Open-vocabulary object 6D pose estimation

Jaime Corsetti[1,2] Davide Boscaini[1] Changjae Oh[3] Andrea Cavallaro[4,5] Fabio Poiesi[1]

[1]Fondazione Bruno Kessler [2]University of Trento [3]Queen Mary University of London [4]Idiap Research Institute [5]EPFL

{jcorsetti,dboscaini,poiesi}@fbk.eu c.oh@qmul.ac.uk andrea.cavallaro@epfl.ch

Abstract

We introduce the new setting of open-vocabulary object 6D pose estimation, in which a textual prompt is used to specify the object of interest. In contrast to existing approaches, in our setting (i) the object of interest is specified solely through the textual prompt, (ii) no object model (e.g., CAD or video sequence) is required at inference, and (iii) the object is imaged from two RGBD viewpoints of different scenes. To operate in this setting, we introduce a novel approach that leverages a Vision-Language Model to segment the object of interest from the scenes and to estimate its relative 6D pose. The key of our approach is a carefully devised strategy to fuse object-level information provided by the prompt with local image features, resulting in a feature space that can generalize to novel concepts. We validate our approach on a new benchmark based on two popular datasets, REAL275 and Toyota-Light, which collectively encompass 34 object instances appearing in four thousand image pairs. The results demonstrate that our approach outperforms both a well-established hand-crafted method and a recent deep learning-based baseline in estimating the relative 6D pose of objects in different scenes. Code and dataset are available at https://jcorsetti.github.io/oryon.

1. Introduction

Accurate 6D pose estimation is fundamental for a wide range of applications, such as robot manipulation [9], autonomous driving [28] and augmented reality [29]. While recent works achieve high degree of pose estimation accuracy, even in cluttered scenes [6, 23, 46] and with challenging symmetric objects [12], most are *instance-level* pose estimation models, which are trained and tested with the same set of objects [40, 46]. *Generalizable* or *one-shot* [2, 39, 42] pose estimation methods remove the above limitation by training a model on a diverse set of objects using recent large-scale datasets (e.g. ShapeNet6D [50] and

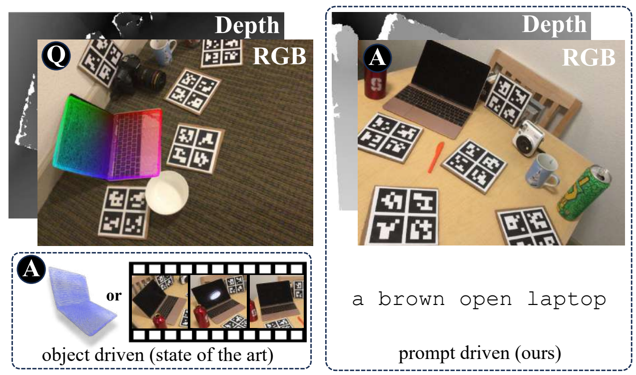


Figure 1. Our open-vocabulary setting enables the estimation of the pose of an object captured in two distinct scenes. State-of-the-art generalizable 6D pose estimation methods from RGB on RGBD images typically depend on the object CAD model [2] or a video sequence of the object at test time [24, 42] as a reference (A) to compute the object pose in the query image (Q). In contrast, our method uses a textual prompt to guide the pose estimation process, and requires a single view as reference.

MegaPose [20]), and test on novel objects without constraints about shape or category. However, *model-based* methods still require a model of the novel object at test time [2, 20, 39], while methods defined as *model-free* require a set of reference views [19, 47] or a video [13, 24, 42] of the novel object, and additional preprocessing. Such methods are unsuitable when novel objects are not physically available, as a video sequence cannot be acquired.

In this paper, we introduce a new 6D pose estimation formulation that changes the assumptions of previous approaches (Fig. 1), by using a textual prompt to identify the object of interest. To accomplish this, we integrate in the architecture a Vision-Language Model (VLM) to identify the object of interest from the two scenes and to estimate its relative 6D pose. We name our approach *Oryon* (**O**pen-vocabula**ry** **o**bject pose estimatio**n**). The proposed new formulation uses textual information (prompt) provided by the user to not only localize the object of interest in a cluttered scene, but also to guide our VLM to focus on points which are more specific to the target object. We define this formulation as *Open-Vocabulary* as we do not put any constraint on the input prompt. While previous works use language in related tasks [3], their contribution is limited to the localization of the object of interest. Instead, we show that textual prompts can provide rich semantic information to guide the VLM, and are fundamental to the generalization capabilities of our method. This setting can be applied to different scenarios. We focus on the *cross-scene* setting, in which the two RGBD images show different scenes, with one or possibly more objects in common. We validate Oryon on a new benchmark that is based on two popular datasets, namely REAL275 [43] and Toyota-Light [16]. The first shows an high variation of object poses and scenes with mild occlusions, while the second presents challenging light conditions. We compare Oryon against a well-established hand-crafted feature extraction method, i.e., SIFT [27], and a recent deep learning approach specifically designed for registration of point clouds with low overlap, i.e., ObjectMatch [11]. Oryon outperforms both SIFT and ObjectMatch on both datasets. We carry out an extensive ablation study to validate the different components. In summary, our contributions are as follows:

- We introduce a novel setting in object 6D pose estimation, featuring a new set of assumptions. This includes specifying the object of interest via a textual prompt, as opposed to relying on a 3D model or a video sequence;
- We propose an architecture based on a Vision-Language Model, capable of segmenting and producing local distinctive features for object matching;
- We establish a new benchmark based on two popular object 6D pose estimation datasets, REAL275 [43] and Toyota-Light [16]. The benchmark focuses on determining the relative pose of a text-prompted object captured from two different scenes.

2. Related work

Model-free pose estimation of novel objects. Gen6D [24] represents one of the first approach to tackle the pose estimation of novel objects without CAD models, using only RGB images. Gen6D implements an object-agnostic detector, a viewpoint selector, and a pose refinement step. During evaluation, Gen6D requires the acquisition of an RGB video sequence of the novel object, upon which a Structure-from-Motion method (COLMAP [38]) is applied to recover the camera pose for each frame. However, a significant limitation of Gen6D is the complexity of its evaluation procedure, which also requires manual cropping and orientation of the reconstructed point cloud. Similarly, OnePose [42] demands an RGB video sequence of the novel object and a comparable procedure at test time. OnePose++ [13] extends OnePose to address pose estimation for low-textured objects, but maintains the same test-time requirements as OnePose. In contrast, Oryon eliminates the need for SfM at test time, and only requires a textual description of the object, which can be effortlessly provided by a user without technical expertise.

Relative pose estimation aims to estimate the pose of an object in a scene (*query*) with respect to a reference view (*anchor*) [22, 31, 32, 51]. NOPE [31] trains an autoencoder to learn a representation conditioned on the relative orientation of the query image with the reference image. At test time, a set of reference features are produced, each associated with an orientation. The orientation of the reference feature more similar to the query one is chosen. Similarly to Oryon, NOPE is evaluated on novel objects and does not use object models. However, NOPE only predicts the relative rotation and has not been tested in real-world scenarios. LatentFusion [32] shares a similar setting, but by adopting RGBD scenes instead of RGB only is capable of estimating also the translation component. A similar task is 6D pose tracking, where several model-free methods have been recently proposed [30, 44, 45]. In this setting, the relative pose is estimated between pairs of consecutive frames. Methods like BundleTrack [44] and BundleSDF [45] rely on the information provided by previous frames by storing their representation in a memory pool. Instead, our reference is a single viewpoint image. Moreover, the segmentation module allows Oryon to tackle the case in which the query and reference frames are acquired in different scenes. See the Supplementary Material for an extended discussion of relative pose estimation methods.

Point cloud registration. A related problem to RGBD-based 6D pose estimation is point cloud registration. This task is commonly addressed by using specific methods for feature extraction [5, 6, 8, 33, 34] paired by matching and registration, or by end-to-end optimization [48, 49]. A recent advancement on this field is ObjectMatch [11], which tackles the problem of registration of two scene fragments captured from two 3D viewpoints with low overlap. This is achieved by implicitly computing object matches through the detection of objects in the scenes. Our scenario is significantly more challenging as the only overlapping parts of the scenes are the ones belonging to the object of interest. Therefore, our problem could be considered as a particular case of point cloud registration, in which the point cloud parts to be aligned are described by the textual prompt.

3. Our approach

3.1. Overview

The new setting is defined as follows: (i) the object of interest is specified solely through textual prompt (no object model or video sequence is required); (ii) the object is imaged from two different viewpoints of two different scenes; and (iii) the object was not observed during the training

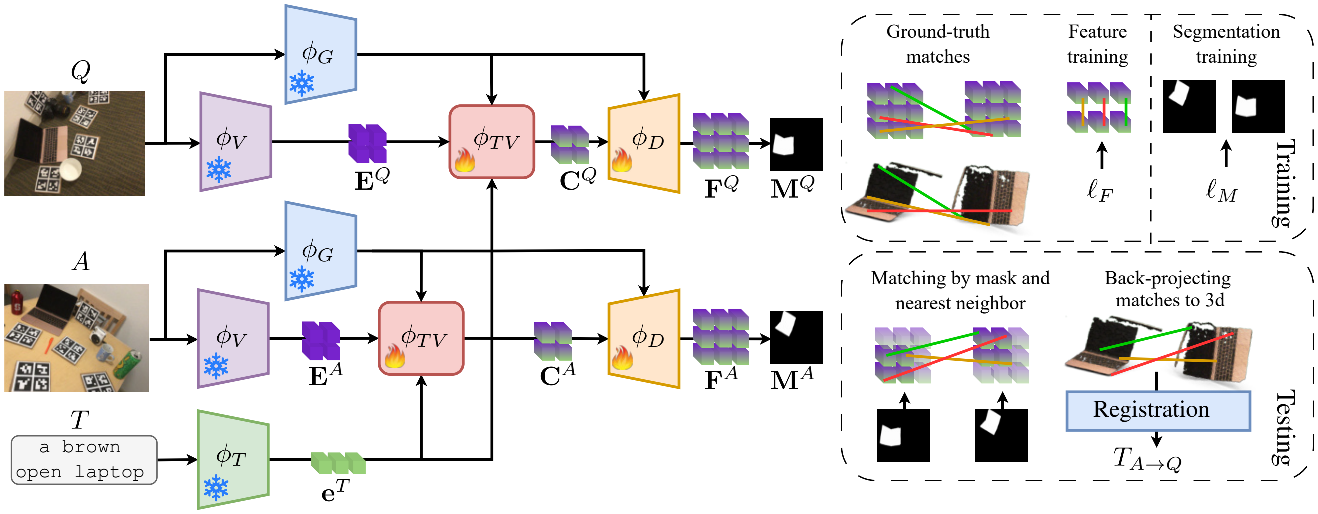


Figure 2. The training pipeline of Oryon can be separated in four stages. In the first stage, the pair of images A, Q are encoded by the CLIP image encoder ϕ_V, while the prompt T is encoded by the CLIP text encoder ϕ_T. The guidance network ϕ_G is used to produce a rich visual representation which is used in the next stages. The resulting visual feature maps $\mathbf{E}^A$, $\mathbf{E}^Q$ and text features $\mathbf{e}^T$ are used in the second stage in the fusion module ϕ_{TV}. This outputs a pair of cost features $\mathbf{C}^A$,$\mathbf{C}^Q$ which in the third stage are upsampled to the final feature maps $\mathbf{F}^A$, $\mathbf{F}^Q$. The same feature maps are fed to a segmentation head to obtain the predicted masks $\mathbf{M}^A$, $\mathbf{M}^Q$. At train time, $\mathbf{F}^A$, $\mathbf{F}^Q$ are optimized by a contrastive loss, while a dice loss supervises the training of the segmentation masks. At test time, the predicted masks are used to filter $\mathbf{F}^A$, $\mathbf{F}^Q$, and matches are obtained by nearest neighbor. Finally, the matches are projected back to the 3D domain, and a registration algorithm is used to retrieve the final pose $T_{A\rightarrow Q}$.

phase. From these requirements, we formulate the problem as a relative pose estimation task [31, 51] between two scenes, the anchor A and the query Q, which are represented as RGB image RGB^A (RGB^Q) and depth map D^A (D^Q). To match A and Q we propose Oryon, which relies on a fusion module based on cost-aggregation that relates the information of the textual prompt to the local visual feature map. The model generalizes to unseen objects and is trained on a large dataset of object models annotated with brief textual descriptions [50].

We address the localization problem jointly with the matching task, by predicting a segmentation mask on each view (scene) to locate the object of interest. The final features encode *semantic* information about the objects for the segmentation task and *geometric* information, as the pose is derived from the relative position of keypoints. At test time, we use the predicted masks to select points on the object of interest and match pairs of features by nearest neighbor. The resulting matches are projected back to the 3D domain to obtain the final relative object pose across the two scenes through point cloud registration [1].

Fig. 2 shows Oryon's architecture. We describe the object of interest with a user-provided textual prompt T. We use T to guide the extraction of keypoints and to compute the matches, as only the points which match the textual description provided by the prompt itself are considered through the use of the mask. Given a pair of scenes (A, Q), we compute a set of coordinates $\mathbf{x}^A$ and $\mathbf{x}^Q$ that form a set of matches between scenes (i.e., they describe the most similar locations in color and 3D structure), which are filtered by the predicted masks $\mathbf{M}^A$, $\mathbf{M}^Q$.

In order to jointly process T and (A, Q), their representations should share the same feature space. We achieve this by leveraging a VLM trained for semantic alignment, such as CLIP [35]. We process RGB^A and RGB^Q with the CLIP image encoder ϕ_V, and extract the features before the final pooling layer. Let ($\mathbf{E}^A$, $\mathbf{E}^Q$) $\in \mathbb{R}^{D\times H\times H}$ be a pair of feature maps extracted from A and Q, where D is the feature dimension, and H is the spatial dimension of the feature map. D and H depend on the CLIP encoder used. Note that RGB^A and RGB^Q are processed without any prior crop, unlike other pose estimation approaches which adopt prior detectors or segmenters [13, 24, 40, 42, 46]. T is encoded by the CLIP text encoder ϕ_T. To provide a rich set of representations, we adopt a set of templates to generate N versions of T [4, 35], which are encoded by ϕ_T. Hence, we obtain the prompt features $\mathbf{e}^T \in \mathbb{R}^{N\times D}$, where N is the number of templates. Both CLIP encoders are frozen.

3.2. Text-visual fusion

The objective of the text-visual module ϕ_{TV} is to provide a visual representation which is semantically consistent with the represented object (and therefore influenced by $\mathbf{e}^T$), but which is also representative of the object local appearance in $\mathbf{E}^A$ and $\mathbf{E}^Q$. The first requirement is fundamental for the segmentation task, while the second is needed to perform image matching. We implement the fusion module ϕ_{TV} by building a cost volume, i.e. by computing the cosine similarity among each feature map location in $\mathbf{E}^A$, $\mathbf{E}^Q$ and $\mathbf{e}^T$, thus obtaining a pair of matrices of shape $\mathbb{R}^{N\times H\times H}$. The cost volume represents the cost of associating each prompt feature to each location in the visual feature maps.

Note that the cost volume obtained does not enforce a spatial consistency between neighboring patches. In order to let the model learn these relations and refine the cost volume, we adopt a cost aggregation block based on two Transformer layers [25]. Both layers rely on self-attention to model the relationship between the image patches, with different modalities. The first applies self-attention within the same window, while the second applies self-attention among patches of shifted windows. This allows the model to perform attention both within the patch and between neighboring patches. Objects can have different shapes and sizes in the image, therefore only applying local attention can not be sufficient to build an effective representation. To enrich the representation provided by CLIP, we adopt guidance features [4] from another backbone ϕ_G which are concatenated to the query and keys of the Transformer layers. The output of the fusion module is a pair of cost features $\mathbf{C}^A$, $\mathbf{C}^Q \in \mathbb{R}^{D\times H\times H}$.

3.3. Decoding

The feature maps obtained in the previous step have the original low resolution of the CLIP feature map (24×24 in our case). As we pursue an image matching objective, we require a higher resolution to compute fine-grained matches. To this end, we adopt a decoder ϕ_D composed by three upsampling layers. Note that CLIP was trained for semantic alignment of the global feature, and this property partially transfers to its visual feature map [52]. However, in our setting we also require information based on the appearance of the object, as opposed to semantic information only. Therefore, we find beneficial to add the guidance features of ϕ_G in the decoder. Specifically, the two feature maps obtained by the guidance network are projected and concatenated to the input feature map before each upsampling layer [4]. The final layer does not use any guidance.

The resulting feature maps $\mathbf{F}^A$, $\mathbf{F}^Q$ are used both for computing the matches and for the segmentation task. For the latter, we add a segmentation head to compute the activations and output a pair of binary masks $\mathbf{M}^A$, $\mathbf{M}^Q$.

3.4. Optimization

Our formulation implies two optimization objectives. For effective image matching, we directly optimize the feature maps by forcing low similarity between dissimilar locations and high similarity between similar locations across A and Q. As supervision, we adopt the ground-truth matches between A and Q (i.e. pair of points on A and Q which belong to the same portion of the object of interest). In practice, we adopt a contrastive loss ℓ_F with hardest negative mining [5, 6]. There are two components: the first promotes the corresponding features (i.e. the matches) to be close in the feature space, while the second increases the distance between a feature and its *hardest negative*. Given the pair of scenes A, Q, the set of positive pairs is defined as $\mathcal{P} = \{(i,j) \colon \mathbf{x}_i^A \in \mathrm{A}, \mathbf{x}_j^Q \in \mathrm{Q}, \phi(\mathbf{x}_i^A) = \mathbf{x}_j^Q\}$, where $\phi\colon A \to Q$ is a match mapping between A and Q pixels. We define $\mathbf{f}^A$, $\mathbf{f}^Q \in \mathbb{R}^{C\times D}$ as the set of features extracted respectively from the feature maps $\mathbf{F}^A$, $\mathbf{F}^Q$, by using the ground-truth matches $\mathcal{P}$. $C = |\mathcal{P}|$ is the total number of matches and D is the feature dimension. The positive loss component ℓ_P is defined as

$$\ell_P = \sum_{(i,j)\in\mathcal{P}} \frac{1}{|\mathcal{P}|} \left(\mathrm{dist}(\mathbf{f}_i^A, \mathbf{f}_j^Q) - \mu_P\right)_+, \tag{1}$$

where μ_P is a positive margin and $(\cdot)_+ = \max(0,\cdot)$. The positive margin is an hyperparameter, and represents the desired distance in the feature space of a positive pair.

To define negative pairs, we consider a set of features $\mathbf{f}$ and their corresponding 2D coordinates $\mathbf{x}$ on the image. Given a single feature $\mathbf{f}_i$, we define its set of candidate negatives as $\mathcal{N}_i = \{k \colon \mathbf{x}_k \in \mathbf{x}, k \neq i, \|\mathbf{x}_i - \mathbf{x}_k\| \geq \tau\}$. Note that this set excludes all the points which are closer than the distance τ from the reference point $\mathbf{x}_i$ in the image, so that features describing the same points are not considered. We compute the candidate negatives set for all points of $\mathbf{x}^A$ and $\mathbf{x}^Q$, and define the negative loss component ℓ_N as

$$\begin{aligned}\ell_N = \sum_{(i,j)\in\mathcal{P}} & \frac{1}{2|\mathcal{P}_i|} \left(\mu_N - \min_{k\in\mathcal{N}_i} \mathrm{dist}(\mathbf{f}_i, \mathbf{f}_k)\right)_+ \\ & + \frac{1}{2|\mathcal{P}_j|} \left(\mu_N - \min_{k\in\mathcal{N}_j} \mathrm{dist}(\mathbf{f}_j, \mathbf{f}_k)\right)_+ .\end{aligned} \tag{2}$$

For each $\mathbf{f}_i$, the nearest $\mathbf{f}_k$ in the feature space (i.e. the hardest negative) is selected. The negative margin μ_N is an hyperparameter, which defines the desired distance in the feature space of a negative pair. In Eqs. (1) and (2), $\mathrm{dist}(\cdot)$ is the inverted and normalized cosine similarity. The feature loss ℓ_F is thus defined as

$$\ell_F = \lambda_N \ell_N + \lambda_P \ell_P, \tag{3}$$

where λ_N and λ_P are hyperparameters.

We implement the segmentation loss ℓ_M as a Dice loss [41]. We found the Dice loss to be more effective than cross entropy, as the first is specifically designed to handle high class imbalance in the foreground mask. This is consistent with our scenario, in which the object of interest may occupy a small portion of the image. The final loss ℓ is defined as

$$\ell = \lambda_M \ell_M + \ell_F, \tag{4}$$

where the weight factor λ_M is an hyperparameter.

3.5. Matching and registration

At test time, the predicted masks $\mathbf{M}^A$, $\mathbf{M}^Q$ are used to retain the features of $\mathbf{F}^A$, $\mathbf{F}^Q$ belonging to the objects, thus obtaining two lists $\mathbf{F}^A_M \in \mathbb{R}^{C^1 \times D}$, $\mathbf{F}^Q_M \in \mathbb{R}^{C^2 \times D}$. For each feature $\mathbf{f}^A_i \in \mathbf{F}^A_M$, we compute the nearest neighbor $\mathbf{f}^Q_i \in \mathbf{F}^Q_M$ in the feature space. We reject the pairs $\mathbf{f}^A_i, \mathbf{f}^Q_i$ for which $\mathrm{dist}(\mathbf{f}^A_i, \mathbf{f}^Q_i) > \mu_t$. We select all points belonging to a match and back-project them to 3D, by using the depth maps and the intrinsic camera parameters of A and Q, thus obtaining two point clouds $\mathbf{P}^A, \mathbf{P}^Q \in \mathbb{R}^{C \times 3}$.

The pose $T_{A \to Q}$ is estimated with a point cloud registration algorithm. Due to noise in the predicted masks and possible ambiguity in the prompt, we expect spurious matches to be present. We use PointDSC [1] for its robust matching capabilities based on spatial consistency, which allows it to reject inconsistent matches and provide precise poses.

4. Results

4.1. Experimental setup

We train our model for 20 epochs, and adopt CLIP ViT-L/14 [35] as visual and text encoder. We use Adam [18] with learning rate 10^{-4}, weight decay $5 \cdot 10^{-4}$, and a cosine annealing scheduler [26] to lower the learning rate to 10^{-5}. Loss weights are set as $\lambda_P = \lambda_N = 0.5$ and $\lambda_M = 1.0$. As guidance backbone, we adopt a pretrained Swin Transformer [25]. As augmentations we randomly apply color jittering, horizontal flipping and vertical flipping.

We set the positive and negative margins in Eqs. (1) and (2) as $\mu_P = 0.2$ and $\mu_N = 0.9$, and the excluding distance for hardest negatives as $\tau = 5$. At test time we set $\mu_t = 0.25$ as maximum feature distance to identify a match, and limit the match number to $C = 500$. The output resolution of $\mathbf{F}^A$, $\mathbf{F}^Q$ is 192×192, and the feature dimension is $F = 32$.

4.2. Datasets

We use the synthetic dataset ShapeNet6D [50] for training, as it provides several diverse objects and scenes. We evaluate Oryon on two real-world datasets: REAL275 [43] and Toyota-Light [16]. The standard prompt T we adopt at test time is composed by the object name (e.g., “laptop”) preceded by a brief description (e.g., “brown open”). For REAL275 and TOYL, we manually annotate each object instance with a textual prompt, while for ShapeNet6D we rely on its metadata. For all datasets, we train and test on a set of scene pairs of images from the original datasets.

ShapeNet6D (SN6D) [50] is a large-scale synthetic dataset built by using ShapeNetSem [37] object models. For each object model, ShapeNetSem includes a name and a set of synonyms, which we use to build the textual prompt. Note that in this dataset the prompt is only composed by the object name, as no description is provided. To enrich the learned representations, during training we randomly substitute the object name in the prompt with one of the synonyms provided in the metadata (e.g., possible synonyms for “television” are “tv” and “telly”). Note that ShapeNet6D does not contain the object instances present in the two test datasets, although it may contain objects of similar category. We train on 20K image pairs from SN6D.

REAL275 [43] provides a set of RGBD images in different scenes, with a total of six object categories and three instances for each category. REAL275 provides an high variety of viewpoints between the scenes, and also present mild occlusions. We evaluate on 2K image pairs from the original real-world test set.

Toyota-Light (TOYL) [16] focuses on pose estimation under challenging light conditions, in which a single object is present in each scene. This is relevant in our setting, as we process pairs of images: a significant difference in light across the two scenes poses an important challenge in the image matching task. We evaluate on 2K image pairs.

4.3. Evaluation metrics

We evaluate all our pose results by using the metrics proposed for the BOP Benchmark [17], namely AR (Average Recall), which is the average of VSD, MSSD and MSPD. We also report ADD(S)-0.1d, as it is typically used in 6D pose estimation [14, 15, 24]. ADD(S)-0.1d is a recall on the pose error: it considers a success when the error is less than one tenth of the object diameter. Instead, VSD, MSSD and MSPD are averages of recalls on multiple thresholds. These latter metrics are more suitable to represent performance in this challenging scenario. Therefore, we adopt AR as main metric. See the Supplementary Material for an extended discussion of the pose metrics. For all experiments we also report the mean Intersection-Over-Union (mIoU) to quantify the predicted mask quality [4, 52].

4.4. Comparison procedure

Our setting is fundamentally different from the ones of current 6D pose estimation methods. Gen6D [24] and OnePose [42] both rely on video sequences of the novel objects, while methods for relative pose estimation [31, 51] only work with RGB images and do not estimate the translation component. Therefore, we compare Oryon with ObjectMatch [11], a state-of-the-art method designed for point cloud registration with low overlap. This is consistent with our scenario, as we assume that the only overlapping section between two scenes is the one showing the query object. ObjectMatch is based on SuperGlue [36] to estimate the matches, and on a custom pose estimator. We run SuperGlue on each image pair. As in our approach, the predicted mask is used to reject all matches on the background, and the resulting matches are passed to the final ObjectMatch module that outputs the pose. We use the model trained

Table 1. Results on REAL275 [43]. We report in **bold** font and underlined respectively the best and second best result when using our predicted mask. The Δ score is the difference between Oryon and the nearest competitor when using our predicted mask. Key: Obj.Mat.: ObjectMatch, ADD: ADD(S)-0.1d.

	Method	Prior	AR↑	VSD↑	MSSD↑	MSPD↑	ADD↑	mIoU↑
1	SIFT [27]	Oracle	34.1	16.5	37.9	48.0	16.4	100.0
2		OVSeg [21]	18.3	8.6	19.9	26.5	7.4	56.4
3		Ours	24.4	12.2	27.3	33.8	12.8	66.5
4	Obj.Mat. [11]	Oracle	26.0	15.5	31.7	30.8	13.4	100.0
5		OVSeg [21]	14.9	9.1	18.8	16.8	7.8	56.4
6		Ours	22.4	14.1	27.9	25.2	13.2	66.5
7	Oryon	Oracle	46.5	32.1	50.9	56.7	34.9	100.0
8		OVSeg [21]	26.4	18.3	29.4	31.5	17.2	56.4
9		Ours	**32.2**	**23.6**	**36.6**	**36.4**	**24.3**	**66.5**
10	Δ score		+7.8	+9.5	+8.7	+2.6	+11.1	+10.1

Table 2. Results on Toyota-Light [16]. We report in **bold** and underlined respectively the best and second best result when using our predicted mask when using our predicted mask. The Δ score is the difference between Oryon and the nearest competitor. Key: Obj.Mat.: ObjectMatch, ADD: ADD(S)-0.1d.

	Method	Prior	AR↑	VSD↑	MSSD↑	MSPD↑	ADD↑	mIoU↑
1	SIFT [27]	Oracle	30.3	7.3	39.6	44.1	14.1	100.0
2		OVSeg [21]	25.8	6.4	34.2	36.9	11.8	75.5
3		Ours	27.2	5.7	35.4	40.6	9.9	68.1
4	Obj.Mat. [11]	Oracle	9.8	2.4	13.0	14.0	5.4	100.0
5		OVSeg [21]	9.2	2.6	12.1	13.0	5.3	75.5
6		Ours	8.3	2.2	10.5	12.1	3.8	68.1
7	Oryon	Oracle	34.1	13.9	42.9	45.5	22.9	100.0
8		OVSeg [21]	29.2	11.9	36.8	38.9	18.9	**75.5**
9		Ours	**30.3**	**12.1**	**37.5**	**41.4**	**20.9**	68.1
10	Δ score		+3.1	+6.4	+2.1	+0.8	+11.0	-7.4

on ScanNet [7] from the official repository. We also compare Oryon with a pipeline based on SIFT features [27] and PointDSC [1]. SIFT is used to extract keypoints and descriptors, which are then filtered by using the segmentation prior. As in Oryon, the resulting features are used to compute matches between A and Q, which are unprojected in 3D and used to register the point clouds with PointDSC.

For all experiments we report results by using different segmentation priors: (i) the mask predicted by our method (Ours), (ii) the mask predicted with OVSeg [21], an off-the-shelf method for open-vocabulary segmentation, and (iii) the ground-truth mask (Oracle).

4.5. Quantitative results

Tab. 1 reports the results on REAL275. When OVSeg is used as the image segmenter, Oryon outperforms SIFT by $+8.1$ in AR and ObjectMatch by $+11.5$ in AR. With our segmentation head as the image segmenter, Oryon shows a performance increase over SIFT by $+7.8$ in AR and over ObjectMatch by $+9.8$ in AR. We attribute the smaller performance gap between Oryon and SIFT compared to that with ObjectMatch to the latter's greater sensitivity to domain shifts. The performance gap between Oryon based on its own predicted masks and the Oracle masks is -14.3 AR, which indicates that a portion of the objects are not estimated correctly due to errors in the segmentation.

Tab. 2 reports the results on TOYL. When OVSeg is used as segmenter, Oryon outperforms SIFT by $+3.4$ AR and ObjectMatch by $+20.0$ AR. With our segmentation head as the image segmenter, Oryon shows a performance increase over SIFT by $+3.1$ in AR and over ObjectMatch by $+22.0$ in AR. In this dataset, SIFT performances are closer to ours than in REAL275. On the contrary, ObjectMatch performs much worse. The lower performances of ObjectMatch are due to its sensitivity to the domain shift, which in TOYL is more present due to the different light conditions in the scene pair. Another characteristic of TOYL is its high variation in poses, for which objects can have very different apparent sizes in the two images. SIFT scale-invariant design allows it to tackle this variance in appearance and reach better results than ObjectMatch. We also observe that the performance gap between Oryon with its own masks and the Oracle ones is narrower than in REAL275. This suggests that Oryon does not require highly accurate segmentation masks, as PointDSC is able to reject spurious matches.

4.6. Qualitative results

We report in Figs. 3, 4 some qualitative results on REAL275 [43] and TOYL [16], respectively. The predicted mask correctly localizes the objects, but ObjectMatch fails at estimating an accurate pose, due to translation (Fig. 3, top) or rotation errors, (Fig. 3, bottom). Due to the small objects size, this scenario is challenging for ObjectMatch, while Oryon's higher resolution allows it to retrieve a correct pose also in this case. Despite small errors in the translation component, SIFT reaches a good degree of accuracy.

On TOYL the results show that our method can handle different light conditions between A and Q. On the other hand, we observe that the translation component is not as accurate as in REAL275 (see Fig. 4, bottom). Both ObjectMatch and SIFT present large errors in this challenging dataset. SIFT in particular shows large errors in translation, which suggests that such hand-crafted features are unsuitable for this scenario with different light conditions.

4.7. Ablation study

We report in Tab. 3 the results obtained on REAL275 when we remove one of the modules of our architecture. In rows 1, 2 we remove the feature guidance respectively from the fusion module ϕ_{TV} and from the decoder ϕ_D. Both changes result in a significant drop both in pose estimation performance (-7.4 and -11.0 in AR respectively) and in the segmentation (-8.0 and -12.9 in mIoU respectively). This confirms the intuition that adding the guidance from a general-purpose image encoder ϕ_G [25] can specialize the semantic features of CLIP [35] for our task. Removing the

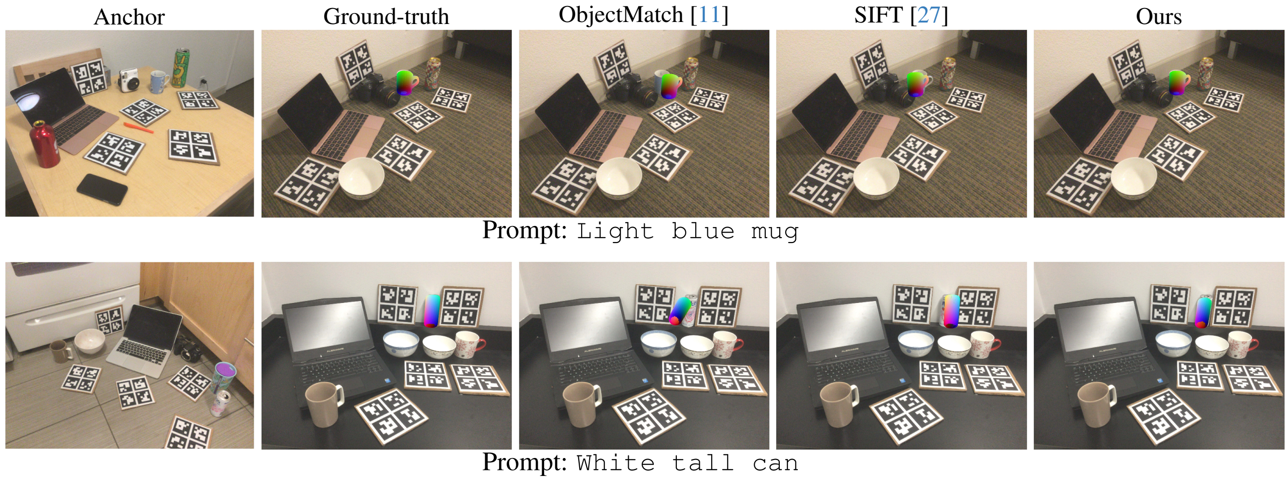


Figure 3. Examples of qualitative pose results from the REAL275 [43] dataset. All the results use the segmentation mask predicted by Oryon. We show the object model colored my mapping its 3D coordinates to the RGB space.

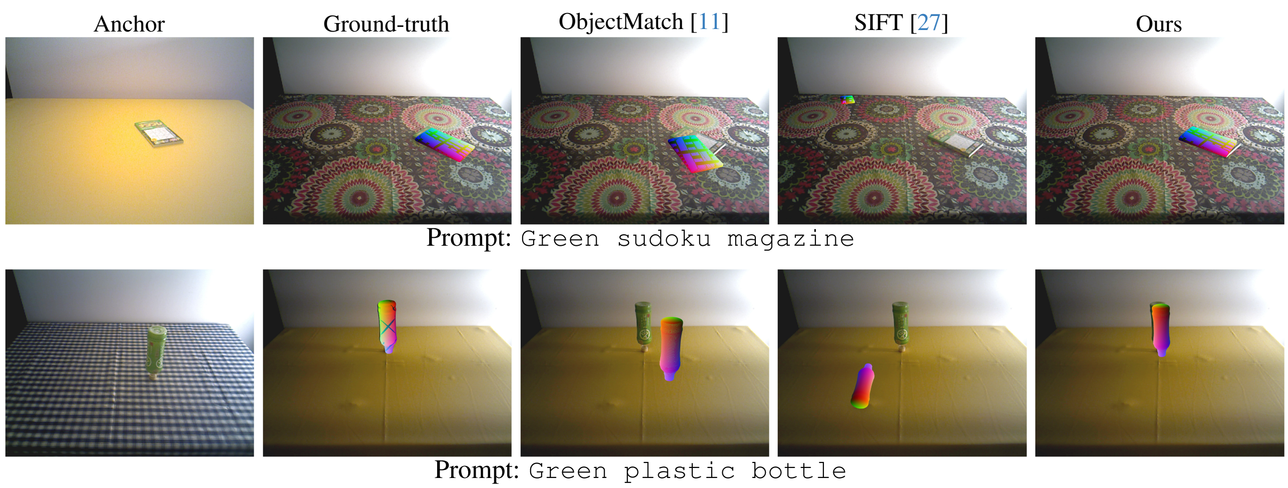


Figure 4. Examples of qualitative pose results from the TOYL [16] dataset. All the results use the segmentation mask predicted by Oryon. We show the object model colored my mapping its 3D coordinates to the RGB space.

upsampling layer (row 3) results in a worse performance on pose estimation, while the impact on the segmentation is small (-7.1 and -1.5 in AR and mIoU respectively). This highlights the importance of a higher-resolution feature map to compute the matches, as only using two upsampling layers result in a feature map of size 96×96 instead of our default 192×192. In row 4, we replace PointDSC with a RANSAC-based algorithm from a state-of-the-art pose estimation method [46]. This causes an important drop in AR (-16.2), which shows the importance of a registration algorithm capable of dealing with spurious matches [1].

In Tab. 4 we report the results of two experiments on REAL275 in which we train Oryon for a single task only (segmentation in rows 1-2, and matching in row 3). When training only with the segmentation loss ℓ_M the features learned by Oryon led to worse pose estimation performance than our baseline, as the AR changes by -7.4 when using the predicted mask (row 1 vs row 5). Also the segmentation performance is lower (-2.2 mIoU): this suggests that jointly training Oryon for the two tasks can benefit the segmentation performance. In row 3 we only train Oryon with the feature loss ℓ_F. In this case, the AR drops by -6.2 when comparing with the same segmenter (row 3 vs row 4). This confirms our intuition that jointly training a model for pose estimation and segmentation can benefit both tasks.

In Tab. 5 we report the results of the evaluation on REAL275 with different prompts. In rows 1-2 the object name is replaced by "object", in order to simulate the possible behaviour of a user describing an unknown object, for which a name cannot be provided. In this case, the segmentation completely fails, and subsequently the predicted poses are poor (-29.2 AR and -63.5 mIoU with respect to

Table 3. Ablation on the architecture components on REAL275 [43], with our predicted masks. Key: ADD: ADD(S)-0.1d, **Bold**: best method.

	Method	AR↑	VSD↑	MSSD↑	MSPD↑	ADD↑	mIoU↑
1	w/o cost agg. guidance	24.8	17.2	28.8	28.4	17.9	58.5
2	w/o decoder guidance	21.2	16.4	23.2	24.0	14.4	53.6
3	w/o extra upsampling	25.1	16.4	27.6	31.1	18.5	65.0
4	w RANSAC [10]	15.0	4.7	16.0	24.4	5.2	66.5
5	Oryon	**32.2**	**23.6**	**36.6**	**36.4**	**24.3**	**66.5**

Table 4. Ablation on the two tasks performed by Oryon on REAL275 [43]. Key: ADD: ADD(S)-0.1d, **Bold**: best method.

	Method	Prior	AR↑	VSD↑	MSSD↑	MSPD↑	ADD↑	mIoU↑
1	Segm. only	Ours	24.8	15.1	28.2	31.0	14.6	63.3
2	Segm. only	Oracle	33.2	18.4	35.8	45.4	18.1	100.0
3	Matches only	Oracle	40.3	27.3	42.8	50.7	26.6	100.0
4	Oryon	Oracle	46.5	32.1	50.9	56.7	34.9	100.0
5		Ours	**32.2**	**23.6**	**36.6**	**36.4**	**24.3**	**66.5**

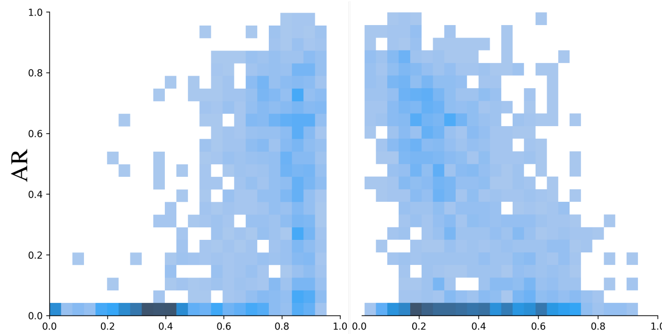
Table 5. Ablation on the prompt on REAL275 [43]. Key: ADD: ADD(S)-0.1d, **Bold**: best method.

	Prompt type	Prior	AR↑	VSD↑	MSSD↑	MSPD↑	ADD↑	mIoU↑
1	No name	Oracle	38.6	22.2	42.7	50.8	21.5	100.0
2		Ours	3.0	1.4	4.1	3.6	0.4	3.0
3	Misleading	Oracle	39.4	22.1	43.5	52.8	24.9	100.0
4		Ours	25.4	19.1	29.0	28.0	14.5	56.4
5	Generic	Oracle	39.0	22.7	43.4	50.7	26.1	100.0
6		Ours	30.0	21.9	34.0	34.1	19.0	63.4
7	Oryon	Oracle	46.5	32.1	50.9	56.7	34.9	100.0
8		Ours	**32.2**	**23.6**	**36.6**	**36.4**	**24.3**	**66.5**

the baseline). In rows 3-4 we input a misleading prompt to the network, by keeping the original object name but changing the brief description (e.g., "a white closed laptop" for a laptop that appears open and brown). This results in a change of -6.8 AR and -10.1 mIoU with respect to the baseline (row 8). However, the model does not fail completely. This suggests that Oryon has some capabilities of selecting the most useful information from the prompt given the appearance on the scenes. When the object description is removed and only the object name is kept (rows 5-6), the drop in performance with respect to the baseline is less severe (-2.2 in AR, -3.1 in mIoU). REAL275 presents some scenes with multiple objects with the same name (e.g., Fig. 3 at the bottom shows two "bowl" objects). In these scenes, the description can be decisive to solve the ambiguity, while in the other cases it is less important. See the Supplementary Material for the complete list of prompts.

4.8. Sensitivity analysis

We report in Fig. 5 the distribution of AR with segmentation quality and camera distance between A and Q across all samples of the REAL275 test set. We observe in Fig. 5(a) that AR is low before reaching about 0.4 IoU, after which there is a sharp increase in pose quality. On the other hand,



(a) mIoU (b) Camera distance

Figure 5. Distribution of AR with mIoU (a) and camera distance between image pairs (b) of our best experiment on REAL275 [43].

AR appears to reach a plateau after 0.7 IoU. This confirms our intuition that Oryon does not require an high segmentation accuracy to successfully estimate the pose. However, there is not a direct dependency relation between the two metrics, as there are many samples with low AR and high mIoU. On the other hand, the relation between pose quality and camera distance in Fig. 5(b) is more linear, as the latter is lower when AR is higher. Intuitively, an high camera distance implies a lower number of matches, and therefore lowers the pose quality.

5. Conclusions

We present Oryon, an approach to tackle generalizable pose estimation from a new perspective. Instead of relying on a visual representation of the novel object and on complex onboarding procedures, we use a textual description of the object of interest, which can be provided by a non-expert user. We show that our approach is successful in the challenging scenario in which the image pair shows different scenes. Oryon accomplishes this by jointly addressing image matching and segmentation in an open-vocabulary approach, and by leveraging on a text-visual fusion module. We provide extensive ablation studies to assess the importance of each component and of the prompt composition.

Limitations. Oryon requires depth maps in order to run the registration algorithm and retrieve the relative pose. This limits the applicability of our method to contexts in which the depth information is available. Another limitation is in providing a description for certain objects (e.g., mechanical components) in order to express the prompt.

Future work. Oryon could be adapted to work with RGB images only by considering depth prediction.

Acknowledgements. This work was supported by the European Union's Horizon Europe research and innovation programme under grant agreement No 101058589 (AI-PRISM), and it made use of time on the Tier 2 HPC facility JADE2, funded by EPSRC (EP/T022205/1).

References

[1] X. Bai, Z. Luo, L. Zhou, H. Chen, L. Li, Z. Hu, H. Fu, and C. L. Tai. PointDSC: Robust Point Cloud Registration using Deep Spatial Consistency. In *CVPR*, 2021. 3, 5, 6, 7

[2] D. Cai, J. Heikkila, and E. Rahtu. OVE6D: Object Viewpoint Encoding for Depth-based 6D Object Pose Estimation. In *CVPR*, 2022. 1

[3] C. Cheang, H. Lin, Y. Fu, and X. Xue. Learning 6-DoF Object Poses to Grasp Category-Level Objects by Language Instructions. In *ICRA*, 2022. 2

[4] S. Cho, H. Shin, S. Hong, S. An, S. Lee, A. Arnab, P. H. Seo, and S. Kim. Cat-seg: Cost aggregation for open-vocabulary semantic segmentation. In *arXiv:2303.11797*, 2023. 3, 4, 5

[5] C. Choy, J. Park, and V. Koltun. Fully Convolutional Geometric Features. In *ICCV*, 2019. 2, 4

[6] J. Corsetti, D. Boscaini, and F. Poiesi. Revisiting Fully Convolutional Geometric Features for Object 6D Pose Estimation. In *ICCV*, 2023. 1, 2, 4

[7] A. Dai, Angel X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. In *CVPR*, 2017. 6

[8] D. DeTone, T. Malisiewicz, and A. Rabinovich. SuperPoint: Self-Supervised Interest Point Detection and Description. In *CVPR*, 2018. 2

[9] G. Du, K. Wang, S. Lian, and K. Zhao. Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: A review. In *Artificial Intelligence Review*, 2021. 1

[10] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. In *Communications of the ACM*, 1981. 8

[11] C. Gümeli, A. Dai, and M. Nießner. ObjectMatch: Robust Registration using Canonical Object Correspondences. In *CVPR*, 2023. 2, 5, 6, 7

[12] R.L. Haugaard and A.G. Buch. SurfEmb: Dense and Continuous Correspondence Distributions for Object Pose Estimation with Learnt Surface Embeddings. In *CVPR*, 2022. 1

[13] X. He, J. Sun, Y. Wang, D. Huang, H. Bao, and X. Zhou. OnePose++: Keypoint-Free One-Shot Object Pose Estimation without CAD Models. In *NeurIPS*, 2022. 1, 2, 3

[14] Y. He, H. Huang, H. Fan, Q. Chen, and J. Sun. FFB6D: A Full Flow Bidirectional Fusion Network for 6D Pose Estimation. In *CVPR*, 2021. 5

[15] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab. Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes. In *ACCV*, 2012. 5

[16] T. Hodan, F. Michel, E. Brachmann, W. Kehl, A. GlentBuch, D. Kraft, B. Drost, J. Vidal, S. Ihrke, X. Zabulis, et al. Bop: Benchmark for 6D object pose estimation. In *ECCV*, 2018. 2, 5, 6, 7

[17] T. Hodan, M. Sundermeyer, B. Drost, Y. Labbe, E. Brachmann, F. Michel, C. Rother, and J. Matas. BOP challenge 2020 on 6D object localization. In *ECCV*, 2020. 5

[18] D.P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *ICLR*, 2015. 5

[19] A. Kuznetsova, S. J. Hwang, B. Rosenhahn, and L. Sigal. Exploiting view-specific appearance similarities across classes for zero-shot pose prediction: A metric learning approach. In *AAAI*, 2016. 1

[20] Y. Labbé, L. Manuelli, A. Mousavian, S. Tyree, S. Birchfield, J. Tremblay, J. Carpentier, M. Aubry, D. Fox, and J. Sivic. Megapose: 6D pose estimation of novel objects via render & compare. In *CoRL*, 2022. 1

[21] F. Liang, B. Wu, X. Dai, K. Li, Y. Zhao, H. Zhang, P. Zhang, P. Vajda, and D. Marculescu. Open-vocabulary semantic segmentation with mask-adapted clip. In *CVPR*, 2023. 6

[22] A. Lin, J. Y. Zhang, D. Ramanan, and S. Tulsiani. Relpose++: Recovering 6d poses from sparse-view observations. In *arXiv:2305.04926*, 2023. 2

[23] S. Lin, Z. Wang, Y. Ling, Y. Tao, and C. Yang. E2EK: End-to-End Regression Network Based on Keypoint for 6D Pose Estimation. In *RA-L*, 2022. 1

[24] Y. Liu, Y. Wen, S. Peng, C. Lin, X. Long, T. Komura, and W. Wang. Gen6d: Generalizable model-free 6-DoF object pose estimation from rgb images. In *ECCV*, 2022. 1, 2, 3, 5

[25] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 4, 5, 6

[26] I. Loshchilov and F. Hutter. SGDR: Stochastic gradient descent with warm restarts. In *ICLR*, 2017. 5

[27] D. Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999. 2, 6, 7

[28] F. Manhardt, W. Kehl, and A. Gaidon. Roi-10d: Monocular lifting of 2D detection to 6D pose and metric shape. In *CVPR*, 2019. 1

[29] E. Marchand, H. Uchiyama, and F. Spindler. Pose estimation for augmented reality: a hands-on survey. In *TVCG*, 2015. 1

[30] V. Nguyen, Y. Du, Y. Xiao, M. Ramamonjisoa, and V. Lepetit. Pizza: A powerful image-only zero-shot zero-cad approach to 6 DoF tracking. In *3DV*, 2022. 2

[31] V. Nguyen, T. Groueix, Y. Hu, M. Salzmann, and V. Lepetit. Nope: Novel object pose estimation from a single image. In *arXiv:2303.13612*, 2023. 2, 3, 5

[32] K. Park, A. Mousavian, Y. Xiang, and D. Fox. Latentfusion: End-to-end differentiable reconstruction and rendering for unseen object pose estimation. In *CVPR*, 2020. 2

[33] F. Poiesi and D. Boscaini. Distinctive 3D local deep descriptors. In *ICPR*, 2021. 2

[34] F. Poiesi and D. Boscaini. Learning general and distinctive 3D local deep descriptors for point cloud registration. In *TPAMI*, 2022. 2

[35] A. Radford, J. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 3, 5, 6

[36] P. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 5

[37] M. Savva, A. Chang, and Hanrahan P. Semantically-Enriched 3D Models for Common-sense Knowledge. In *CVPR*, 2015. 5

[38] J. Schönberger and J. Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 2

[39] I. Shugurov, F. Li, B. Busam, and S. Ilic. OSOP: a multi-stage one shot object pose estimation framework. In *CVPR*, 2022. 1

[40] Y. Su, M. Saleh, T. Fetzer, J. Rambach, N. Navab, B. Busam, D. Stricker, and F. Tombari. ZebraPose: Coarse to Fine Surface Encoding for 6DoF Object Pose Estimation. In *CVPR*, 2022. 1, 3

[41] C. Sudre, W. Li, T. Vercauteren, S. Ourselin, and J. M. Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *MICCAI*, 2017. 4

[42] J. Sun, Z. Wang, S. Zhang, X. He, H. Zhao, G. Zhang, and X. Zhou. Onepose: One-shot object pose estimation without cad models. In *CVPR*, 2022. 1, 2, 3, 5

[43] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L. J. Guibas. Normalized object coordinate space for category-level 6D object pose and size estimation. In *CVPR*, 2019. 2, 5, 6, 7, 8

[44] B. Wen and K. Bekris. BundleTrack: 6D Pose Tracking for Novel Objects without Instance or Category-Level 3D Models. In *IROS*, 2021. 2

[45] B. Wen, J. Tremblay, V. Blukis, S. Tyree, T. Muller, A. Evans, D. Fox, J. Kautz, and S. Birchfield. Bundlesdf: Neural 6-DoF tracking and 3D reconstruction of unknown objects. In *CVPR*, 2023. 2

[46] C. Wu, L. Chen, S. Wang, H. Yang, and J. Jiang. Geometric-aware Dense Matching Network for 6D Pose Estimation of Objects from RGB-D Images. In *Pattern Recognition*, 2023. 1, 3, 7

[47] Y. Xiao, V. Lepetit, and R. Marlet. Few-shot object detection and viewpoint estimation for objects in the wild. *TPAMI*, 2022. 1

[48] H. Xu, S. Liu, G. Wang, G. Liu, and B. Zeng. Omnet: Learning overlapping mask for partial-to-partial point cloud registration. In *ICCV*, 2021. 2

[49] Z. Yew and G. Lee. Regtr: End-to-end point cloud correspondences with transformers. In *CVPR*, 2022. 2

[50] H. Yisheng, W. Yao, F. Haoqiang, C. Qifeng, and S. Jian. Fs6d: Few-shot 6D pose estimation of novel objects. In *CVPR*, 2022. 1, 3, 5

[51] J. Zhang, D. Ramanan, and S. Tulsiani. Relpose: Predicting probabilistic relative rotation for single objects in the wild. In *ECCV*, 2022. 2, 3, 5

[52] C. Zhou, C. Loy, and B. Dai. Extract free dense labels from clip. In *ECCV*, 2022. 4, 5