

# FoundPose: Unseen Object Pose Estimation with Foundation Features

Evin Pınar Örnek<sup>1\*</sup> Yann Labbé<sup>2</sup> Bugra Tekin<sup>2</sup> Lingni Ma<sup>2</sup>  
Cem Keskin<sup>2</sup> Christian Forster<sup>2</sup> Tomas Hodan<sup>2</sup>

<sup>1</sup>Technical University of Munich <sup>2</sup>Meta Reality Labs

**Abstract.** We propose FoundPose, a model-based method for 6D pose estimation of unseen objects from a single RGB image. The method can quickly onboard new objects using their 3D models without requiring any object- or task-specific training. In contrast, existing methods typically pre-train on large-scale, task-specific datasets in order to generalize to new objects and to bridge the image-to-model domain gap. We demonstrate that such generalization capabilities can be observed in a recent vision foundation model trained in a self-supervised manner. Specifically, our method estimates the object pose from image-to-model 2D-3D correspondences, which are established by matching patch descriptors from the recent DINOv2 model between the image and pre-rendered object templates. We find that reliable correspondences can be established by kNN matching of patch descriptors from an intermediate DINOv2 layer. Such descriptors carry stronger positional information than descriptors from the last layer, and we show their importance when semantic information is ambiguous due to object symmetries or a lack of texture. To avoid establishing correspondences against all object templates, we develop an efficient template retrieval approach that integrates the patch descriptors into the bag-of-words representation and can promptly propose a handful of similarly looking templates. Additionally, we apply featuremetric alignment to compensate for discrepancies in the 2D-3D correspondences caused by coarse patch sampling. The resulting method noticeably outperforms existing RGB methods for refinement-free pose estimation on the standard BOP benchmark with seven diverse datasets and can be seamlessly combined with an existing render-and-compare refinement method to achieve RGB-only state-of-the-art results. Project page: [evinpinar.github.io/foundpose](https://evinpinar.github.io/foundpose).

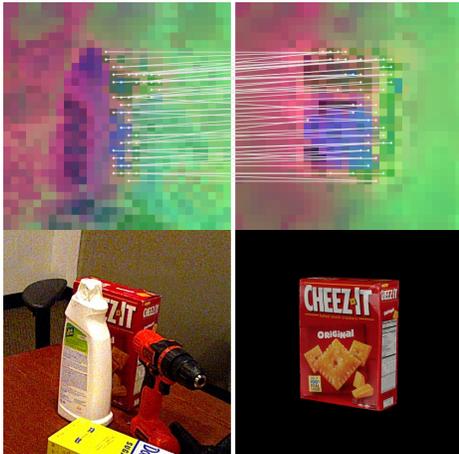
## 1 Introduction

Image-based estimation of the 6D object pose (3D rotation and 3D translation) is an important research problem in the field of spatial AI. In robotics, for example, the information about object poses allows a robot to act upon the objects, which enables fully automated solutions for warehouse operation or assembly. In mixed-reality applications, this information unlocks physical interaction with replicas of real-world objects, such as a computer keyboard, for effective text input when fully immersed.

In this work, we address the problem of model-based 6D pose estimation of *unseen* objects. We assume that 3D models of the objects are available and that budget for onboarding the objects is limited (*e.g.*, not sufficient for rendering a large-scale dataset and training a neural network). This is a practical problem setup for many applications

---

\*Work done during Evin’s internship at Meta.



**Fig. 1. Bridging synthetic-to-real gap.**

Patch descriptors from an intermediate layer of DINOv2 [62] (top), a recent vision foundation model, are the key enabler of FoundPose. Thanks to the generalization capability of these descriptors, it is possible to establish reliable correspondences between a *real query image* (left) and a *synthetic template* (right) by a simple nearest-neighbor matching. The patch descriptors are colored by the top three components of a PCA space computed from descriptors of all object templates. Note that colors of the same object parts are consistent, despite the real-to-synthetic domain gap.

since efficient object onboarding is often a key requirement and 3D object models can be obtained from the manufacturer or readily reconstructed [57, 71, 90].

The very first methods for object pose estimation can, in fact, handle unseen objects as they do not require any training, typically just a set of reference images. These methods rely on classical techniques such as matching hand-crafted image features [14, 49, 69, 72] or template matching [29, 55]. Later, with the rise of machine learning techniques in computer vision, most object pose estimation methods started to rely on deep neural networks. This shift brought a significant improvement in pose estimation accuracy [82] but limited generalization capability as large numbers of training images and a lengthy training process are usually necessary for every new object instance or category. As a result, the majority of these methods focus on a small set of objects. Only recently, with the accuracy scores of seen object pose estimation slowly saturating, the research field started to focus again on unseen objects [56], with the first attempts achieving noticeably lower accuracy scores while being computationally more demanding [12, 34, 41, 75].

With their impressive generalization capabilities, foundation models [7] provide a solid ground for solving the problem at hand. Models such as DINOv2 [15, 62], CLIP [70] or ALIGN [37] have been successfully applied on various vision tasks without any task-specific training [24, 53, 58, 88]. For example, CNOS [58] leverages frozen DINOv2 [15, 62] and Segment Anything [40] and outperforms Mask R-CNN [25] on the object segmentation task. Goodwin *et al.* [24] show that DINO patch descriptors [11] can be used to establish semantic correspondences between instances of the same object category.

Inspired by these success stories, we propose FoundPose, a method for model-based pose estimation of unseen objects, which brings the power of modern foundation features into classical computer vision techniques via careful design choices. Despite being surprisingly simple, easy-to-interpret, and requiring no object- or task-specific training, the method achieves state-of-the-art results on the standard BOP benchmark [34].

First, given an RGB image and an object mask from CNOS [58], we perspectively crop the object region and rapidly retrieve a small set of similarly looking, pre-rendered object templates. To this end, we develop an efficient retrieval approach by integrating DINOv2 into the *bag-of-words* representation from 2003 [76]. This approach is 15X faster while only slightly less accurate than the heavy render-and-compare coarse stage

of MegaPose [41], and requires 100X less templates (several hundreds vs 90K+) than previous approaches [60, 75] (the overall memory footprint is 25X lower).

Second, we establish 2D-2D correspondences between each retrieved *synthetic template* (with fixed light and black background) and the *real image crop* by simple one-way kNN search of DINOv2 patch descriptors (Fig. 1). In contrast, existing methods typically train on large-scale, heavily randomized, and task-specific datasets to bridge the synthetic-to-real domain gap [33, 41, 83]. We demonstrate that patch descriptors from an *intermediate* DINOv2 layer, which were shown to carry stronger positional information [2], are crucial for achieving geometrically consistent correspondences when semantic information is ambiguous due to object symmetries or a lack of texture. We show that the intermediate DINOv2 descriptors are in fact the key enabler of FoundPose, yielding significantly higher accuracy also compared to descriptors extracted with SAM [40], CLIP [70], LoFTR [78], S2DNet [23], and dense SIFT [49]. Next, for each retrieved template, we generate a pose hypothesis from image-to-model 2D-3D correspondences, which are established by lifting the matched 2D patch locations in the template to 3D using rendered depth. Finally, we further optimize the top-quality hypothesis by *featuremetric refinement*, which applies the idea of the classical photometric refinement [4] to DINOv2 patch descriptors. The refinement effectively compensates for the discrepancy in the 2D-3D correspondences caused by coarse sampling of DINOv2 patches.

In summary, we make the following contributions:

1. *A training-free method for model-based object pose estimation* which relies on a surprising simple and easy-to-interpret DINOv2-based pipeline and achieves state-of-the-art results on the standard BOP benchmark [34].
2. *An efficient template retrieval approach* which requires 100X fewer templates than previous approaches and is robust to partial object occlusions.
3. *A lightweight object representation* which is fast to build and has a 25X lower memory footprint than competitors, enabling scaling to large numbers of objects.
4. *A featuremetric refinement approach* which compensates for coarse patch sampling.
5. *Demonstrated importance of intermediate DINOv2 descriptors* for handling symmetric and texture-less objects, also outperforming descriptors from other foundation models.

## 2 Related work

This paper builds on over 60 years of research in object pose estimation and on the recent large-scale vision foundation models.

**Classical methods.** Estimating the 6D pose of rigid objects from a single image is one of the first computer vision problems [72]. Early methods relied on local feature matching [14, 49, 69] or template matching [29, 55], and could rapidly onboard new objects if provided with a set of reference images annotated with model-to-camera transformations. With the introduction of Microsoft Kinect, the attention of the research field was steered towards object pose estimation from RGB-D or D-only images, yielding methods based on 3D local features [85], notably successful point-pair features [21, 32], and methods based on RGB-D template matching [28, 36]. The RGB-D methods produce more accurate poses and are therefore popular in industry, but their application in open-world scenarios is limited. Besides boosting the pose estimation accuracy, the additional depth channel from

Kinect-like sensors enabled easy 3D object reconstruction [57], and, in turn, methods relying on 3D mesh models started to emerge. The model-based object pose estimation setup is still popular [34,82], among both RGB and RGB-D methods, and is relevant for factory and warehouse scenarios where CAD object models are often available. On the other hand, the model-free setup, recently revisited in [26,79], is relevant for mixed reality applications where the set of target objects is typically small and capturing reference images is easy.

**Deep learning methods.** As in other fields of computer vision, methods based on hand-crafted features and techniques have been progressively replaced by methods based on deep neural networks [42,45,51,87,91], which can operate on RGB or RGB-D inputs. These methods represent the current state of the art in terms of accuracy [34,82]. However, their scalability is hindered by the requirement of a large-scale training dataset for learning new objects. To address this issue, deep-learning methods that can onboard new objects without any object-specific training have been proposed recently. As examples of model-based methods, Nguyen *et al.* [60], Shugurov *et al.* (OSOP) [75], and Thalhammer *et al.* [84] learn descriptors for template matching by contrastive learning, Sundermeyer *et al.* [80] generate such descriptors by an augmented auto-encoder, Pitteri *et al.* [67,68] predict generic 3D keypoints or local surface embeddings, and Xiao *et al.* [92] directly predict the 3D object orientation. Model-free methods [26,27,48,79], and methods relying on depth measurements [5,12,61,64] have been also proposed.

The top-performing methods on the unseen object pose estimation task of the BOP Challenge 2023 [34] include GenFlow [54], MegaPose [41], GigaPose [59], FoundationPose [89], SAM-6D [46], and PoMZ [10]. To achieve generalization to novel objects, all except PoMZ require generating millions of task-specific training images showing thousands of different objects. Generating such datasets requires significant effort and opens up new types of challenges, including positioning objects in the scene [16,35], collecting a sufficiently large set of object models, or texturing the models [89]. In contrast, FoundPose does not require any training, uses frozen DINOv2 features, and outperforms RGB-only GenFlow, MegaPose, and GigaPose. FoundationPose, SAM-6D, PoMZ, as well as ZeroPose [12], are RGB-D methods and do not have RGB-only versions. To the best of our knowledge, the only training-free methods for model-based object pose estimation are PoMZ [10], which requires RGB-D inputs, and ZS6D [3], which achieves significantly lower accuracy than FoundPose. ZS6D uses features from the last ViT [18] layer for establishing correspondences, which we show inferior to our solution (Sec. 4). Note that [3,10,12,54,59,89] are all unpublished at the time of submission.

**Foundation models.** A foundation model is a machine-learning model trained on broad data by self-supervised learning that can be adapted to a wide range of downstream tasks [7]. Foundation models initially appeared in natural language processing with examples such as BERT [17] or GPT-3 [9]. In computer vision, foundation models already achieve on-par or better results than supervised models [7,11,13,53,62,63,88]. A prominent example is DINOv2 [62], which is based on the Vision Transformer architecture [18], trained in a self-distillation fashion, and has been shown to encode fine spatial information about the object parts as well as semantic information about object categories [2]. It has been successfully used in zero-shot setups, *i.e.*, without any training, for establishing semantic correspondences [2,24,47,93]. FoundPose builds on these insights. Compared to Goodwin *et al.* [24] which use DINO patch descriptors to establish semantic correspondences within an object category, FoundPose shows that DINOv2 descriptors can be used to establish synthetic-to-real correspondences. Furthermore, [24] requires

RGB-D inputs at test time, cannot handle symmetric objects (such objects are omitted in their evaluation), and is compared only with custom baselines. FoundPose assumes RGB-only test inputs, can handle symmetric objects by design, and achieves state-of-the-art results on the standard BOP benchmark. From the already reviewed methods for pose estimation of specific objects, PoMZ [10] and ZS6D [3] also rely on frozen DINOv1/v2. However, the first needs RGB-D inputs, and the latter achieves noticeably lower accuracy.

### 3 FoundPose

In this section, we describe FoundPose, the proposed method for unseen object pose estimation. We first provide a high-level overview of the method in Sec. 3.1 and then focus on the key components and rationale of our design choices in Sec. 3.2–3.5.

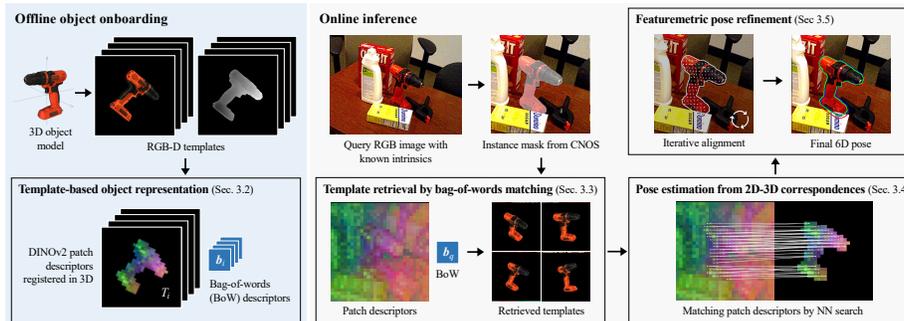
#### 3.1 Method overview

**Problem definition.** We consider the problem of estimating the 6D pose of rigid objects from a single RGB image with known intrinsics. The objective is to estimate the pose of all instances of target objects that are visible in the image. We assume that the only information provided for the target objects are their 3D mesh models and that there is only a limited budget for onboarding the objects, *i.e.*, for preparing object representations that can be used for online pose estimation. We constrain the onboarding process to 5 minutes and 1 GPU, as required by the BOP Challenge 2023 [34]. We additionally assume that segmentation masks of the target object instances, together with per-mask object identity, are provided at inference time. In our experiments, we obtain the masks by CNOS [58], a recent method for segmentation of unseen objects that also requires only 3D models for onboarding the objects.

**Onboarding and inference.** During an offline onboarding stage, we render templates showing 3D object models in different orientations. From each template, we extract DINOv2 descriptors of image patches and register the descriptors in 3D, *i.e.*, each patch descriptor is associated with the corresponding 3D location in the object model space (Sec. 3.2). At inference time, given a segmentation mask of an object instance, we crop the image region around the mask, extract DINOv2 patch descriptors of the crop, and apply a bag-of-words retrieval technique to efficiently identify a small set of templates that show the object in orientations similar to the observation (Sec. 3.3). For each retrieved template, we establish 2D-3D correspondences by matching patch descriptors from the crop against patch descriptors from the template, and generate a pose hypothesis by the PnP-RANSAC algorithm (Sec. 3.4). Finally, we refine the best pose hypothesis by featuremetric alignment, an optimization-based algorithm inspired by photometric alignment that operates on features (Sec. 3.5). The pipeline of the method is shown in Fig. 2.

#### 3.2 Template-based object representation

**Template generation.** Given a texture-mapped 3D object model, we render  $n$  RGB-D templates showing the model under different orientations. The orientations are sampled to uniformly cover the  $SO(3)$  group of 3D rotations [1], and the model is rendered using a standard rasterization technique [74] with a black background and fixed lighting. The



**Fig. 2. FoundPose overview.** During a short onboarding stage, we render RGB-D templates showing the object in different orientations, extract DINOv2 patch descriptors [15, 62] from the RGB channels and register the descriptors in 3D using the depth channel. At inference time, we crop the RGB query image around the object mask predicted by CNOS [58] and retrieve a small set of most similar templates using a bag-of-words approach (with words defined by  $k$ -means clusters of patch descriptors from all templates). For each retrieved template, a pose hypothesis is generated by PnP-RANSAC [22, 43] from 2D-3D correspondences established by matching patch descriptors of the image crop and the template. Finally, the pose hypothesis with the highest number of inlier correspondences is refined by featuremetric alignment.

size of templates is  $S \times S$  pixels, and the objects are rendered such that the longer side of their 2D bounding box is  $\delta S$  pixels long, with  $\delta < 1$ . At inference, we generate crops of the query image with the same size and padding (to allow for errors of segmentation masks around which we crop the image).

**Patch descriptors registered in 3D.** For each RGB-D template with an index  $t \in \{1, \dots, n\}$ , we split the RGB channels into  $m$  non-overlapping patches of  $14 \times 14$  pixels and calculate their patch descriptors  $\{\mathbf{p}_{t,i}\}_{i=1}^m$ . A patch descriptor is calculated as  $\mathbf{p}_{t,i} = \phi_d(\mathbf{p}'_{t,i})$ , where  $\mathbf{p}'_{t,i}$  is the raw patch descriptor extracted by DINOv2 and  $\phi_d: \mathbb{R}^r \mapsto \mathbb{R}^d$  projects the  $r$ -dimensional raw descriptor to the top  $d$  PCA components, which are calculated from valid patch descriptors of all  $n$  templates. A patch is considered valid if its 2D center falls inside the object mask and the PCA-based dimensionality reduction is applied to increase efficiency. Then we represent a template  $t$  by a set  $T_t = \{(\mathbf{p}_{t,j}, \mathbf{x}_j) \mid j \in M\}$ , where  $M$  are indices of valid patches,  $\mathbf{x}_j$  is a 3D location (in the coordinate space of the 3D object model) whose 2D projection is at the center of patch  $j$ . The 3D locations, which are calculated from the depth channel of the template and known camera intrinsics, enable establishing 2D-3D correspondences at inference.

**Bag-of-words descriptors.** At onboarding, we also pre-calculate bag-of-words descriptors of all templates to enable efficient template retrieval at inference using the classical bag-of-words image retrieval technique [66, 76], which mimics text-retrieval systems with the analogy of visual words. Specifically, we define visual words as the centroids of  $k$ -means clusters of patch descriptors extracted from all templates of an object. To calculate the bag-of-words descriptor of a template  $t$ , we assign patch descriptors from the template representation  $T_t$  to the nearest visual words and describe the template by a vector  $\mathbf{b}_t = (b_1, b_2, \dots, b_k)$ . This vector consists of weighted word frequencies defined as  $b_i = (n_{i,t}/n_t) \log(N/n_i)$ , where  $n_{i,t}$  is the number of occurrences of word  $i$  in template  $t$ ,

$n_t$  is the total number of words in template  $t$ , and  $n_i$  is the number of occurrences of word  $i$  in all  $N$  templates. The first term ( $n_{i,t}/n_t$ ) weights words that occur often in a particular template and therefore describe the template well, while the second term ( $\log(N/n_i)$ ) downweights words that occur often in any template. As visual words generated by clustering may suffer from quantization errors, we follow [66, 86] and soft-assign each patch descriptor to several nearest words with weights defined by  $\exp(-d^2/2\sigma^2)$ , where  $d$  is the Euclidean distance of the descriptor from the word and  $\sigma$  is a parameter of the method.

### 3.3 Template retrieval by bag-of-words matching

**Perspective cropping.** At inference, we start by cropping the image region around a given object segmentation mask. To minimize perspective distortion and achieve a crop that resembles a template, we generate the crop by warping the query image to a virtual pinhole camera focused on the segmentation mask. The virtual camera is constructed such that its optical axis passes through the center of the 2D bounding box of the mask, the viewport size is  $S \times S$  pixels, and the longer side of the warped 2D bounding box is  $\delta S$  pixels long.

**Retrieving similar templates.** To retrieve a small set of templates, we calculate the bag-of-words descriptor of the crop (as in Sec. 3.2) and calculate its cosine similarity (*i.e.*, normalized scalar product) with bag-of-words descriptors of all object templates. We select  $h$  templates with the highest cosine similarity, which provide approximate hypotheses on the object orientation for the subsequent pose estimation stage.

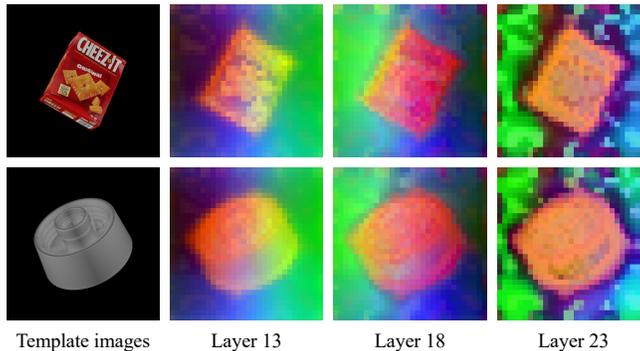
This retrieval technique is efficient and robust to partial occlusions. When an object is partially occluded, its visible part still contributes visual words describing the object. The cosine similarity then normalizes the magnitude and focuses on the direction of the bag-of-words descriptors and is, therefore, less sensitive to the number and more to the type of present words. This robustness has been described in prior work [65, 76] and also in our experiments (see results on LM-O and T-LESS in Sec. 4).

Since the bag-of-words descriptor represents an image as a bag of unordered visual words, the retrieval can be typically improved by re-ranking the results with a spatial verification stage [65]. However, in our case, a similar verification is implicitly done by the subsequent PnP-RANSAC (spatially consistent correspondences are expected to yield a better pose estimate), and bags of words are not actually unordered as the used patch descriptors from an intermediate DINOv2 layer, from which the words are constructed, contains 2D positional information.

### 3.4 Pose estimation from 2D-3D correspondences

**Crop-to-template patch matching.** For each retrieved template  $t$ , we match patch descriptors from the crop to the nearest descriptors from  $T_t$  (in terms of the Euclidean distance), and establish 2D-3D correspondences  $C_t = \{(\mathbf{u}_i, \mathbf{x}_i)\}_{i=1}^m$ , where  $\mathbf{u}_i$  is the 2D center of a query patch and  $\mathbf{x}_i$  is the 3D location associated with the matched patch descriptor from  $T_t$ . The cyclic matching from Goodwin *et al.* [24] did not help in our setup.

Establishing 2D-3D correspondences by crop-to-template patch matching is a considerably simpler problem than exhaustive matching against patch descriptors from all templates, which would be necessary without the template retrieval stage. Moreover, we demonstrate that the template-based approach can effectively handle arbitrary objects,



**Fig. 3. Visualization of DINOv2 patch descriptors.** Shown are top three PCA components of patch descriptors from different layers of DINOv2 ViT-L [15], for a textured object from YCB-V [91] (top) and a symmetric and texture-less object from T-LESS [31] (bottom). As observed in [2] and also clearly visible in these visuals, the patch descriptors contain gradually less positional and more semantic information when going from shallower to deeper layers – the different coloring of object sides (red left vs yellow right) in Layer 13 gradually blends to a solid color (orange) in Layer 23. FoundPose performs the best with descriptors from layer 18, which presumably provides the right information mix. We observed that these descriptors produce geometrically consistent correspondences even on symmetric and texture-less objects – when the semantic information is ambiguous (due to symmetries or a lack of texture), the positional information prioritizes matching patches from the same object side.

including challenging objects with symmetries and without a significant texture. The ambiguity of 2D-3D correspondences, for which such objects are notoriously known [30], is eliminated by (1) restricting the candidate patches to only a single template and (2) using patch descriptors from an intermediate layer of DINOv2 which contain both semantic and 2D positional information [2]. We find that the positional information is crucial for producing geometrically consistent correspondences when the semantic information is not discriminative due to symmetries or a lack of texture (see Fig. 3 and Sec. 4.3).

**Pose fitting.** An object pose  $(\mathbf{R}_t, \mathbf{t}_t)$ , defined by a 3D rotation  $\mathbf{R}_t$  and a 3D translation  $\mathbf{t}_t$  from the model space to the camera space, is estimated for each retrieved template  $t$  from 2D-3D correspondences  $C_t$  by solving the Perspective- $n$ -Point ( $PnP$ ) problem. We solve this problem by the  $EPnP$  algorithm [43] combined with the RANSAC fitting scheme [22] for robustness. In this scheme,  $PnP$  is solved repeatedly on a randomly sampled minimal set of 4 correspondences, and the final output is defined by the pose hypothesis with the highest quality, which we define by the number of inlier correspondences [22]. From the set of  $h$  poses estimated from the  $h$  retrieved templates, the pose with the highest quality is selected as the final coarse pose estimate  $(\mathbf{R}_c, \mathbf{t}_c)$ .

### 3.5 Featuremetric pose refinement

**Reducing 2D-3D discrepancy.** The 2D-3D correspondences are established by linking the 2D centers of matched crop and template patches and lifting the centers of the template patches to 3D using the rendered depth channel. Since the patches are relatively large ( $14 \times 14$  px), the 2D centers of the crop patches may not precisely align

with 2D projections of the corresponding 3D points, even when projected using the ground-truth model-to-crop transformation. To compensate for the potential discrepancy, we refine the pose estimates by aligning template patches to their optimal locations in the image crop by a featuremetric optimization described below.

**Featuremetric alignment.** The coarse pose  $(\mathbf{R}_c, \mathbf{t}_c)$  is refined by Levenberg-Marquardt (L-M) [44,52], an iterative non-linear optimization algorithm. Each L-M iteration updates the pose parameters by a gradient-descent step, minimizing a real-valued cost while adaptively selecting between the first and second-order gradients depending on the cost value. We initialize the optimization with the coarse pose  $(\mathbf{R}_c, \mathbf{t}_c)$  and obtain a refined pose  $(\mathbf{R}_r, \mathbf{t}_r)$  by minimizing the following featuremetric error:

$$(\mathbf{R}_r, \mathbf{t}_r) = \underset{(\mathbf{R}, \mathbf{t})}{\operatorname{argmin}} \sum_{(\mathbf{p}_i, \mathbf{x}_i) \in T_t} \rho\left(\mathbf{p}_i - \mathbf{F}_q\left(\pi(\mathbf{R}\mathbf{x}_i + \mathbf{t})/s\right)\right),$$

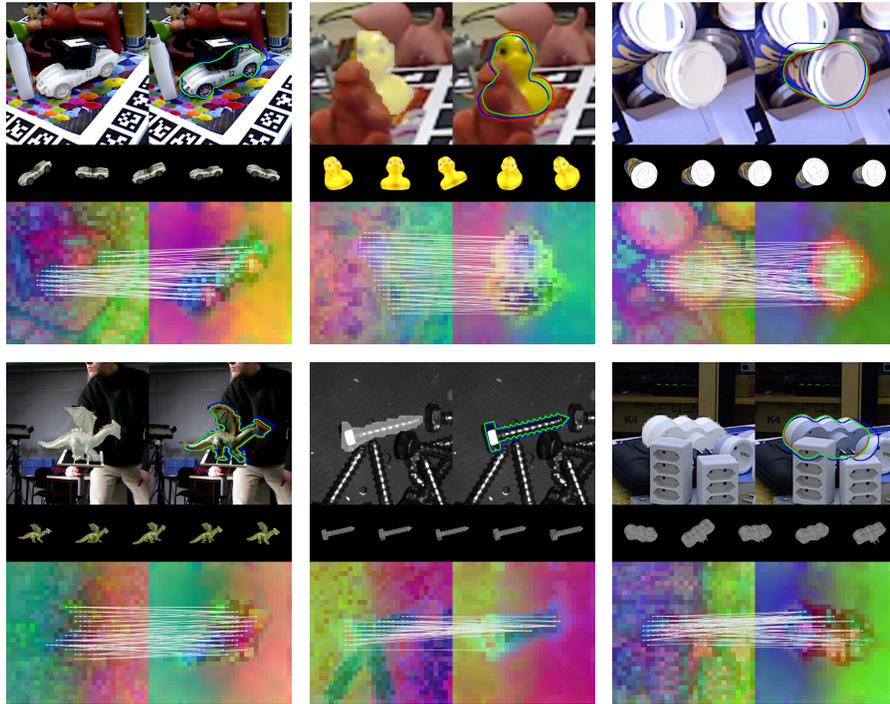
where  $\rho$  is a robust cost function by Barron [6],  $(\mathbf{p}_i, \mathbf{x}_i) \in T_t$  is a descriptor and the corresponding 3D location of patch  $i$  from template  $t$ , and  $\pi: \mathbb{R}^3 \mapsto \mathbb{R}^2$  represents the 2D projection. The feature map  $\mathbf{F}_q \in \mathbb{R}^{a \times a \times d}$  is obtained by stacking the patch descriptors of the query image and is sampled with bilinear interpolation at normalized 2D projections. The spatial resolution of the feature map is  $a \times a$  with  $a = S/s$ , where  $s \times s$  is the patch size. Note that a similar featuremetric alignment was applied in several works [73, 77], typically in combination with features trained specifically for L-M. In our work, we apply the alignment directly on DINOv2 features without any training. Even though there is no guarantee for reaching the global optimum, as in other methods optimizing photo/featuremetric objectives, we observed that a plausible solution is often reached.

## 4 Experiments

In this section, we compare the accuracy and speed of FoundPose with the state-of-the-art methods evaluated on the BOP benchmark [32–34, 82] and present ablation experiments.

### 4.1 Experimental setup

**Evaluation protocol.** We follow the protocol of the BOP Challenge 2019–2023 [33]. In summary, a method is evaluated on the 6D object localization problem, and the error of an estimated pose w.r.t. the ground-truth pose is calculated by three pose-error functions: Visible Surface Discrepancy (VSD) treats ambiguous poses as equivalent by considering only the visible object part, Maximum Symmetry-Aware Surface Distance (MSSD) considers a set of pre-identified global object symmetries and measures the surface deviation in 3D, and Maximum Symmetry-Aware Projection Distance (MSPD) considers the object symmetries and measures the perceivable deviation. An estimated pose is considered correct w.r.t. a pose-error function  $e$ , if  $e < \theta_e$ , where  $e \in \{\text{VSD}, \text{MSSD}, \text{MSPD}\}$  and  $\theta_e$  is the threshold of correctness. The fraction of annotated object instances for which a correct pose is estimated is referred to as Recall. The Average Recall w.r.t. a function  $e$ , denoted as  $\text{AR}_e$ , is defined as the average of the Recall rates calculated for multiple settings of the threshold  $\theta_e$  and also for multiple settings of a misalignment tolerance  $\tau$  in the case of VSD. The overall accuracy of a method is measured by the Average Recall:  $\text{AR} = (\text{AR}_{\text{VSD}} + \text{AR}_{\text{MSSD}} + \text{AR}_{\text{MSPD}})/3$ .



**Fig. 4. Example FoundPose results** on HB, LM-O, IC-BIN, TUD-L, ITODD and T-LESS datasets, showing that our method can handle a broad range of objects, including textured, texture-less and symmetric ones. Each example shows the query image crop with the CNOS mask in white (top left), retrieved templates (middle row), matched patch descriptors of the crop and the template that led to the top-quality pose estimate (bottom row), and the contour of the ground-truth pose in red, the coarse pose in blue, and the refined pose in green (top right).

**Datasets.** The experiments are conducted on the seven core BOP datasets: LM-O [8], T-LESS [31], ITODD [20], HB [39], YCB-V [91], IC-BIN [19], and TUD-L [32]. The datasets feature 108 diverse objects ranging from texture-less and symmetric industrial objects (ITODD, T-LESS) to typical household objects. The images show scenes whose complexity varies from simple scenes with several isolated objects to challenging ones with multiple object instances and a high amount of clutter and occlusion. Only 3D object models and test images from these datasets were used for experiments with FoundPose, not the provided synthetic nor real training images since no training is required.

**Compared methods.** FoundPose is compared against model-based RGB methods evaluated on the unseen object pose estimation task of the BOP Challenge 2023 [34]: GenFlow [54], MegaPose [41], GigaPose [59], and also against ZS6D [3] and OSOP [75]. Except for OSOP, all of these methods (including FoundPose) use the same segmentation masks that were produced by CNOS [58] and provided to the challenge participants. OSOP relies on a custom detector of unseen objects for LM-O, HB, and YCB-V, and on Mask R-CNN [25] trained for specific objects for T-LESS (hence we do not include the T-LESS result). Besides variants of FoundPose where the coarse poses are refined

by the featuremetric refinement (Sec. 3.5), we evaluate variants with poses refined by 5 iterations of the MegaPose refiner (*i.e.*, the last stage of the MegaPose pipeline [41]).

**Implementation details.** Unless stated otherwise, we use the following parameter settings in the presented experiments. We rendered 800 templates per object with approximately  $25^\circ$  angle between depicted object orientations. We set the size of templates and of the query image crop to  $420 \times 420$  px with  $\delta=0.6$ . With the patch size of  $14 \times 14$  px (for which DINOv2 is trained), we extract  $30 \times 30$  patch descriptors from each template/crop and reduce their dimensionality by projecting them to the top 256 PCA components. We use the output tokens from layer 18 of DINOv2 ViT-L/14 with registers [15] as the patch descriptors. Visual words for the bag-of-words template retrieval are defined per object by the centroids of 2048  $k$ -means clusters of patch descriptors from all templates of the object. The bag-of-words descriptors are constructed by soft-assigning each patch descriptor to 3 nearest words with  $\sigma=10$ . For each query image crop, we retrieve 5 templates, and estimate the pose from 2D-3D correspondences (established between the query image crop and the template) by  $PnP$ -RANSAC running for up to 400 iterations with the inlier threshold set to 10 px. The featuremetric refinement is applied to the best coarse pose and runs until convergence for up to 30 iterations, with the Barron loss [6] parameters set to  $\alpha=-5$  and  $c=0.5$ . By default, the evaluated FoundPose variants use  $n$  CNOS masks per object, where  $n$  is the number of object instances to localize (provided as input in the 6D object localization task in BOP). The only exceptions are variants in the bottom part of Tab. 1, which use  $5n$  CNOS masks per object. Note that all masks were loaded from files with default CNOS masks, which were provided for BOP 2023 [34] and contain multiple masks per object instance. The number of CNOS masks used by other methods is unknown.

## 4.2 Main results

**Accuracy.** Among methods that do not apply any refinement stage, FoundPose (without the featuremetric refinement) produces significantly more accurate poses than the competitors, achieving +10, +14, and +16 AR on the seven BOP datasets on average compared to the coarse versions of GigaPose [59], GenFlow [54], and MegaPose [41] (rows 1–4 in Tab. 1). The featuremetric refinement brings an extra improvement of +5 AR on average (rows 1 vs 7). At an additional computational cost, a large improvement of +17 AR (rows 1 vs 8) can be achieved if the coarse poses from FoundPose are refined by the iterative render-and-compare approach from MegaPose, which is trained on 2M+ synthetic images of diverse objects and proven remarkably effective. When initiated with coarse poses from FoundPose, the MegaPose refiner achieves +4 higher AR score than when initiated with poses from the original coarse pose estimation stage of MegaPose (rows 8 vs 11). Further improvements at further computational cost can be achieved if the refinement is applied to multiple pose hypotheses and the top refined pose is reported as the final estimate (rows 12–17). We achieve the overall best average AR score of 59.6 AR when top 5 pose hypotheses (generated from 5 retrieved templates) are optimized with the featuremetric refinement followed by the MegaPose refinement. On both the single and the multi hypotheses setups, combining the two refinement approaches achieves the best scores (rows 9 and 14), suggesting their complementarity. This entry outperforms multi-hypotheses versions of MegaPose and GenFlow (rows 16 and 17), which are the top-performing RGB methods from the BOP Challenge 2023 [34], as well as GigaPose [59] with the MegaPose refinement (row 15).

#	Method	Pose refinement	No train.	LM-O	T-LESS	TUD-L	IC-BIN	ITODD	HB	YCB-V	Average	Time
<i>Coarse pose estimation:</i>												
1	FoundPose	–	✓	<b>39.6</b>	<b>33.8</b>	<b>46.7</b>	<b>23.9</b>	<b>20.4</b>	<b>50.8</b>	<b>45.2</b>	<b>37.2</b>	1.7
2	GigaPose [59]	–	✗	29.9	<u>27.3</u>	<u>30.2</u>	<u>23.1</u>	<u>18.8</u>	34.8	29.0	<u>27.6</u>	<b>0.9</b>
3	GenFlow [54]	–	✗	25.0	21.5	30.0	16.8	15.4	28.3	27.7	23.5	3.8
4	MegaPose [41]	–	✗	22.9	17.7	25.8	15.2	10.8	25.1	28.1	20.8	15.5
5	OSOP [75]	–	✗	<u>31.2</u>	–	–	–	–	<u>49.2</u>	<u>33.2</u>	–	–
6	ZS6D [3]	–	✓	29.8	21.0	–	–	–	–	32.4	–	–
<i>With pose refinement (a single hypothesis):</i>												
7	FoundPose	Featuremetric	✓	39.5	39.6	56.7	28.3	26.2	58.5	49.7	42.6	<u>2.6</u>
8	FoundPose	MegaPose	✗	55.4	<u>51.0</u>	<u>63.3</u>	43.0	34.6	<u>69.5</u>	<b>66.1</b>	<u>54.7</u>	4.4
9	FoundPose	Feat. + MegaPose	✗	<b>55.7</b>	<u>51.0</u>	<u>63.3</u>	<u>43.3</u>	<u>35.7</u>	<b>69.7</b>	<b>66.1</b>	<b>55.0</b>	6.4
10	GigaPose [59]	MegaPose	✗	55.6	<b>54.6</b>	57.8	<b>44.3</b>	<b>37.8</b>	69.3	63.4	<u>54.7</u>	<b>2.4</b>
11	MegaPose [41]	MegaPose	✗	49.9	47.7	<b>65.3</b>	36.7	31.5	65.4	60.1	50.9	31.7
<i>With pose refinement (5 hypotheses):</i>												
12	FoundPose	Featuremetric	✓	42.0	43.6	60.2	30.5	27.3	53.7	51.3	44.1	<u>7.4</u>
13	FoundPose	MegaPose	✗	58.6	<u>54.9</u>	65.7	44.4	36.1	70.3	<u>67.3</u>	56.8	11.2
14	FoundPose	Feat. + MegaPose	✗	<b>61.0</b>	<b>57.0</b>	<b>69.4</b>	<b>47.9</b>	<b>40.7</b>	<u>72.3</u>	<b>69.0</b>	<b>59.6</b>	20.5
15	GigaPose [59]	MegaPose	✗	59.9	<b>57.0</b>	64.5	46.7	39.7	72.2	66.3	<u>57.9</u>	<b>7.3</b>
16	GenFlow [54]	GenFlow	✗	56.3	52.3	<u>68.4</u>	45.3	39.5	<b>73.9</b>	63.3	57.1	20.9
17	MegaPose [41]	MegaPose	✗	56.0	50.7	<u>68.4</u>	41.4	33.8	70.4	62.1	54.7	47.4

**Table 1. Performance on the seven core BOP datasets [82].** The table shows Average Recall (AR) scores per dataset, the average AR score, and the time to estimate poses of all objects in an image averaged over the datasets (in seconds). Methods for coarse pose estimation (without applying any refinement) are in the upper part, methods applying a refinement stage on a single pose hypothesis are in the middle, and methods refining multiple pose hypotheses and reporting the top refined pose (with the highest refinement score) are at the bottom. Methods that do not undergo any task-specific training are marked with ✓.

**Speed.** The presented variants of FoundPose are the most accurate and the second-fastest entries in all three categories evaluated in Tab. 1.<sup>12</sup> Notably, the coarse pose estimation stage from FoundPose is significantly faster (1.7 vs. 15.5 s) and more accurate than the coarse render-and-compare stage from MegaPose (rows 1 vs. 4). The presented FoundPose variants provide a spectrum of trade-offs between speed and accuracy, with the faster variants (rows 1 and 7) being relevant for online applications such as robotic manipulation, while the more accurate variants being relevant for offline applications such as ground-truth object pose annotation for training supervised methods [82]. The trade-off can be further controlled by, *e.g.*, the template size or different types of the DINOv2 backbone – rows 3–4 in Tab. 2 show that FoundPose runs at around 1.3 s per image if the ViT-L backbone (used for results in Tab. 1) is replaced with the smaller ViT-S. Note that the reported times also include the CNOS segmentation stage that takes around 0.3 s per image. The offline object onboarding takes less than 5 minutes, as requested by BOP [34].

<sup>1</sup> The time of GigaPose [59], the only faster method, was obtained with a more powerful GPU (V100 48GB vs P100 16GB used in our experiments).

<sup>2</sup> OSOP [75] has a light version that runs at < 1 s per image, but has lower accuracy scores (-3 AR) than the entry in Tab. 1, which takes 5–12 s per image.

#	Method	LM-O	T-LESS	TUD-L	IC-BIN	ITODD	HB	YCB-V	Average	Time
<i>Backbones for extracting patch descriptors:</i>										
1	DINOv2 ViT-L – layer 18	39.6	33.8	46.7	23.9	20.4	50.8	45.2	37.2	1.7
2	DINOv2 ViT-L – layer 23	23.2	22.8	31.2	10.3	9.7	33.0	34.0	23.5	1.5
3	DINOv2 ViT-S – layer 9	34.0	31.6	42.7	21.7	16.8	46.8	44.7	34.0	1.3
4	DINOv2 ViT-S – layer 11	22.8	24.2	29.8	11.9	10.5	30.4	36.4	23.7	1.3
5	SAM ViT-L [40] – layer 23	2.2	12.8	9.2	7.5	6.0	10.6	26.9	10.7	3.4
6	DenseSIFT – step size 7px	3.2	2.6	6.5	10.5	2.9	5.6	22.2	7.6	1.4
7	S2DNet [23]	0.8	1.2	0.8	1.4	1.2	1.2	1.3	1.1	1.8
<i>Template retrieval by matching c1s token from layer 18 of DINOv2 ViT-L:</i>										
8	Retrieval by c1s token	19.9	17.8	24.6	10.3	13.6	17.7	23.6	18.2	1.6
9	Retrieval by c1s token with black bg.	25.5	26.2	30.3	16.7	13.6	29.3	34.4	25.1	1.6
<i>Other ablations:</i>										
10	Pose given by the top matched template	20.3	18.5	23.0	12.8	12.4	19.6	17.6	17.7	1.0
11	Ground-truth instead of CNOS masks	45.6	53.1	57.1	30.6	–	–	50.9	–	–

**Table 2. Ablation experiments.** We analyze different backbones for extracting patch descriptors (rows 1–7), compare our bag-of-words template retrieval with alternative approaches based on matching the `c1s` token (rows 1 vs 8 and 9), evaluate our retrieval approach by considering the pose associated with the top template as the final pose estimate (row 10), and show the accuracy of coarse poses from FoundPose when ground-truth masks are used instead of the CNOS masks [58] (row 11; ground-truth annotations for ITODD and HB are not publicly available).

### 4.3 Ablation experiments

**Feature extractors.** Tab. 2 (rows 1–7) evaluates the performance of FoundPose (without the featuremetric refinement) with different patch descriptors. We achieve the best pose accuracy when we use the output tokens of layer 18 (out of 23) from the DINOv2 ViT-L model with registers [15] as the patch descriptors. We also observed qualitatively that this layer provides a good balance between the positional and semantic information – the resulting correspondences are more geometrically consistent than correspondences obtained using later layers (Fig. 3). Furthermore, we evaluate FoundPose with patch descriptors extracted from layer 23 of SAM ViT-L [40], from the last backbone layers of the feature matching pipelines LoFTR [78] and S2DNet [23], from the last layer of the CLIP image encoder [70], and with patch descriptors defined by the classical SIFT descriptor [50] calculated on a regular 2D grid with the cell size of 7 px. In the case of S2DNet, descriptors were obtained by sampling the last CNN feature map. All method parameters were fixed in these experiments, only the patch descriptors differed. As shown in Tab. 2, patch descriptors from DINOv2 are the key enabler of FoundPose, yielding significantly higher accuracy than the alternatives (we only show the performance of the better alternatives).

**Template retrieval.** Next, on rows 8–9 in Tab. 2, we evaluate the proposed bag-of-words template retrieval approach against an alternative based on matching the `c1s` token from DINOv2, which is used in CNOS [58] for object identification. In this alternative, we use the `c1s` token from layer 18 of DINOv2 ViT-L as the template and crop descriptor and retrieve templates whose descriptors have the highest cosine similarity with the crop descriptor. Compared to the bag-of-words descriptor constructed from patch descriptors, we observed that the `c1s` token is less robust to occlusions and contains limited

information about the object pose (as also shown in Fig. 5 of [58]). To avoid the influence of the background on the `cls` token, we additionally evaluate a variant where pixels outside the object mask are made black, as in [3, 58]. As shown on rows 8 vs. 9 in Tab. 2, this modification improves the performance of the `cls`-based approach but is still far from the performance of our bag-of-words approach (row 1). When tried with the `cls` token from layer 23, the alternative retrieval approach achieved even lower accuracy.

We further analyze the quality of templates retrieved by the bag-of-words approach by evaluating the object pose associated with the top retrieved template. The pose from the template is adjusted such that the 2D bounding circle of the object is aligned with the 2D bounding circle of the object segmentation mask in the query image. This approach is surprisingly effective, reaching accuracy close to the coarse poses from MegaPose while being 15 times faster (row 10 in Tab. 2 vs row 4 in Tab. 1).

Another possible retrieval approach is to directly match patches of each query-template pair, and calculate the pair similarity from the matching quality, *e.g.*, as done by Goodwin *et al.* [24]. This approach needs a kNN search for each pair and is noticeably less efficient than our approach (0.64s vs 0.0008s per query; measured with `faiss` [38]).

**Ground-truth segmentation masks.** A common source of failure cases are erroneous segmentation masks. When ground-truth masks are used instead of masks predicted by CNOS [58], FoundPose achieves a large +6–19 AR improvement (rows 1 vs 11 in Tab. 2).

**Effect of other parameters.** We also evaluated FoundPose with different numbers of templates (400, 800, 1600), PCA components (128, 256, 512, 1024), visual words (1024, 2048, 4096), and RANSAC iterations (200, 400, 800). These settings influence memory requirements but lead to only marginal differences in speed and accuracy.

**Memory requirements.** With around 300 valid patches per template, patch descriptors projected to a  $256d$  PCA space, and 4B per scalar, the patch descriptors take 300kB per template. With 2048 visual words, the bag-of-words descriptor takes an additional 8kB per template. In total, an object representation built from 800 templates takes 234MB. Compared to previous template-based approaches, this is *25X less memory* than required by OSOP [75] and Nguyen *et al.* [60], which both need 5.6GB per object (16384d vector for each of 92232 templates). With a sub 1 AR drop, the memory required by FoundPose can be reduced to 59MB with 400 templates,  $128d$  PCA space, and 1024 visual words. The template-based method of Sundermeyer *et al.* [81] requires 45MB per object, but its accuracy is -13.1 AR lower compared to FoundPose (see [81] for details). The concurrent works GigaPose [59] and ZS6D [3] have similar per-template memory requirements as FoundPose, use 162 and 300 templates, respectively but achieve lower accuracy (Tab. 1).

## 5 Conclusion

We have proposed an RGB method for model-based pose estimation of unseen objects, which significantly outperforms existing methods on the standard BOP benchmark. We believe that achieving this without any object- nor task-specific training, just with a frozen vision foundation model, is an important and non-obvious outcome. Furthermore, we have shown that the method can be seamlessly combined with an existing render-and-compare refinement approach to achieve RGB-only state-of-the-art results. Our strong results are encouraging to revisit efficient classical computer vision which is often overlooked in the modern literature.

## References

1. Alexa, M.: Super-Fibonacci Spirals: Fast, low-discrepancy sampling of  $SO(3)$ . *CVPR (2022)* [5](#)
2. Amir, S., Gandselman, Y., Bagon, S., Dekel, T.: Deep ViT features as dense visual descriptors. *ECCVW (2022)* [3, 4, 8](#)
3. Ausserlechner, P., Habegger, D., Thalhammer, S., Weibel, J.B., Vincze, M.: ZS6D: Zero-shot 6D object pose estimation using vision transformers. *arXiv preprint arXiv:2309.11986 (2023)* [4, 5, 10, 12, 14](#)
4. Baker, S., Matthews, I.: Lucas-Kanade 20 years on: A unifying framework. *IJCV (2004)* [3](#)
5. Balntas, V., Doumanoglou, A., Sahin, C., Sock, J., Kouskouridas, R., Kim, T.K.: Pose guided RGBD feature learning for 3D object pose estimation. *ICCV (2017)* [4](#)
6. Barron, J.T.: A general and adaptive robust loss function. *CVPR (2019)* [9, 11](#)
7. Bommasani, R., Hudson, D.A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M.S., Bohg, J., Bosselut, A., Brunskill, E., et al.: On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258 (2021)* [2, 4](#)
8. Brachmann, E., Krull, A., Michel, F., Gumhold, S., Shotton, J., Rother, C.: Learning 6D object pose estimation using 3D object coordinates. *ECCV (2014)* [10](#)
9. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. *NeurIPS (2020)* [4](#)
10. Caraffa, A., Boscaini, D., Hamza, A., Poiesi, F.: Object 6D pose estimation meets zero-shot learning (2023) [4, 5](#)
11. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. *ICCV (2021)* [2, 4](#)
12. Chen, J., Sun, M., Bao, T., Zhao, R., Wu, L., He, Z.: ZeroPose: CAD-model-based zero-shot pose estimation. *arXiv preprint arXiv:2305.17934 (2023)* [2, 4](#)
13. Cherti, M., Beaumont, R., Wightman, R., Wortsman, M., Ilharco, G., Gordon, C., Schuhmann, C., Schmidt, L., Jitsev, J.: Reproducible scaling laws for contrastive language-image learning. *CVPR (2023)* [4](#)
14. Collet, A., Martinez, M., Srinivasa, S.S.: The MOPED framework: Object recognition and pose estimation for manipulation. *IJRR (2011)* [2, 3](#)
15. Darcet, T., Oquab, M., Mairal, J., Bojanowski, P.: Vision transformers need registers. *arXiv preprint arXiv:2309.16588 (2023)* [2, 6, 8, 11, 13](#)
16. Denninger, M., Sundermeyer, M., Winkelbauer, D., Olefir, D., Hodan, T., Zidan, Y., Elbadrawy, M., Knauer, M., Katam, H., Lodhi, A.: BlenderProc: reducing the reality gap with photorealistic rendering. *Robotics: Science and Systems (RSS) Workshops (2020)* [4](#)
17. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. *ACL (2019)* [4](#)
18. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR (2021)* [4](#)
19. Doumanoglou, A., Kouskouridas, R., Malassiotis, S., Kim, T.K.: Recovering 6D object pose and predicting next-best-view in the crowd. *CVPR (2016)* [10](#)
20. Drost, B., Ulrich, M., Bergmann, P., Hartinger, P., Steger, C.: Introducing MVTEC ITODD – A dataset for 3D object recognition in industry. *ICCVW (2017)* [10](#)
21. Drost, B., Ulrich, M., Navab, N., Ilic, S.: Model globally, match locally: Efficient and robust 3D object recognition. *CVPR (2010)* [3](#)
22. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM (1981)* [6, 8](#)
23. Germain, H., Bourmaud, G., Lepetit, V.: S2DNet: Learning accurate correspondences for sparse-to-dense feature matching. *ECCV (2020)* [3, 13](#)

24. Goodwin, W., Vaze, S., Havoutis, I., Posner, I.: Zero-shot category-level object pose estimation. *ECCV (2022)* [2](#), [4](#), [7](#), [14](#)
25. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. *ICCV (2017)* [2](#), [10](#)
26. He, X., Sun, J., Wang, Y., Huang, D., Bao, H., Zhou, X.: OnePose++: Keypoint-free one-shot object pose estimation without CAD models. *NeurIPS (2022)* [4](#)
27. He, Y., Wang, Y., Fan, H., Sun, J., Chen, Q.: FS6D: Few-shot 6D pose estimation of novel objects. *CVPR (2022)* [4](#)
28. Hinterstoisser, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G., Konolige, K., Navab, N.: Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes. *ACCV (2012)* [3](#)
29. Hinterstoisser, S., Lepetit, V., Ilic, S., Fua, P., Navab, N.: Dominant orientation templates for real-time detection of texture-less objects. *CVPR (2010)* [2](#), [3](#)
30. Hodan, T., Barath, D., Matas, J.: EPOS: Estimating 6D pose of objects with symmetries. *CVPR (2020)* [8](#)
31. Hodan, T., Haluza, P., Obdrzalek, S., Matas, J., Lourakis, M., Zabulis, X.: T-LESS: An RGB-D dataset for 6D pose estimation of texture-less objects. *WACV (2017)* [8](#), [10](#)
32. Hodan, T., Michel, F., Brachmann, E., Kehl, W., Glent Buch, A., Kraft, D., Drost, B., Vidal, J., Ihrke, S., Zabulis, X., Sahin, C., Manhardt, F., Tombari, F., Kim, T.K., Matas, J., Rother, C.: BOP: Benchmark for 6D object pose estimation. *ECCV (2018)* [3](#), [9](#), [10](#)
33. Hodan, T., Sundermeyer, M., Drost, B., Labbé, Y., Brachmann, E., Michel, F., Rother, C., Matas, J.: BOP challenge 2020 on 6D object localization. *ECCVW (2020)* [3](#), [9](#)
34. Hodan, T., Sundermeyer, M., Labbe, Y., Wang, G., Brachmann, E., Drost, B., Rother, C., Matas, J.: BOP challenge 2023 on detection, segmentation and pose estimation of unseen rigid objects. To be published. (2023), <https://bop.felk.cvut.cz/leaderboards/pose-estimation-unseen-bop23/core-datasets/>, the results are available at [bop.felk.cvut.cz](https://bop.felk.cvut.cz/). [2](#), [3](#), [4](#), [5](#), [9](#), [10](#), [11](#), [12](#)
35. Hodaň, T., Vineet, V., Gal, R., Shalev, E., Hanzelka, J., Connell, T., Urbina, P., Sinha, S., Guenter, B.: Photorealistic image synthesis for object instance detection. *IEEE International Conference on Image Processing (ICIP) (2019)* [4](#)
36. Hodaň, T., Zabulis, X., Lourakis, M., Obdrzálék, Š., Matas, J.: Detection and fine 3d pose estimation of texture-less objects in RGB-D images. *IEEE International Conference on Intelligent Robots and Systems (IROS) (2015)* [3](#)
37. Jia, C., Yang, Y., Xia, Y., Chen, Y.T., Parekh, Z., Pham, H., Le, Q.V., Sung, Y., Li, Z., Duerig, T.: Scaling up visual and vision-language representation learning with noisy text supervision. *ICML (2021)* [2](#)
38. Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data (2019)* [14](#)
39. Kaskman, R., Zakharov, S., Shugurov, I., Ilic, S.: HomebrewedDB: RGB-D dataset for 6D pose estimation of 3D objects. *ICCVW (2019)* [10](#)
40. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollár, P., Girshick, R.: Segment anything. *ICCV (2023)* [2](#), [3](#), [13](#)
41. Labbé, Y., Manuelli, L., Mousavian, A., Tyree, S., Birchfield, S., Tremblay, J., Carpentier, J., Aubry, M., Fox, D., Sivic, J.: MegaPose: 6D pose estimation of novel objects via render & compare. *CoRL (2022)* [2](#), [3](#), [4](#), [10](#), [11](#), [12](#)
42. Labbé, Y., Carpentier, J., Aubry, M., Sivic, J.: CosyPose: Consistent multi-view multi-object 6D pose estimation. *ECCV (2020)* [4](#)
43. Lepetit, V., Moreno-Noguer, F., Fua, P.: EPnP: An accurate O(n) solution to the PnP problem. *IJCV (2009)* [6](#), [8](#)
44. Levenberg, K.: A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics (1944)* [9](#)
45. Li, Y., Wang, G., Ji, X., Xiang, Y., Fox, D.: DeepIM: Deep iterative matching for 6D pose estimation. *IJCV (2020)* [4](#)

46. Lin, J., Liu, L., Lu, D., Jia, K.: SAM-6D: Segment anything model meets zero-shot 6d object pose estimation. arXiv preprint arXiv:2311.15707 (2023) [4](#)
47. Liu, Y., Zhu, M., Li, H., Chen, H., Wang, X., Shen, C.: Matcher: Segment anything with one shot using all-purpose feature matching. arXiv preprint arXiv:2305.13310 (2023) [4](#)
48. Liu, Y., Wen, Y., Peng, S., Lin, C., Long, X., Komura, T., Wang, W.: Gen6D: Generalizable model-free 6-DoF object pose estimation from RGB images. ECCV (2022) [4](#)
49. Lowe, D.G.: Object recognition from local scale-invariant features. ICCV (1999) [2](#), [3](#)
50. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV (2004) [13](#)
51. Manhardt, F., Kehl, W., Navab, N., Tombari, F.: Deep Model-Based 6D Pose Refinement in RGB. ECCV (2018) [4](#)
52. Marquardt, D.W.: An algorithm for least-squares estimation of nonlinear parameters. Journal of the Society for Industrial and Applied Mathematics (1963) [9](#)
53. Melas-Kyriazi, L., Rupprecht, C., Laina, I., Vedaldi, A.: Deep Spectral Methods: A surprisingly strong baseline for unsupervised semantic segmentation and localization. CVPR (2022) [2](#), [4](#)
54. Moon, S., Son, H.: GenFlow, a submission to the BOP Challenge 2023 (bop.felk.cvut.cz). Unpublished (2023) [4](#), [10](#), [11](#), [12](#)
55. Murase, H., Nayar, S.K.: Visual learning and recognition of 3-D objects from appearance. IJCV (1995) [2](#), [3](#)
56. Naeem, M.F., Örnek, E.P., Xian, Y., Van Gool, L., Tombari, F.: 3D compositional zero-shot learning with decompositional consensus. In: ECCV (2022) [2](#)
57. Newcombe, R.A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A.J., Kohi, P., Shotton, J., Hodges, S., Fitzgibbon, A.: KinectFusion: Real-time dense surface mapping and tracking. ISMAR (2011) [2](#), [4](#)
58. Nguyen, V.N., Groueix, T., Ponimatkin, G., Lepetit, V., Hodan, T.: CNOS: A strong baseline for CAD-based novel object segmentation. ICCVW (2023) [2](#), [5](#), [6](#), [10](#), [13](#), [14](#)
59. Nguyen, V.N., Groueix, T., Salzmann, M., Lepetit, V.: GigaPose: Fast and robust novel object pose estimation via one correspondence. In: CVPR (2024) [4](#), [10](#), [11](#), [12](#), [14](#)
60. Nguyen, V.N., Hu, Y., Xiao, Y., Salzmann, M., Lepetit, V.: Templates for 3D object pose estimation revisited: Generalization to new objects and robustness to occlusions. CVPR (2022) [3](#), [4](#), [14](#)
61. Okorn, B., Gu, Q., Hebert, M., Held, D.: ZePHyR: Zero-shot pose hypothesis rating. ICRA (2021) [4](#)
62. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: DINOv2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023) [2](#), [4](#), [6](#)
63. Örnek, E.P., Krishnan, A.K., Gayaka, S., Kuo, C.H., Sen, A., Navab, N., Tombari, F.: SupeRGB-D: Zero-shot instance segmentation in cluttered indoor environments. IEEE RA-L (2023) [4](#)
64. Park, K., Mousavian, A., Xiang, Y., Fox, D.: LatentFusion: End-to-end differentiable reconstruction and rendering for unseen object pose estimation. CVPR (2020) [4](#)
65. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. CVPR (2007) [7](#)
66. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Lost in quantization: Improving particular object retrieval in large scale image databases. CVPR (2008) [6](#), [7](#)
67. Pitteri, G., Bugeau, A., Ilic, S., Lepetit, V.: 3D object detection and pose estimation of unseen objects in color images with local surface embeddings. ACCV (2020) [4](#)
68. Pitteri, G., Ilic, S., Lepetit, V.: CorNet: Generic 3D corners for 6D pose estimation of new objects without retraining. ICCVW (2019) [4](#)
69. Ponce, J., Lazebnik, S., Rothganger, F., Schmid, C.: Toward true 3D object recognition. In: Reconnaissance de Formes et Intelligence Artificielle (2004) [2](#), [3](#)
70. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. ICML (2021) [2](#), [3](#), [13](#)

71. Reizenstein, J., Shapovalov, R., Henzler, P., Sbordone, L., Labatut, P., Novotny, D.: Common objects in 3D: Large-scale learning and evaluation of real-life 3D category reconstruction. *ICCV* (2021) [2](#)
72. Roberts, L.G.: Machine perception of three-dimensional solids. Ph.D. thesis, Massachusetts Institute of Technology (1963) [2](#), [3](#)
73. Sarlin, P.E., Unagar, A., Larsson, M., Germain, H., Toft, C., Larsson, V., Pollefeys, M., Lepetit, V., Hammarstrand, L., Kahl, F., Sattler, T.: Back to the feature: Learning robust camera localization from pixels to pose. *CVPR* (2021) [9](#)
74. Shreiner, D.: OpenGL programming guide: the official guide to learning OpenGL, versions 3.0 and 3.1. Pearson Education (2009) [5](#)
75. Shugurov, I., Li, F., Busam, B., Ilic, S.: OSOP: A multi-stage one shot object pose estimation framework. *CVPR* (2022) [2](#), [3](#), [4](#), [10](#), [12](#), [14](#)
76. Sivic, Zisserman: Video Google: A text retrieval approach to object matching in videos. *ICCV* (2003) [2](#), [6](#), [7](#)
77. von Stumberg, L., Wenzel, P., Yang, N., Cremers, D.: LM-Reloc: Levenberg-Marquardt based direct visual relocalization. *3DV* (2020) [9](#)
78. Sun, J., Shen, Z., Wang, Y., Bao, H., Zhou, X.: LoFTR: Detector-free local feature matching with transformers. *CVPR* (2021) [3](#), [13](#)
79. Sun, J., Wang, Z., Zhang, S., He, X., Zhao, H., Zhang, G., Zhou, X.: OnePose: One-shot object pose estimation without CAD models. *CVPR* (2022) [4](#)
80. Sundermeyer, M., Durner, M., Puang, E.Y., Marton, Z.C., Vaskevicius, N., Arras, K.O., Triebel, R.: Multi-path learning for object pose estimation across domains. *CVPR* (2020) [4](#)
81. Sundermeyer, M., Durner, M., Puang, E.Y., Marton, Z.C., Vaskevicius, N., Kai, O.A., Triebel, R.: Multi-path learning for object pose estimation across domains. *CVPR* (2020) [14](#)
82. Sundermeyer, M., Hodan, T., Labbé, Y., Wang, G., Brachmann, E., Drost, B., Rother, C., Matas, J.: BOP challenge 2022 on detection, segmentation and pose estimation of specific rigid objects. *CVPRW* (2023) [2](#), [4](#), [9](#), [12](#)
83. Sundermeyer, M., Marton, Z.C., Durner, M., Brucker, M., Triebel, R.: Implicit 3D orientation learning for 6D object detection from rgb images. *ECCV* (2018) [3](#)
84. Thalhammer, S., Weibel, J.B., Vincze, M., Garcia-Rodriguez, J.: Self-supervised vision transformers for 3D pose estimation of novel objects. *arXiv preprint arXiv:2306.00129* (2023) [4](#)
85. Tombari, F., Salti, S., Di Stefano, L.: Performance evaluation of 3D keypoint detectors. *IJCV* (2013) [3](#)
86. Torii, A., Sivic, J., Pajdla, T., Okutomi, M.: Visual place recognition with repetitive structures. *CVPR* (2013) [7](#)
87. Wang, G., Manhardt, F., Tombari, F., Ji, X.: GDR-Net: Geometry-guided direct regression network for monocular 6D object pose estimation. *CVPR* (2021) [4](#)
88. Wang, Y., Shen, X., Hu, S., Yuan, Y., Crowley, J., Vafreydaz, D.: Self-supervised transformers for unsupervised object discovery using normalized cut. *CVPR* (2022) [2](#), [4](#)
89. Wen, B., Yang, W., Kautz, J., Birchfield, S.: FoundationPose: Unified 6D pose estimation and tracking of novel objects (2023) [4](#)
90. Wu, C.Y., Johnson, J., Malik, J., Feichtenhofer, C., Gkioxari, G.: Multiview compressive coding for 3D reconstruction. *CVPR* (2023) [2](#)
91. Xiang, Y., Schmidt, T., Narayanan, V., Fox, D.: PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes. *RSS* (2018) [4](#), [8](#), [10](#)
92. Xiao, Y., Qiu, X., Langlois, P., Aubry, M., Marlet, R.: Pose from shape: Deep pose estimation for arbitrary 3D objects. *BMVC* (2019) [4](#)
93. Zhang, J., Herrmann, C., Hur, J., Cabrera, L.P., Jampani, V., Sun, D., Yang, M.H.: A tale of two features: Stable diffusion complements DINO for zero-shot semantic correspondence. *NeurIPS* (2023) [4](#)