewpoints defined in a regular icosphere which is created by subdividing each triangle of the icosphere primitive of Blender into four smaller triangles. This has been shown in previous works [1, 41, 42] to provide well-distributed view coverage of CAD models.

3.2. Predicting Azimuth and Elevation

Training the feature extractor $\mathbf{F}_{ae}$. $\mathbf{F}_{ae}$ extracts dense features from both the input image and each of the templates *independently*. Compared to estimating features jointly, this approach eliminates the need for extensive feature extraction at runtime, a process that scales linearly with the number of templates and in-plane rotations considered. Instead, we can offload the computation of the features for each template to an onboarding stage for each novel object. We now describe how we train the feature extractor $\mathbf{F}_{ae}$ and how we design the similarity metric to compare the template and query features.

The extracted features aim to match a query image to a set of templates with different out-of-plane rotations, but with fixed scale, in-plane rotation, and translation. The features should thus be invariant to scale, in-plane rotation, and 2D translation, but be sensitive to out-of-plane rotation.

We achieve this with a local contrastive learning scheme. The main difficulty lies in defining the positive and negative patch pairs. Figure 3 illustrates our training procedure. We construct batches of B image pairs $(\mathcal{Q}_k, \mathcal{T}_k)$, such that the query $\mathcal{Q}_k$ is a rendering of a 3D object in any pose, and template $\mathcal{T}_k$ is another rendering of that object with the same out-of-plane rotation but different in-plane rotation, scale,

and 2D translation. Because we have access to the 3D model, we can compute ground-truth 2D-to-2D correspondences to create positive and negative pairs. We detail in the supplementary material how we compute these correspondences.

Additionally, since our goal is to close the domain gap between real images and synthetic renderings, we apply color augmentation along with random cropping and in-plane rotation to the input pairs. We use the training sets provided by the BOP challenge [54], originally sourced from MegaPose [24]. These datasets consist of 2 million images and are generated from CAD models of Google Scanned Objects [9] and ShapeNet [5] using BlenderProc [7]. We show typical training samples in the middle part of Figure 3.

We pass each image $\mathcal{Q}_k$ and $\mathcal{T}_k$ independently through $\mathbf{F}_{ae}$ to extract dense feature maps $\mathbf{q}_k$ and $\mathbf{t}_k$. Below, we use the superscript i to denote a 2D location in the local feature map. Note that because of the downsizing done by the ViT, each location i in the feature grid corresponds to a 14×14 patch in the respective input image. Each feature map has a respective segmentation mask $\mathbf{m}_{\mathcal{Q}_k}$ and $\mathbf{m}_{\mathcal{T}_k}$ corresponding to the foreground of the object in the images $\mathcal{Q}_k$ and $\mathcal{T}_k$.

For a location i in the query feature map $\mathbf{q}_k$, we denote by i^* the corresponding location in the template feature map. We arrange the query patches $(\mathbf{q}_k^i)$ and their corresponding patch $(\mathbf{t}_k^{i^*})$ in a square matrix such that the diagonal contains the positive pairs, and all other entries serve as negative pairs. For each pair $(\mathcal{Q}_k, \mathcal{T}_k)$, we thus obtain $|\mathbf{m}_{\mathcal{Q}_k}|$ positive pairs and $|\mathbf{m}_{\mathcal{Q}_k}| \times (|\mathbf{m}_{\mathcal{Q}_k}| - 1)$ negative pairs.

To improve the efficiency of contrastive learning, we use additional negative pairs from a query image $\mathcal{Q}_k$ and a template $\mathcal{T}_{k'}$, where $k' \neq k$ in the current batch. This process results in total in $\left(\sum_{k=1}^{B} |\mathbf{m}_{\mathcal{Q}_k}|\right)^2 - \sum_{k=1}^{B} |\mathbf{m}_{\mathcal{Q}_k}|$ negative pairs for the current batch. We train $\mathbf{F}_{ae}$ to align the representations of the positive pairs while separating the

negative pairs using the InfoNCE loss [44]:

$$\mathcal{L}_{\text{out}} = -\sum_{k=1}^{B} \sum_{i=1}^{|\mathbf{m}_{\mathcal{Q}_k}|} \ln \frac{e^{S(\mathbf{q}_k^i, \mathbf{t}_k^{i^*})/\tau}}{\sum_{(k',i')\neq(k,i^*)} e^{S(\mathbf{q}_k^i, \mathbf{t}_{k'}^{i'})/\tau}} , \quad (1)$$

where $S(.,.)$ is the cosine similarity between the local image features computed by the network $\mathbf{F}_{\text{ae}}$. The temperature parameter τ is set to 0.1 in our experiments.

Since the positive pairs of patches have different scales and in-plane rotations, our network $\mathbf{F}_{\text{ae}}$ learns to become invariant to these two factors, as demonstrated in our experiments. We initialize our feature extractor $\mathbf{F}_{\text{ae}}$ as DINOv2 [45] pretrained on ImageNet, because it has proven to be highly effective in extracting features for vision tasks.

Azimuth and elevation prediction. We define a pairwise similarity metric for each query-template $(\mathcal{Q}, \mathcal{T})$ pair with their respective dense feature grid $(\mathbf{q}, \mathbf{t})$ and feature segmentation masks $(\mathrm{m}_{\mathcal{Q}}, \mathrm{m}_{\mathcal{T}})$.

For each local query feature $\mathbf{q}^i$, corresponding to patch at location i, we compute its nearest neighbor in the template features $\mathbf{t}$, denoted as $\mathbf{t}^{i_{\max}}$, as

$$i_{\max} = \arg\max_{j|\mathrm{m}_{\mathcal{T}}^j>0} S\left(\mathbf{q}^i, \mathbf{t}^j\right) . \quad (2)$$

This nearest neighbor search yields a list of correspondences $\{(i, i_{\max})\}$. To improve the robustness of our method against outliers, we keep only the correspondences $\{(i, i_{\max})\}$ having a similarity score ≥ 0.5. The final similarity for this $(\mathcal{Q}, \mathcal{T})$ pair is defined as the mean of all the remaining correspondences, weighted by their similarity score:

$$\text{sim}(\mathbf{q}, \mathbf{t}) = \frac{1}{|\mathrm{m}_{\mathcal{Q}}|} \sum_i \mathrm{m}_{\mathcal{Q}}^i S\left(\mathbf{q}^i, \mathbf{t}^{i_{\max}}\right) . \quad (3)$$

We compute this score for all templates $\mathcal{T}_k$ ($1 \leq k \leq 162$) and find the top-K candidates yielding the most similar out-of-plane rotations. This nearest neighbor search is very fast and delivers results within TODO milliseconds. In practice, we experiment with $K = 1$ and $K = 5$. For the latter, the final template is selected by the RANSAC-based estimation detailed in Section 3.3 below.

3.3. Predicting the Remaining DoFs

Once we have identified the template candidates, we seek to estimate the remaining 4 DoFs, i.e., in-plane rotation, scale, and 2D translation, which yield the affine transformation $\mathbf{M}_{t\to q}$ transforming each template candidate $\mathcal{T}$ to the query image $\mathcal{Q}$. Specifically, we have

$$\mathbf{M}_{t\to q} = \begin{bmatrix} s\cos(\alpha) & -s\sin(\alpha) & \mathbf{t}_x \\ s\sin(\alpha) & s\cos(\alpha) & \mathbf{t}_y \\ 0 & 0 & 1 \end{bmatrix} , \quad (4)$$

where s is the 2D scaling factor, α is the relative in-plane rotation, and $[\mathbf{t}_x, \mathbf{t}_y]$ is the 2D translation between the input query image $\mathcal{Q}$ and the template $\mathcal{T}$.

Training the feature extractor $\mathbf{F}_{\text{ist}}$ and the MLP. We have already obtained from the features of $\mathbf{F}_{\text{ae}}$ a list of 2D-2D correspondences $\{(i, i_{\max})\}$. Each correspondence can inherently provide 2D translation $[\mathbf{t}_x, \mathbf{t}_y]$ information through the patch locations i and $i_{\max}$. To recover the remaining 2 DoFs, scale $\mathbf{s}$ and in-plane rotation α, we train deep networks to directly regress these values from a single 2D-2D correspondence. Since the feature extractor $\mathbf{F}_{\text{ae}}$ is invariant to in-plane rotation and scaling, the corresponding features cannot be used to regress those values, hence we have to train another feature extractor we call $\mathbf{F}_{\text{ist}}$. Given a 2D-2D match from a pair $(\mathcal{Q}, \mathcal{T})$, and their corresponding feature computed by $\mathbf{F}_{\text{ist}}$, we pass them through two small MLPs, which outputs directly α and $\mathbf{s}$. This enables us to predict 2D scale and in-plane rotations for each 2D-2D correspondence. To address the 2π periodicity of in-plane rotation, we predict $\left[\cos(\alpha_k^l), \sin(\alpha_k^l)\right]$ instead of α_k^l.

We train jointly both $\mathbf{F}_{\text{ist}}$ and the MLPs on the same data samples as $\mathbf{F}_{\text{ae}}$ using the loss:

$$\mathcal{L}_{\text{inp}} = \sum_{k=1}^{B} \sum_{i=1}^{n_k} \left[\left(\ln(s_k^i) - \ln(s_k^*)\right)^2 + \text{geo}(\alpha_k^i, \alpha_k^*)\right], \quad (5)$$

where s_k^* and α_k^* are the ground-truth scale and in-plane rotation between $\mathcal{Q}$ and $\mathcal{T}_k$, and $\text{geo}(\cdot, \cdot)$ indicates the geodesic loss defined as

$$\text{geo}(\alpha_1, \alpha_2) = \text{acos}\big(\cos(\alpha_1)\cos(\alpha_2) + \sin(\alpha_1)\sin(\alpha_2)\big) . \quad (6)$$

RANSAC-based $\mathbf{M}_{t\to q}$ estimation. For each template $\mathcal{T}$, we employ RANSAC on each $\mathbf{M}_{t\to q}$ predicted by each correspondence and validate them against the remaining correspondences using a 2D error threshold of δ. In practice, we set δ to the size of a patch, corresponding to an error of 14 pixels in image space. The final prediction for $\mathbf{M}_{t\to q}$ is determined by the correspondence with the highest number of inliers. The complete 6D object pose can finally be recovered from the out-of-plane rotation, in-plane rotation, 2D scale and 2D translation using the explicit formula provided in the supplementary material.

We initialize $\mathbf{F}_{\text{ist}}$ with a modified version of ResNet18 [12] instead of the DINOv2 [45] as DINOv2 is trained with random augmentations that includes in-plane rotations and cropping, making its features invariant to scale and in-plane rotation. Similarly to the features from $\mathbf{F}_{\text{ae}}$, we offload the feature computation of $\mathbf{F}_{\text{ist}}$ to the onboarding stage for all templates to avoid the computational burden at runtime.

Implementation details. We use the input image of size 224×224, resulting in features of size $16\times16\times1024$ and $16\times16\times256$ via the networks $\mathbf{F}_{\text{ae}}$ and $\mathbf{F}_{\text{ist}}$ respectively. We train our networks using the Adam optimizer with an initial learning rate of 1e-5 for $\mathbf{F}_{\text{ae}}$ and 1e-3 for $\mathbf{F}_{\text{ist}}$. The training process takes less than 10 hours when using four V100 GPUs. All the inference experiments are run on a single V100 GPU.

#	Method	Detections	Refinement	LM-O	T-LESS	TUD-L	IC-BIN	ITODD	HB	YCB-V	MEAN	RUN-TIME
			Num. instances:	1445	6423	600	1786	3041	1630	4123		
1	OSOP [53]	OSOP [53]	–	27.4	40.3	–	–	–	–	29.6	–	–
2	MegaPose [24]	Mask R-CNN [11]	–	18.7	19.7	20.5	15.3	8.00	18.6	13.9	16.2	–
3	ZS6D [1]	CNOS [41]	–	29.8	21.0	–	–	–	–	32.4	–	–
4	MegaPose [24]	CNOS [41]	–	22.9	17.7	25.8	15.2	10.8	25.1	28.1	20.8	15.5 s
5	GigaPose (Ours)	CNOS [41]	–	29.6	26.4	30.0	22.3	17.5	34.1	27.8	26.8	0.4 s
6	MegaPose [24]	CNOS [41]	MegaPose [24]	49.9	47.7	65.3	36.7	31.5	65.4	60.1	50.9	17.0 s
7	GigaPose (Ours)	CNOS [41]	MegaPose [24]	55.7	54.1	58.0	45.0	37.6	69.3	63.2	54.7	2.3 s
8	MegaPose [24]	CNOS [41]	MegaPose + 5 Hypotheses [24]	56.0	50.7	68.4	41.4	33.8	70.4	62.1	54.7	21.9 s
9	GigaPose (Ours)	CNOS [41]	MegaPose + 5 Hypotheses [24]	59.8	56.5	63.1	47.3	39.7	72.2	66.1	57.8	7.7 s
10	MegaPose [24]	CNOS [41]	GenFlow + 5 Hypotheses [37]	56.3	52.3	68.4	45.3	39.5	73.9	63.3	57.0	20.8 s
11	GigaPose (Ours)	CNOS [41]	GenFlow + 5 Hypotheses [37]	63.1	58.2	66.4	49.8	45.3	75.6	65.2	60.5	10.6 s

Table 1. **Results on the BOP datasets.** We report the AR score on each of the seven core datasets of the BOP challenge and the mean score across datasets. The best results with CNOS's detections [41] without refinement are highlighted in blue, with MegaPose's refinement using 1 hypothesis in yellow, and using 5 hypotheses in orange, and with GenFlow's refinement using 5 hypotheses in red.

4. Experiments

In this section, we first describe our experimental setup (Section 4.1). Next, we compare our method with previous works [1, 24, 53] on the seven core datasets of the BOP challenge [16] (Section 4.2). We conduct this comparison to evaluate our method's accuracy, runtime performance, and robustness to segmentation errors, highlighting our contributions. Finally, we present an ablation study that explores different settings of our method (Section 4.4).

4.1. Experimental Setup

Evaluation Datasets. We evaluate our method on the seven core datasets of the BOP challenge [16]: LineMod Occlusion (LM-O) [2], T-LESS [13], TUD-L [14], IC-BIN [8], ITODD [10], HomebrewedDB (HB) [21] and YCB-Video (YCB-V) [60]. These datasets consist of a total of 132 different objects and 19048 testing instances, presented in cluttered scenes under partial occlusions. It is worth noting that, in contrast to the *seen* object setting, the *novel* object pose estimation setting is far from being saturated in terms of both accuracy and run-time.

Evaluation metrics. For all experiments, we use the standard BOP evaluation protocol [15], which relies on three metrics: Visible Surface Discrepancy (VSD), Maximum Symmetry-Aware Surface Distance (MSSD), and Maximum Symmetry-Aware Projection Distance (MSPD). The final score, referred to as the average recall (AR), is calculated by averaging the individual average recall scores of these three metrics across a range of error thresholds.

Baselines. We compare our method with MegaPose [24], ZS6D [1], and OSOP [53]. As of the time of writing, the source codes for ZS6D and OSOP are not available. Therefore, we can only report their performance as provided in their papers, but not their run-time.

Refinement. To demonstrate the potential of GigaPose, we have applied the refinement methods from MegaPose [24] and GenFlow [37] to our results. We extract the top-1 and the top-5 pose candidates and subsequently refine them using 5 iterations of MegaPose's refinement network [24] or GenFlow's refinement network [37]. For the top-5 hypotheses case, these refined hypotheses are scored by the coarse network of MegaPose [24], and the best one is selected.

Pose estimation with a 3D model predicted from a single image. We use Wonder3D [34] to predict a 3D model from a single image for objects from LM-O. We then evaluate the performance of MegaPose and our method using reconstructed models instead of the accurate CAD models provided by the dataset. Due to the sensitivity of Wonder3D to the quality of input images, we carefully select reference images. More details about this setting are present in the supplementary material.

4.2. Comparison with the State of the Art

Accuracy. Table 1 compares the results of our method with those of previous work [1, 24, 53]. Across all settings, whether with or without refinement, our method consistently outperforms MegaPose while maintaining significantly faster processing times. Notably, our method significantly improves accuracy on the challenging T-LESS, IC-BIN, and ITODD, with more than a 5% increase in AR score for coarse pose estimation and more than a 4% increase in AR score after refinement compared to MegaPose.

It is important to note that although the coarse and refinement networks in MegaPose [24] are not trained together, they were trained to *work* together: As mentioned in Section 3.2 of [24], the positive samples of the coarse network *"are sampled from the same distribution used to generate the perturbed poses the refiner network is trained to correct"*. This pose sampling biases MegaPose's refinement process towards MegaPose's coarse estimation errors. This explains

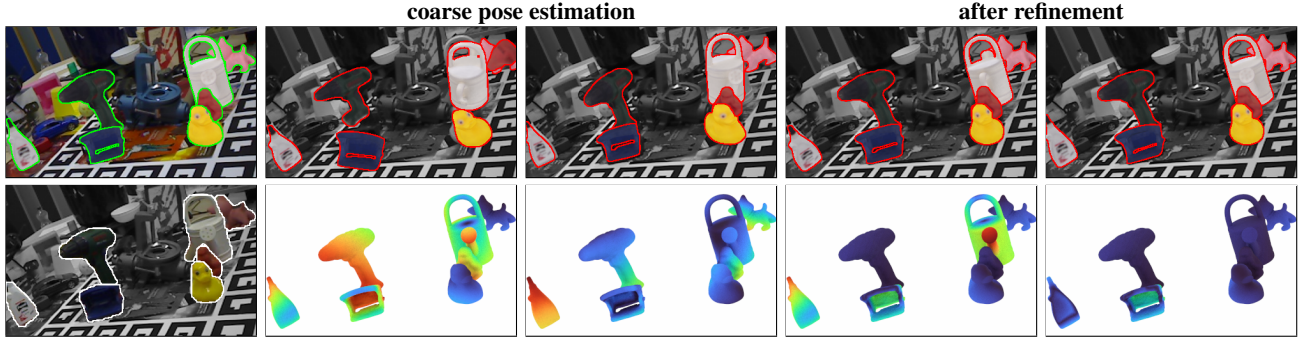**coarse pose estimation**   **after refinement**

Figure 4. **Qualitative results on LM-O [2].** The first column shows the ground-truth and CNOS [41] segmentation. The second and third columns show the results without refinement for both MegaPose [24] and our method, including depth error heatmaps at the bottom. The last two columns compare the results using the same refinement [24] for MegaPose [24] and our method. In the error heatmap, darker red indicates higher error with respect to the ground truth pose (legend: 0 cm ▬▬▬▬ 10 cm). As demonstrated in this figure, our method estimates a more accurate coarse pose and avoids local minima during refinement, such as with the white "watering_can" object from LM-O.
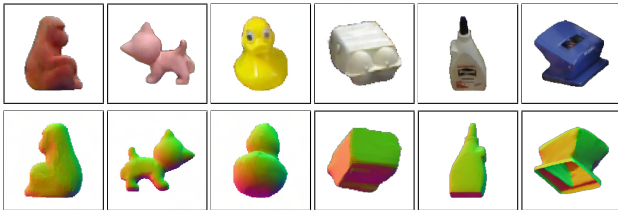
Figure 5. **3D recontruction by Wonder3D [34].** The first row displays the input reference image, the second shows the predicted normal maps from the view opposite to the reference image. More visualizations are provided in the supplementary material.

Method	Detection [41]	Single image		GT 3D model w/o refinement
		Coarse	Refined	
1 MegaPose [24]	GT 3D model	16.3	25.6	22.9
2 GigaPose (ours)	GT 3D model	**18.3**	**27.9**	**29.6**
3 MegaPose [24]	Single image	15.4	25.2	22.7
4 GigaPose (ours)	Single image	**17.6**	**27.2**	**29.4**

Table 2. **Results with predicted 3D models on LM-O [2].** We report AR score using 3D models predicted from a single reference image by Wonder3D [34]. The 3D reconstruction is shown in Figure 5. Rows 3 and 4 display additional results for MegaPose and our method, where CNOS [41] is also given 3D predicted models.

why the refinement process brings larger improvements to MegaPose than GigaPose, in particular on TUD-L where the refinement improves MegaPose by 39.5% and our method only by 28.0%. However, TUD-L represents only about 3% of the total test data, our method still outperforms MegaPose over the 7 datasets in all settings.

Figure 4 shows qualitative comparisons with Mega-Pose [24] before and after refinement showcasing our more accurate pose estimates. More qualitative results are provided in the supplementary material.

Accuracy when using predicted 3D models. As shown in Table 2, our method outperforms MegaPose when using predicted 3D models. Results in Table 2 implies that when

Method	Run-time		
	Onboarding	Coarse pose	Refinement [24]
MegaPose [24]	0.82 s	1.68 s	33 ms
GigaPose (ours)	11.5 s	48 ms	33 ms

Table 3. **Run-time.** Breakdown of the average run-time for each stage of MegaPose [24] and our method on a single V100 GPU to estimate the pose per object (i.e., per detection). Our method is more than $35\times$ faster than MegaPose for coarse pose estimation.

no CAD model is available for an object, we can use Wonder3D to predict a 3D model from a single image, then apply GigaPose and MegaPose refinement. These results are close to GigaPose's performance and surpass MegaPose's coarse performance when using an accurate CAD model.

Run-time. We report the speed of GigaPose in Table 1 (rightmost column) following the BOP evaluation protocol. It measures the total processing time *per image* averaged over the datasets including the time taken by CNOS [41] to segment each object, the time to estimate the object pose for all detections, and the refinement time if applicable.

Table 3 gives a breakdown of the run-time *per detection* for each stage of MegaPose and of our method. Our method takes only 48 ms for coarse pose estimation, more than 38x faster than the 1.68 seconds taken by MegaPose. This improvement can be attributed to our sublinear nearest neighbor search, significantly faster than feed-forwarding each of the 576 input-template pairs as done in MegaPose.

Robustness to segmentation errors. To demonstrate the robustness of our method, we analyze its performance under various levels of segmentation errors on three standard datasets: LM-O [2], T-LESS [13], and YCB-V [60]. We use the ground-truth masks to classify the segmentation errors produced by CNOS's segmentation [41] using the Intersection over Union (IoU) metric. For each IoU threshold, we retain only the input masks from CNOS that matched the
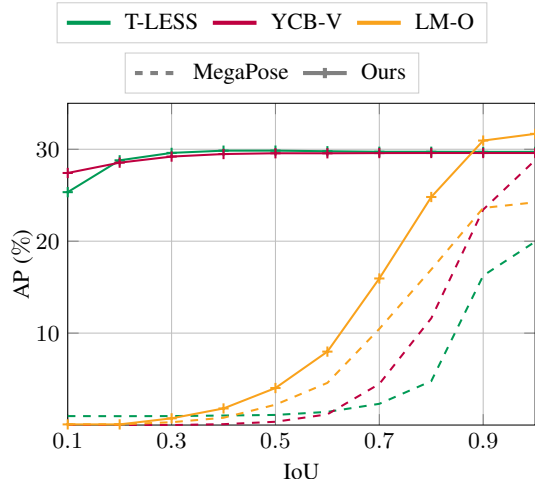
Figure 6. **Robustness to segmentation errors.** We analyze the performance of MegaPose and our method under various levels of segmentation errors, defined by the IoU between the predicted masks from CNOS [41] and the ground-truth masks. Our method demonstrates much higher stability in AP across all IoU thresholds than MegaPose, showing its robustness against segmentation errors. The improvement is more limited on LM-O because of the small appearance size of the objects especially after occlusions.
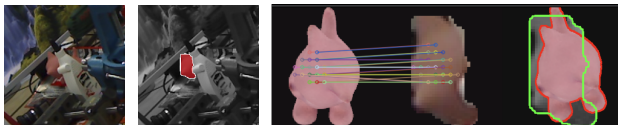
Input RGB   Segmentation [41]   2D-to-2D correspondences   Prediction

Figure 7. **Failure case.** The "Cat" object of LM-O [2] is not retrieved correctly here because of the small size of its segment and the low-fidelity CAD models, resulting in outlier matches.

ground-truth masks with an IoU smaller than this threshold and evaluate the AR score for coarse pose estimation.

As shown in Figure 6, our method has a stable AR score across all IoU thresholds for both T-LESS and YCB-V, in contrast with MegaPose, which yields high scores primarily for high IoU thresholds only.

4.3. Failure cases

Figure 6 also shows that the AR score stability is less important in the case of LM-O where many challenging conditions are present, including heavy occlusions, low resolution segmentation, and low-fidelity CAD models. We show in Figure 7 this failure case on the "Cat" object of LM-O.

4.4. Ablation Study

In Table 4, we present several ablation evaluations on the three standard datasets LM-O [2], T-LESS [13], and YCB-V [60]. Our results are in Row 5 .

Fine-tuning $\mathbf{F}_{ae}$. Row 1 of Table 4 presents the results of using the DINOv2 features [45] without fine-tuning $\mathbf{F}_{ae}$. As shown in Row 5, fine-tuning significantly improves template-correspondences, leading to a 8.9% increase in AR score.

	Fine tune $\mathbf{F}_{ae}$	Templates in-plane	PnP Type	n	LM-O	T-LESS	YCB-V	MEAN
1	✗	✗	2D-to-2D	1	20.1	19.3	17.7	19.0
2	✓	✓	2D-to-2D	1	23.3	21.1	22.1	22.1
3	✓	✗	3D-to-2D	4	28.0	25.3	26.3	26.5
4	✓	✗	2D-to-2D	2	**30.0**	25.6	26.0	27.2
5	✓	✗	2D-to-2D	1	29.6	**26.4**	**27.8**	**27.9**
6	✓	✗	2D-to-2D	1	*30.1*	*27.1*	*28.4*	*28.5*

Table 4. **Ablation study.** We report the AR score of different settings of our method including: without fine-tuning $\mathbf{F}_{ae}$ in Row 1, estimating in-plane rotation with dense 3DoF templates in Row 2, different "PnP" variants in Rows 3 and 4. The results of the complete method are on Row 5 . We show in Row 6 our results using the same 576 templates as in MegaPose [24]. See Section 4.4.

Estimating in-plane rotation with templates. Row 2 of Table 4 shows in-plane rotation estimation results using templates by dividing in-plane angle into 36 bins of 10 degrees, yielding 5832 templates per object. This approach decreases the AR score by 5.8% compared to direct predictions with $\mathbf{F}_{ist}$ and $\mathbf{H}$ in Row 5, underscoring the effectiveness of our hybrid template-patch correspondence approach.

2D-to-2D vs 3D-to-2D correspondences. In Row 3, we introduce a "3D-to-2D correspondence" variant by replacing the 2D locations of the matched patches in the template with their 3D counterparts obtained from the template depth map. We then estimate the complete 6D object pose using the ePnP algorithm [25] implemented in OpenCV [3]. Furthermore, in Row 4, we present a two-"2D-to-2D correspondences" variant, where the scale and in-plane rotation are computed using a 2D variant of the Kabsch algorithm (more details are given in the supplementary material). Our single-correspondence approach in Row 5 is more effective at exploiting patch correspondences for estimating scale and in-plane rotation directly.

Number of templates. In Row 6, we present our results using the same 576 templates as MegaPose [24]. This improves by only 0.6% the AR score compared to using 162 templates (Row 5). This confirms that the correspondences also allows to decrease the memory footprint of the templates without hurting the accuracy.

5. Conclusion

We presented GigaPose, an efficient method for the 6D coarse pose estimation of novel objects. It stands out for its significant speed, robustness, and accuracy compared to existing methods, and can be seamlessly integrated with any refinement methods. We hope that GigaPose will make real-time accurate pose estimation of novel objects practical. We discuss avenues for future work in the supplementary.

References

[1] Philipp Ausserlechner, David Haberger, Stefan Thalhammer, Jean-Baptiste Weibel, and Markus Vincze. ZS6D: Zero-Shot 6D Object Pose Estimation Using Vision Transformers. In *arXiv*, 2023. 2, 3, 4, 6

[2] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6D Object Pose Estimation Using 3D Object Coordinates. In *ECCV*, 2014. 2, 6, 7, 8

[3] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. 8

[4] Dingding Cai, Janne Heikkilä, and Esa Rahtu. Ove6d: Object viewpoint encoding for depth-based 6d object pose estimation. In *CVPR*, 2022. 2

[5] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, and Others. ShapeNet: An Information-Rich 3D Model Repository. In *arXiv*, 2015. 4

[6] Xu Chen, Zijian Dong, Jie Song, Andreas Geiger, and Otmar Hilliges. Category Level Object Pose Estimation via Neural Analysis-By-Synthesis. In *ECCV*, 2020. 2

[7] Maximilian Denninger, Martin Sundermeyer, Dominik Winkelbauer, Youssef Zidan, Dmitry Olefir, Mohamad El-badrawy, Ahsan Lodhi, and Harinandan Katam. BlenderProc. In *arXiv*, 2019. 4

[8] Andreas Doumanoglou, Rigas Kouskouridas, Sotiris Malassiotis, and Tae-Kyun Kim. Recovering 6D Object Pose and Predicting Next-Best-View in the Crowd. In *CVPR*, 2016. 2, 6

[9] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B. Mchugh, and Vincent Vanhoucke. Google Scanned Objects: A High-Quality Dataset of 3D Scanned Household Items. In *ICRA*, 2022. 4

[10] Bertram Drost, Markus Ulrich, Paul Bergmann, Philipp Hartinger, and Carsten Steger. Introducing Mvtec Itodd-A Dataset for 3D Object Recognition in Industry. In *ICCV Workshops*, 2017. 2, 6

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017. 6

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016. 5

[13] Tomas Hodan, Pavel Haluza, Stepan Obdrzalek, Jiri Matas, Manolis Lourakis, and Xenophon Zabulis. T-LESS: An RGB-D Dataset for 6D Pose Estimation of Texture-Less Objects. In *WACV*, 2017. 2, 6, 7, 8

[14] Tomas Hodan, Frank Michel, Eric Brachmann, Wadim Kehl, Anders Glentbuch, Dirk Kraft, Bertram Drost, Joel Vidal, Stephan Ihrke, Xenophon Zabulis, and Others. BOP: Benchmark for 6D Object Pose Estimation. In *ECCV*, 2018. 2, 6

[15] Tomas Hodan, Martin Sundermeyer, Bertram Drost, Yann Labbé, Eric Brachmann, Frank Michel, Carsten Rother, and Jiri Matas. BOP Challenge 2020 on 6D Object Localization. In *ECCV*, 2020. 6

[16] Tomas Hodan, Martin Sundermeyer, Yann Labbé, Van Nguyen Nguyen, Gu Wang, Eric Brachmann, Bertram Drost, Vincent Lepetit, Carsten Rother, and Jiri Matas. BOP Challenge 2023 on Detection, Segmentation and Pose Estimation of Seen and Unseen Rigid Objects. In *arXiv*, 2024. 1, 3, 6

[17] Yinlin Hu, Pascal Fua, Wei Wang, and Mathieu Salzmann. Single-Stage 6D Object Pose Estimation. In *CVPR*, 2020. 2, 3

[18] Yinlin Hu, Joachim Hugonot, Pascal Fua, and Mathieu Salzmann. Segmentation-Driven 6D Object Pose Estimation. In *CVPR*, 2019. 2, 3

[19] Muhammad Zubair Irshad, Thomas Kollar, Michael Laskey, Kevin Stone, and Zsolt Kira. Centersnap: Single-shot multi-object 3d shape reconstruction and categorical 6d pose and size estimation. In *ICRA*, 2022. 2

[20] Muhammad Zubair Irshad, Sergey Zakharov, Rares Ambrus, Thomas Kollar, Zsolt Kira, and Adrien Gaidon. Shapo: Implicit representations for multi-object shape appearance and pose optimization. In *ECCV*, 2022. 2

[21] Roman Kaskman, Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. Homebreweddb: RGB-D Dataset for 6D Pose Estimation of 3D Objects. In *ICCV Workshops*, 2019. 2, 6

[22] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. SSD-6D: Making RGB-Based 3D Detection and 6D Pose Estimation Great Again. In *ICCV*, 2017. 1, 2

[23] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. CosyPose: Consistent Multi-View Multi-Object 6D Pose Estimation. In *ECCV*, 2020. 1, 2

[24] Yann Labbé, Lucas Manuelli, Arsalan Mousavian, Stephen Tyree, Stan Birchfield, Jonathan Tremblay, Justin Carpentier, Mathieu Aubry, Dieter Fox, and Josef Sivic. MegaPose: 6D Pose Estimation of Novel Objects via Render & Compare. In *CoRL*, 2022. 1, 2, 3, 4, 6, 7, 8

[25] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. EP N P: An Accurate O (n) Solution to the P N P Problem. *IJCV*, 81, 2009. 8

[26] Fu Li, Ivan Shugurov, Benjamin Busam, Minglong Li, Shaowu Yang, and Slobodan Ilic. Polarmesh: A Star-Convex 3D Shape Approximation for Object Pose Estimation. *IEEE Robotics and Automation Letters*, 7(2), 2022. 2

[27] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. DeepIM: Deep Iterative Matching for 6D Pose Estimation. In *ECCV*, 2018. 1, 2

[28] Zhigang Li, Gu Wang, and Xiangyang Ji. CDPN: Coordinates-Based Disentangled Pose Network for Real-Time RGB-Based 6DoF Object Pose Estimation. In *ICCV*, 2019. 2, 3

[29] Jiehong Lin, Lihua Liu, Dekun Lu, and Kui Jia. Sam-6d: Segment anything model meets zero-shot 6d object pose estimation. *arXiv preprint arXiv:2311.15707*, 2023. 2

[30] Yunzhi Lin, Jonathan Tremblay, Stephen Tyree, Patricio A. Vela, and Stan Birchfield. Single-Stage Keypoint-Based Category-Level Object Pose Estimation from an RGB Image. In *ICRA*, 2022. 2

[31] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *ICCV*, 2023. 2

[32] Xingyu Liu, Ruida Zhang, Chenyangguang Zhang, Bowen Fu, Jiwen Tang, Xiquan Liang, Jingyi Tang, Xiaotian Cheng, Yukang Zhang, Gu Wang, and Xiangyang Ji. GDRNPP. https://github.com/shanice-l/gdrnpp_bop2022, 2022. 1, 2

[33] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Learning to generate multiview-consistent images from a single-view image. *arXiv preprint arXiv:2309.03453*, 2023. 2

[34] 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. *arXiv preprint arXiv:2310.15008*, 2023. 1, 2, 6, 7

[35] Fabian Manhardt, Gu Wang, Benjamin Busam, Manuel Nickel, Sven Meier, Luca Minciullo, Xiangyang Ji, and Nassir Navab. CPS++: Improving Class-Level 6D Pose and Shape Estimation from Monocular Images with Self-Supervised Learning. In *arXiv*, 2020. 2

[36] Lucas Manuelli, Wei Gao, Peter Florence, and Russ Tedrake. KPAM: Keypoint Affordances for Category-Level Robotic Manipulation. In *ISRR*, 2019. 2

[37] Sungphill Moon, Hyeontae Son, Dongcheol Hur, and Sangwook Kim. GenFlow: Generalizable Recurrent Flow for 6D Pose Refinement of Novel Objects. In *arXiv*, 2024. 2, 6

[38] Francesc Moreno-Noguer, Vincent Lepetit, and Pascal Fua. Accurate Non-Iterative O(n) Solution to the PnP Problem. In *ICCV*, 2007. 2

[39] Van Nguyen Nguyen, Yuming Du, Yang Xiao, Michael Ramamonjisoa, and Vincent Lepetit. PIZZA: A Powerful Image-Only Zero-Shot Zero-CAD Approach to 6 DoF Tracking. In *3DV*, 2022. 2

[40] Van Nguyen Nguyen, Thibault Groueix, Georgy Ponimatkin, Yinlin Hu, Marlet Renaud, Mathieu Salzmann, and Vincent Lepetit. NOPE: Novel Object Pose Estimation from a Single Image. In *CVPR*, 2024. 2

[41] Van Nguyen Nguyen, Thibault Groueix, Georgy Ponimatkin, Vincent Lepetit, and Tomas Hodan. CNOS: A Strong Baseline for CAD-based Novel Object Segmentation. In *ICCV Workshops*, 2023. 1, 3, 4, 6, 7, 8

[42] Van Nguyen Nguyen, Yinlin Hu, Yang Xiao, Mathieu Salzmann, and Vincent Lepetit. Templates for 3D Object Pose Estimation Revisited: Generalization to New Objects and Robustness to Occlusions. In *CVPR*, 2022. 2, 3, 4

[43] Brian Okorn, Qiao Gu, Martial Hebert, and David Held. Zephyr: Zero-Shot Pose Hypothesis Rating. In *ICRA*, 2021. 2

[44] Aaron Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding. In *arXiv*, 2018. 5

[45] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, and Others. Dinov2: Learning Robust Visual Features Without Supervision. In *arXiv*, 2023. 3, 5, 8

[46] Evin Pınar Örnek, Yann Labbé, Bugra Tekin, Lingni Ma, Cem Keskin, Christian Forster, and Tomas Hodan. Foundpose: Unseen object pose estimation with foundation features. *arXiv preprint arXiv:2311.18809*, 2023. 2

[47] Kiru Park, Timothy Patten, and Markus Vincze. Pix2pose: Pixel-Wise Coordinate Regression of Objects for 6D Pose Estimation. In *ICCV*, 2019. 2

[48] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. PVNet: Pixel-Wise Voting Network for 6DoF Pose Estimation. In *CVPR*, 2019. 3

[49] Giorgia Pitteri, Aurélie Bugeau, Slobodan Ilic, and Vincent Lepetit. 3D Object Detection and Pose Estimation of Unseen Objects in Color Images with Local Surface EMbeddings. In *ACCV*, 2020. 2

[50] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. DreamFusion: Text-to-3D Using 2D Diffusion. In *arXiv*, 2022. 2

[51] Guocheng Qian, Jinjie Mai, Abdullah Hamdi, Jian Ren, Aliaksandr Siarohin, Bing Li, Hsin-Ying Lee, Ivan Skorokhodov, Peter Wonka, Sergey Tulyakov, and Bernard Ghanem. Magic123: One image to high-quality 3d object generation using both 2d and 3d diffusion priors. *arXiv preprint arXiv:2306.17843*, 2023. 2

[52] Mahdi Rad and Vincent Lepetit. BB8: A Scalable, Accurate, Robust to Partial Occlusion Method for Predicting the 3D Poses of Challenging Objects Without Using Depth. In *ICCV*, 2017. 1, 2, 3

[53] Ivan Shugurov, Fu Li, Benjamin Busam, and Slobodan Ilic. OSOP: A Multi-Stage One Shot Object Pose Estimation Framework. In *CVPR*, 2022. 2, 6

[54] Martin Sundermeyer, Tomáš Hodaň, Yann Labbe, Gu Wang, Eric Brachmann, Bertram Drost, Carsten Rother, and Jiří Matas. BOP Challenge 2022 on Detection, Segmentation and Pose Estimation of Specific Rigid Objects. In *CVPR Workshops*, 2023. 1, 2, 4

[55] Bugra Tekin, Sudipta N. Sinha, and Pascal Fua. Real-Time Seamless Single Shot 6D Object Pose Prediction. In *CVPR*, 2018. 1, 2, 3

[56] Jonathan Tremblay, Bowen Wen, Valts Blukis, Balakumar Sundaralingam, Stephen Tyree, and Stan Birchfield. Diff-DOPE: Differentiable Deep Object Pose Estimation. In *arXiv*, 2023. 1, 2

[57] Gu Wang, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. GDR-Net: Geometry-Guided Direct Regression Network for Monocular 6D Object Pose Estimation. In *CVPR*, 2021. 1

[58] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A. Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pre-trained 2d diffusion models for 3d generation. *arXiv preprint arXiv:2212.00774*, 2022. 2

[59] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J. Guibas. Normalized Object Coordinate Space for Category-Level 6D Object Pose and Size Estimation. In *CVPR*, 2019. 2

[60] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes. In *RSS*, 2018. 2, 6, 7, 8

[61] Sergey Zakharov, Ivan S. Shugurov, and Slobodan Ilic. DPOD: 6D Pose Object Detector and Refiner. In *ICCV*, 2019. 1, 2, 3

[62] Chen Zhao, Yinlin Hu, and Mathieu Salzmann. Fusing Local Similarities for Retrieval-based 3D Orientation Estimation of Unseen Objects. In *ECCV*, 2022. 2, 3