Christopher Zanoli
23-942-394
czanoli@ethz.ch

Injecting 3D Awareness in 6D Pose Estimation of Unseen Objects with Foundation Features

Practical Work Report

Master's Degree Programme in Computer Science
Photogrammetry and Remote Sensing Group
ETH Zürich

Supervisor: Prof. Dr. Konrad Schindler
Co-supervisors: Dr. Nikolai Kalischek, Dr. Fabian Manhardt

June 1, 2025

Abstract

This project explores novel methodologies for 6D object pose estimation in a train-free setting, relying solely on RGB images and 2D foundation features. In contrast to traditional approaches that require object-specific training or depth inputs, this work investigates how far general-purpose 2D visual representations can be pushed when augmented with geometric reasoning. A central focus is placed on injecting 3D awareness into these 2D features, bridging the gap between spatial understanding and appearance-based representations. By evaluating these strategies, the project aims to uncover the feasibility, strengths, and boundaries of leveraging 2D foundation models for 6D pose estimation in a setting that demands generalization across unseen objects and categories, without tailoring to specialized datasets or architectures. Experimental evaluation on three core BOP benchmark datasets (LM-O, T-LESS, and YCB-V) reveals critical insights into the application of 2D foundation models for RGB-only train-free 6D pose estimation. The results empirically demonstrate that simply adopting any foundation model is insufficient, as different architectures exhibit vastly different capabilities in retaining positional information, with some preserving spatial encoding while others focus primarily on semantic features. Furthermore, when injecting 3D awareness into 2D features through fine-tuning, the nature of the training strategy proves essential: pose estimation demands object-centric fine-tuning rather than broad scene-level approaches to achieve meaningful spatial precision. In light of these insights, future work should focus on developing foundation models that maximize both positional information retention and 3D geometric awareness, enabling the extraction of generalizable 2D-enhanced features capable of robust performance in train-free RGB-only settings.

Contents

Acronyms and Abbreviations

6D	6 Degrees of Freedom
RGB	Red Green Blue
RGB-D	Red Green Blue Depth
LiDAR	Light Detection And Ranging
CAD	Computer-Aided Design
CroCo	Cross-view Completion
DPT	Dense Prediction Transformer
ViT	Vision Transformer
PnP	Perspective-n-Point
RANSAC	RANdom SAmple Consensus
SO(3)	Special Orthogonal group in 3 dimensions
PCA	Principal Component Analysis
RoPE	Rotary Positional Embeddings
MLP	Multi-Layer Perceptron
VSD	Visible Surface Discrepancy
MSSD	Maximum Symmetry-aware Surface Distance
MSPD	Maximum Symmetry-aware Projection Distance
AR	Average Recall

Chapter 1

Introduction

Estimating the 6D pose of an object (its 3D position and 3D orientation) is a fundamental challenge in computer vision with broad applicability in fields such as autonomous driving and robotic manipulation. In industrial environments, such as assembly lines, robotic manipulators must precisely interpret an object's geometry and spatial pose to interact with it effectively. This requirement becomes even more critical during assembly operations, where components must be aligned and joined with high precision. In such scenarios, accurate 6D pose estimation is essential for achieving reliable and repeatable grasping [4, 21].

Modern robotic systems are typically equipped with various sensing modalities, including RGB cameras, depth sensors, stereo vision, and LiDAR. While depth sensors enable several applications, they also present limitations: they often fail to capture accurate data from very dark, bright, transparent, or reflective surfaces, such as glass. Stereo sensors, meanwhile, can be prone to errors due to incorrect stereo matching and triangulation inaccuracies, particularly for distant objects. LiDAR sensors, though capable of delivering 3D data, are costly to deploy and suffer from sparse outputs in environments with reduced visibility [10].

In contrast, monocular RGB cameras are inexpensive, widely available, and simple to integrate into existing systems, making them an attractive alternative. However, inferring 3D pose from 2D RGB images is an ill-posed problem, as information is lost due to the nature of the perspective projection. Despite this challenge, recent research has shown that state-of-the-art performance can be achieved using RGB-only methods, even surpassing approaches leveraging the additional depth information [8, 21]. Nevertheless, object pose estimation remains challenging in real-world scenarios due to factors such as object symmetries, repetitive textures, and occlusions, as illustrated in fig. 1.2. Symmetrical objects can yield multiple

equally plausible poses, and occlusions can obscure critical visual features necessary for precise estimation.

6D pose estimation methods can be broadly classified into model-based and model-free approaches, depending on the available information about the object of interest. Model-based methods require the 3D model of the object as input, whereas model-free methods relax this requirement and only need a set of reference views of the object [3]. Model-free methods often encounter difficulties with low-texture objects and tend to report results on limited datasets or those without occlusions. Due to these challenges, model-based unseen object pose estimation methods have garnered increasing attention [11]. Availability of the object's 3D CAD model mitigates pose ambiguity to a certain degree, facilitating more accurate and reliable monocular 6D pose estimation. An illustration of this process is provided in fig. 1.1.

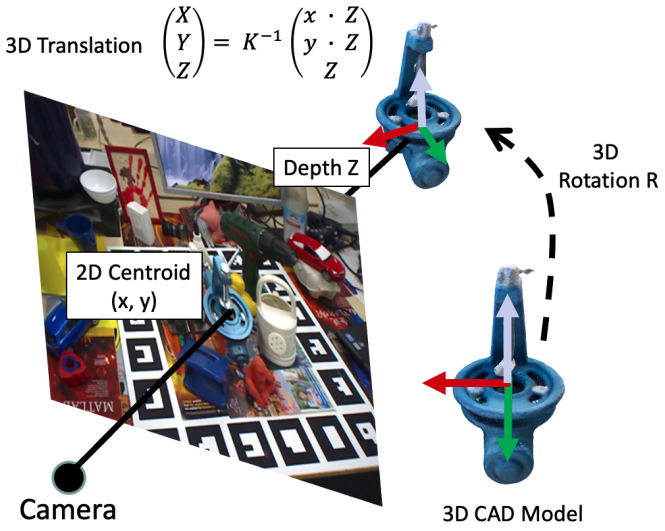

Figure 1.1: Monocular 6D Object Pose Estimation [10]. Estimation of the 6D pose of an object (its 3D orientation and 3D position) from a single RGB image. The object's 2D centroid is first detected in the image plane, and the depth Z is estimated to recover the full 3D translation vector (X, Y, Z). The transformation uses the camera intrinsic matrix K, which encodes the internal parameters of the camera (such as focal length and principal point) and maps 3D coordinates to 2D image points. Combined with a known 3D CAD model, the object's rotation R and translation t relative to the camera can be estimated.

The progression of 6D pose estimation has evolved from instance-level methods to category-level approaches and now towards foundation model integration, driven by the need for scalability that has only recently become attainable through modern computational capabilities.

Early methods focused on instance-level pose estimation, where separate models were trained for each specific object instance. While effective within their narrow scope, these methods depend on a predefined set of known objects and lack the ability to generalize to unseen instances, limiting their practical applicability.

To address this limitation, category-level methods were introduced, aiming to generalize pose estimation across different objects within the same semantic category. However, these approaches come with their own set of challenges, including significant distribution shifts between training and test domains, as well as high intra-class variability. Furthermore, they often require large-scale, finely annotated datasets, which are costly and labor-intensive to obtain. Despite these efforts, category-level models still struggle to capture the full complexity of real-world object variations [17].

Recently, foundation models have emerged as a promising direction to overcome these limitations. Their viability is closely tied to the increasing availability of computational resources, which now make it feasible to train such large-scale, general-purpose models. Foundation models are capable of learning robust, transferable representations of object pose that generalize effectively across domains and categories, with no need for task-specific training, enabling train-free applicability [3, 16].

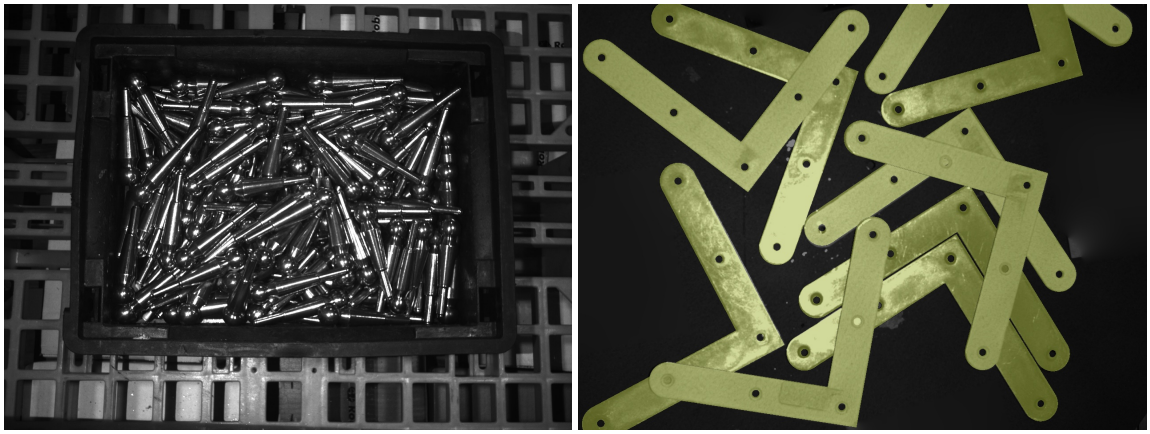

Figure 1.2: Examples of images from the XYZ-IBD and ITODD datasets of the BOP benchmark[1], showcasing realistic industrial environments. The objects in these images are often symmetric, textureless, and heavily occluded, complicating the pose estimation process.

[1]BOP benchmark

Chapter 2

Related Work

2.1 2D Vision Foundation Models

DINOv2 [15] is a self-supervised Vision Transformer (ViT) model trained on large-scale unlabeled image datasets. It learns general feature descriptors at intermediate layers that encode both semantic and positional information. As shown in [16], the model generates robust patch descriptors for a wide range of objects without requiring object-specific training, effectively handling texture variations, symmetries, and changes in viewpoint. Additionally, DINOv2 demonstrates robustness to domain shifts, with representations remaining stable when transitioning from synthetic renderings to real images. This self-supervised learning approach closes the performance gap with (weakly) supervised alternatives across a wide range of benchmarks, all without the need for fine-tuning. Key properties of DINOv2 include an understanding of object parts and scene geometry, independent of image domains, showcasing its strong performance and versatility in diverse settings [15].

CroCov2 [19] is a vision foundation model trained on large-scale datasets for 3D vision tasks. Its pretext task, cross-view completion (CroCo), involves reconstructing a partially masked input image using visible patches and an additional view of the same scene. This pretraining objective is particularly well-suited for geometric downstream tasks, as it utilizes pairs of images and requires geometric understanding of the scene to extract relevant information from the second view. With CroCov2, the pretrained model is fine-tuned on stereo matching and optical flow using a Dense Prediction Transformer (DPT) head, making a meaningful step toward creating a universal vision model capable of solving various vision tasks with a common architecture. The success of CroCov2 demonstrates that large-scale pretraining can effectively address geometric tasks, leveraging a well-adapted pretext

task and real-world data, enabling state-of-the-art performance with a ViT-based architecture without the need for task-specific designs [19].

2.2 Seen vs Unseen Object Pose Estimation

In object pose estimation, a fundamental distinction is made between seen and unseen object pose estimation. Seen object pose estimation refers to the task where the model is trained on a specific set of objects and is evaluated on objects from the same set, allowing the model to leverage prior knowledge of the object's appearance and geometry. Some of these methods regress sparse or dense features of the object, such as keypoints or correspondences, and separately recover the object's pose by applying PnP+RANSAC on the regressed features. However, since these methods are designed to encode specific objects' features into the network weights, they tend to perform poorly when encountering unseen objects.
In contrast, unseen object pose estimation deals with objects that were not part of the training set, requiring the model to generalize to new, previously unseen objects. This scenario is more challenging, as the model must rely on generalizable features rather than object-specific training data. Recent research has focused on model-based unseen object pose estimation, with several approaches exploring different techniques to address this challenge. One class of methods relies on template matching, where 3D models are rendered to create templates, and the object pose is estimated by selecting the template most similar to the query image. Alternatively, other approaches use feature-matching techniques, establishing 2D-3D or 3D-3D correspondences between the 3D model and the query image, subsequently recovering poses using the PnP algorithm.

2.3 Training Paradigms for Object Pose Estimation

Within unseen object pose estimation, methods can be categorized based on their training requirements. The first category includes methods that require a pretraining phase on synthetic datasets before being applied to novel objects. MegaPose [9] estimates object poses by rendering multiple views of a CAD model and matching them with the masked image to obtain an initial coarse pose, but requires pretraining on synthetic data. Similarly, GigaPose [13] generates templates to extract dense features using a ViT model, then performs fast nearest neighbor search in the feature space to identify the best matching template. Additionally, two lightweight

MLPs are employed to estimate 2D scale and in-plane rotation from a single 2D-2D correspondence using local features, requiring training of these MLP components. The second category comprises train-free methods, which require no training whatsoever and can be applied directly to new scenarios using only pre-trained foundation models. ZS6D [2] compares visual features from DINOv2 against a database of features from rendered object templates, followed by a RANSAC-based PnP algorithm for final pose estimation. FoundPose combines DINOv2 with bag-of-words descriptors to shortlist similar templates, and 2D-3D correspondences are established between the query and selected templates using DINOv2 patch-level features. Train-free methods generally employ segmentation masks, such as CNOS [12], to isolate object regions and subsequently match features between the query image and template using the foundation model patch descriptors. This can be viewed as a detect-and-describe framework, where the segmentation mask serves as the feature detector, and the foundation model provides the feature descriptor [11].

Chapter 3

Method

3.1 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 segmentation masks of the target object instances, together with per-mask object identity, are provided at inference time. Specifically, we obtain the masks by CNOS [12], a recent method for segmentation of unseen objects that also requires only 3D models for onboarding the objects.

3.2 FoundPose

FoundPose [16] demonstrated the strong generalization capabilities of self-supervised foundation models, specifically leveraging DINOv2. In their approach, object poses are estimated by establishing 2D-3D correspondences between input images and 3D object models. These correspondences are formed by matching image patches to pre-rendered object templates using descriptors extracted from intermediate layers of the DINOv2 network (specifically, from layer 18 of 24). To reduce computational overhead associated with matching against all possible templates, the authors proposed a template retrieval mechanism based on K-Means clustering. This method represents patch descriptors as a bag-of-words, enabling efficient identification of a small set of visually similar templates for correspondence estimation. In this work, the FoundPose pipeline served as the underlying framework for developing several modifications and integrations.

3.2.1 Object Representation and Retrieval

Given a texture-mapped 3D object model, n RGB-D templates are rendered under varying orientations. In FoundPose, these orientations are sampled to uniformly cover the $SO(3)$ space of 3D rotations [1], and rendering is performed using a standard rasterization pipeline [18] with a black background and fixed lighting. Each object is rendered from 57 distinct viewpoints, and for each viewpoint, 14 in-plane rotations are applied, resulting in a total of 798 templates per object. In contrast, our pipeline, similar to GigaPose [13] and Co-op [11], aims to minimize the number of required templates for pose estimation by focusing exclusively on out-of-plane rotations. Specifically, we generate 42 templates by rendering the CAD model from uniformly distributed viewpoints on the sphere. These viewpoints are obtained by subdividing each triangle of Blender's icosphere primitive into four smaller ones, following the method proposed in previous work [14]. Similar to FoundPose the size of the templates is $S \times S$ pixels, and the objects are rendered such that the longer side of their 2D bounding box is δS pixels long, with $\delta < 1$. At inference, crops of the query image are generated with the same size and padding (to allow for errors of segmentation masks around which the image is cropped). The bag-of-words descriptor component used for efficient template retrieval during inference remains unchanged in this work.

3.2.2 Patch Descriptors

Following the FoundPose protocol, each RGB-D template indexed by $t \in \{1, ..., n\}$ is divided into m non-overlapping image patches, and patch descriptors $\{\mathbf{p}_{t,i}\}_{i=1}^{m}$ are computed. The size of these patches depends on the backbone used for feature extraction: for FiT3D, being based on DINOv2, patches are kept at 14×14 pixels, while for CroCoV2 the patches are 16×16 pixels. Each descriptor is calculated as $\mathbf{p}_{t,i} = \phi_d(\mathbf{p}'_{t,i})$, where $\mathbf{p}'_{t,i}$ is the raw feature from the backbone and $\phi_d : \mathbb{R}^r \mapsto \mathbb{R}^d$ is a projection onto the top d principal components using PCA. The PCA step is optional. A patch is considered valid if its 2D center lies within the object mask. The template t is then represented as a set $T_t = \{(\mathbf{p}_{t,j}, \mathbf{x}_j) \mid j \in M\}$, where M contains indices of valid patches, and $\mathbf{x}_j$ denotes the 3D location (in the object model's coordinate frame) corresponding to the 2D center of patch j. These 3D coordinates are computed using the depth information from the template along with known camera intrinsics, enabling 2D-3D correspondences during inference. For final pose estimation from 2D-3D correspondences, the same crop-to-template patch matching

and pose fitting approach as used in FoundPose is adopted.

3.3 FiT3D

Current visual foundation models are trained purely on unstructured 2D data, limiting their understanding of the 3D structure of objects. Existing works mainly focus on fusing multi-view 2D features into the 3D representation, while little attention has been paid to the other direction of incorporating 3D awareness into 2D representation learning. To this regard, current methods require pre-training the 2D feature extractor, typically a ViT backbone, using hand-crafted pretext tasks. The pre-trained models are then employed for downstream tasks via fine-tuning. By contrast, FiT3D [20] aims to transfer the 3D awareness embedded in multi-view fused features to the 2D feature extractor through fine-tuning to improve DINOv2 2D foundation features. This method starts with lifting 2D image features to a 3D representation. Then the 2D foundation model is fine-tuned using the 3D-aware features. They find that 3D-aware features exhibit cleaner and more detailed feature maps compared with the original 2D features, and they show that incorporating the fine-tuned features results in improved performance on downstream tasks such as semantic segmentation and depth estimation on a variety of datasets.

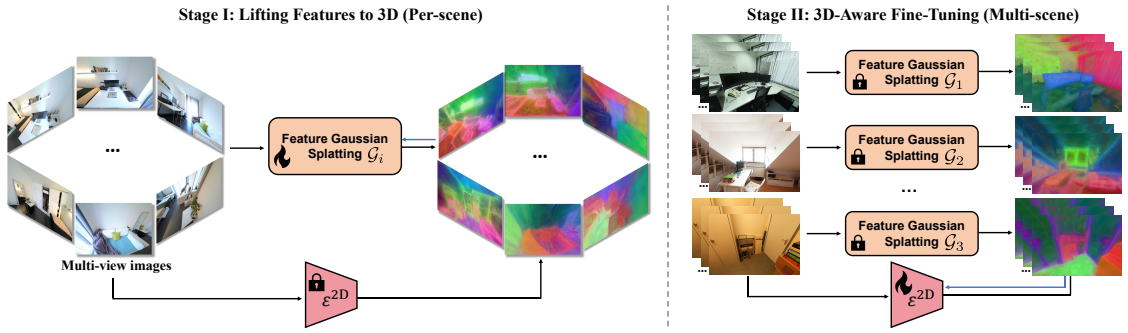

Figure 3.1: Overall FiT3D two-stage pipeline. In the first stage, foundation features are lifted into 3D-aware features by training 3D Gaussian representation $\mathcal{G}_i$. In the second stage, the rendered features are used to finetune the 2D foundation model ε_θ^{2D} . The $\rightarrow$ denotes gradient flow.

The overall 2-stages pipeline is shown in fig. 3.1. In the first stage, per-view 2D features are lifted into a multi-view consistent and 3D-aware representation, the latter obtained through an extension of Gaussian splatting. In the second stage, the obtained 3D-aware feature representations are used as training dataset to finetune the 2D feature extractor. The fine-tuning algorithm is depicted in algorithm 1. The

fine-tuning process requires training pairs of original 2D feature maps and 3D-aware feature maps. In each step of the training loop, a view from all the training images is randomly sampled, then its associated feature Gaussian and scene-specific CNN decoder are retrieved. Finally, after projecting the features into a high-dimensional feature space $\mathbf{F^{high}}$ using a simple CNN, these features are rendered as the ground truth features for fine-tuning. The fine-tuning loss is a $l1$ loss between $\mathbf{F^{high}}$ and the output features of the fine-tuned 2D feature extractor.

Algorithm 1 3D-aware fine-tuning algorithm

Require: Pre-trained Feature Gaussian representations $\{\mathcal{G}_1, ..., \mathcal{G}_K\}$, pre-trained 2D feature extractor ε_θ^{2D}, a set of images $\{\mathbf{I}_i\}_{i=1}^N$ and associated camera poses $\{\mathbf{P}_i\}_{i=1}^N$.
Ensure: Fine-tuned 2D feature extractor $\varepsilon_{\hat{\theta}}^{2D}$.
1: Load $\mathcal{G} \sim \{\mathcal{G}_1, ..., \mathcal{G}_K\}$
2: **while** fine-tuning **do**
3: Sample an image $\mathbf{I}_i$ and camera pose $\mathbf{P}_i$, $i \sim \mathcal{U}\{1, N\}$
4: Retrieve associated feature Gaussian $\mathcal{G}$ and CNN decoder d
5: Render $\mathbf{F}^{\text{high}} \leftarrow d(r^{\text{feat}}(\mathcal{G}, \mathbf{P}_i))$
6: Step θ by minimizing $\mathcal{L}(\varepsilon_\theta^{2D}(\mathbf{I}_i), \mathbf{F}^{\text{high}})$
7: **end while**
8: **return** $\varepsilon_{\hat{\theta}}^{2D}$

Motivated by the rationale behind FiT3D and its demonstrated effectiveness, this project replaces the model component of the original FoundPose pipeline with the FiT3D checkpoint. The modified pipeline is then employed in a train-free setting to evaluate whether the 3D-aware fine-tuning introduced by FiT3D provides tangible benefits for object pose estimation.

3.4 CroCov2

CroCov2 introduces a self-supervised learning framework aimed at producing transferable representations for various 3D vision and geometric tasks, such as depth estimation and optical flow. The method is based on a novel pretext task (cross-view image completion) where the model learns to reconstruct masked regions of one image using both its unmasked content and a second image of the same scene captured from a different viewpoint. Specifically, random regions of the first image are masked, and the model is trained to infer these regions by leveraging both the visible parts of that image and complementary information from a reference image.

While multi-view image completion has been previously studied in the context

of image editing, this work is among the first to explore its utility for self-supervised representation learning. Compared to single-view masked image modeling, cross-view completion enables the model to condition its predictions on geometric relationships between views, thus resolving ambiguities through spatial reasoning.

Figure 3.2 illustrates the architecture of CroCo-Stereo and CroCo-Flow. In both variants, the input image pair (stereo left/right or sequential frames) is divided into patches and encoded using transformer blocks with Rotary Positional Embeddings (RoPE). The decoding stage employs a series of transformer decoder layers, incorporating self-attention over tokens from the first image, cross-attention with tokens from the second image, and a Multi-Layer Perceptron (MLP). Features from multiple intermediate layers are passed to a DPT module to generate the final output.

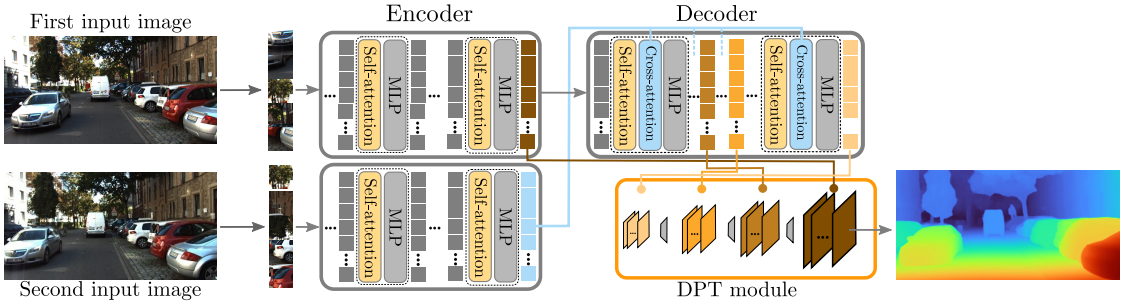

Figure 3.2: Architecture of CroCo-Stereo and CroCo-Flow.

Motivated by the underlying principles of CroCov2 and its demonstrated performance gains in monocular 3D vision tasks, this work also investigates the impact of replacing DINOv2 with CroCov2 within our processing pipeline. The modified pipeline is evaluated in a train-free setting to assess whether this substitution leads to improved, degraded, or stable performance. Additionally, we aim to analyze the underlying factors contributing to the observed outcomes.

Chapter 4

Experiments

4.1 Datasets

The experiments are conducted on three core BOP [5] datasets, namely LM-O, T-LESS and YCB-V, as these were the only ones shared across all the compared methods (see section 4.4). Each dataset provides both 3D models of the objects and test RGB-D images. **T-LESS** includes 30 industry-relevant objects that lack distinctive textures or colors. Many of the objects have symmetrical features and share similarities in shape and/or size, with some being composites of other objects. The test images are drawn from 20 different scenes, each varying in complexity. **YCB-V** features 21 everyday household objects with varying textures, colors, and shapes. The test images are captured from cluttered scenes with significant occlusions and a wide range of viewpoints and lighting conditions. **LM-O** consists of 8 textureless objects placed in cluttered environments, often with occlusions and challenging lighting. The test scenes feature multiple objects simultaneously, increasing the complexity of pose estimation. Only 3D object models and test images from these datasets were used for experiments since no training is required.

Figure 4.1: Examples from the T-LESS, YCB-V and LM-O datasets, respectively.

4.2 Experimental Setup

Evaluation Protocol. The methods' evaluation adheres to the guidelines set by the BOP Challenge 2019-2023 [6]. Specifically, methods for 6D object pose estimation are assessed using three types of pose-error metrics: Visible Surface Discrepancy (VSD), which handles ambiguous poses by evaluating only visible regions; Maximum Symmetry-Aware Surface Distance (MSSD), which measures 3D surface error using predefined global symmetries; and Maximum Symmetry-Aware Projection Distance (MSPD), which evaluates perceived deviations while accounting for object symmetries. The proportion of correctly estimated poses across annotated instances is termed Recall. The Average Recall (AR) for each metric is computed, and the overall performance of a method is quantified using their combined Average Recall: $\text{AR} = (\text{AR}_{\text{VSD}} + \text{AR}_{\text{MSSD}} + \text{AR}_{\text{MSPD}})/3$.

Compared methods. The methods of this work are compared against model-based methods that use only RGB 2D features, evaluated on the unseen object pose estimation task of the BOP Challenge 2023 [7]: MegaPose [9], GigaPose [13], FoundPose [16] and ZS6D [2]. Notably, only FoundPose, ZS6D, and our proposed methods are entirely train-free, leveraging frozen DINOv2 features for the former two and FiT3D/CroCov2 features for the latter. Since this work implements only the coarse versions of each method, we compare against coarse methods from existing approaches.

4.3 Implementation Details

Regardless of the core feature extractor (FiT3D or CroCov2) template generation involves rendering 42 templates per object. The image resolution is set to 420×420 for FiT3D and 480×480 for CroCov2, matching the input requirements of their respective backbones. FiT3D uses a patch size of 14×14, consistent with its DINOv2-based architecture, while CroCov2 uses 16×16 patches. These settings produce 30×30 patch descriptors for both the templates and query image crops in both pipelines. Optionally, these descriptors can be reduced in dimensionality via PCA, projecting them onto the top 256 principal components.

For FiT3D, only the `DINOv2_Base_Finetuned` model was available at the time of experimentation. This model is based on `DINOv2-ViT-B/14`, where `B` denotes the `Base` size in the Vision Transformer family. ViT model variants typically include `Small` (`S`), `Base` (`B`), `Large` (`L`), and `Giant` (`G`), corresponding to increasing model

capacity and parameter count. In our experiments, we extract patch descriptors from layer 9 of `DINOv2_Base_Finetuned` without applying dimensionality reduction, either during template generation or at inference time.

For CroCov2, patch descriptors are extracted from decoder layer 6. In this case, PCA is applied to reduce dimensionality both during template generation and at inference. The CroCov2 variant used is `ViTLarge_BaseDecoder`, which includes 24 encoder blocks and 11 decoder blocks. RoPE is used for positional encoding. In configurations employing the DPT module, we utilize the `PixelwiseTaskWithDPT()` class, which automatically sets the hook indices to $[5, 11, 17, 23]$ and uses a tile overlap of 0.97. The resulting DPT output is fused with the decoder-6 patch descriptors based on confidence scores: the DPT descriptor is preferred when confidence is high, while the original decoder-based descriptor is retained when confidence is low.

To support bag-of-words template retrieval, visual words are defined for each object via K-Means clustering of patch descriptors extracted from all templates. Specifically, we use 2048 cluster centroids. Bag-of-words descriptors are constructed by soft-assigning each patch to its three nearest visual words, using a Gaussian weighting with $\sigma = 10$. For each query crop, five templates are retrieved, and the object pose is estimated via 2D-3D correspondences between the query crop and the retrieved templates. Pose estimation is performed using PnP-RANSAC with a maximum of 400 iterations and an inlier threshold of 10 pixels. All methods rely on masks predicted by CNOS to localize the object within the input query image.

The source code is publicly available at the following link.

4.4 Results

Table 4.1 presents the results of our methods (CroCoPose and FiT3DPose) against other methods published up to and including FoundPose on the object pose estimation task of the BOP Challenge 2023 [7]. All methods belong to the unseen pose estimation category to ensure fair comparison with our work, and are evaluated on three core BOP benchmark datasets: LM-O, T-LESS, and YCB-V. These three datasets were selected to enable direct comparison with all prior methods, as, for instance, ZS6D lacks evaluation results on the remaining four core datasets.
Notably, CroCoPose consistently outperforms MegaPose, achieving a +4.1 improve-

ment in AR. However, CroCoPose falls short when compared to FiT3DPose and FoundPose, showing lower AR by -2.0 and -10.2, respectively. These results empirically highlight the limitations of models like CroCov2 for RGB-only pose estimation in a train-free setting. Notably, FoundPose demonstrates clear dominance over all other methods, suggesting that the intermediate feature maps of DINOv2 retain stronger positional information compared to those of CroCov2. Interestingly, although FiT3DPose employs a fine-tuned checkpoint to inject 3D awareness into DINOv2 features, it consistently underperforms FoundPose while still outperforming other methods overall. This suggests that the nature and quality of the 3D data used during fine-tuning are critical for learning meaningful 3D-aware 2D representations for pose estimation.

#	Method	Train-free	LM-O	T-LESS	YCB-V	AR
1	MegaPose [9]	✗	22.9	17.7	28.1	22.9
2	CroCoPose (ours)	✓	28.3	23.6	29.0	27.0
3	ZS6D [3]	✓	29.8	21.0	32.4	27.7
4	GigaPose [13]	✗	29.9	27.3	29.0	28.7
5	FiT3DPose (ours)	✓	31.8	22.5	32.8	29.0
6	FoundPose [16]	✓	**39.6**	**33.8**	**45.2**	**37.2**

Table 4.1: BOP datasets results (LM-O, T-LESS, YCB-V): We report the Average Recall (AR) for each of the three datasets, as well as the mean AR across them. The best AR scores are reported in **bold** font. Our results are highlighted.

Figures 4.2 and 4.3 present the behavior of FiT3DPose and CroCoPose across textured, texture-less, and symmetric objects, evaluated on the LM-O, T-LESS, and YCB-V datasets. For both methods, the first row illustrates successful pose estimations, while the second row displays representative failure cases. Each example includes: (i) the query image crop with the CNOS mask overlaid in white (top-left); (ii) the top retrieved templates (middle row); (iii) matched patch descriptors between the query and template image that resulted in the highest-quality pose prediction (bottom row); and (iv) the comparison between the ground-truth pose (in red) and the estimated coarse pose (in green, top-right).

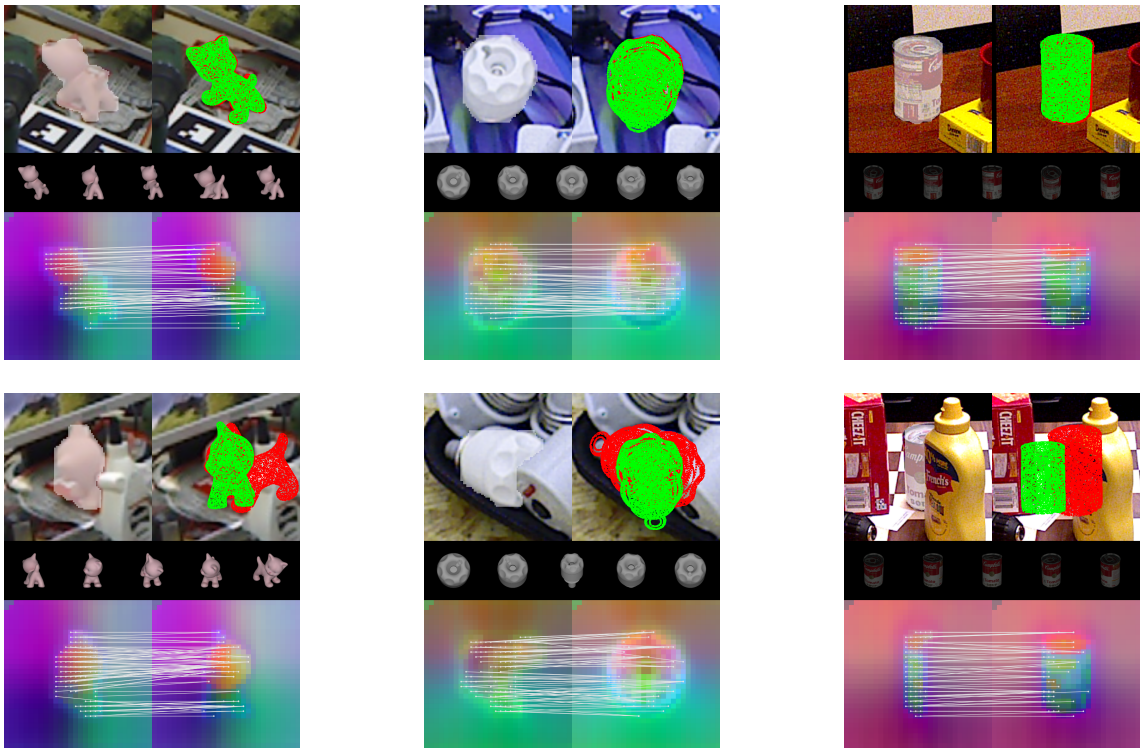

Figure 4.2: FiT3DPose results on textured (LM-O), texture-less (T-LESS), and symmetric (YCB-V) objects. Top row: successful cases; bottom row: failure cases.

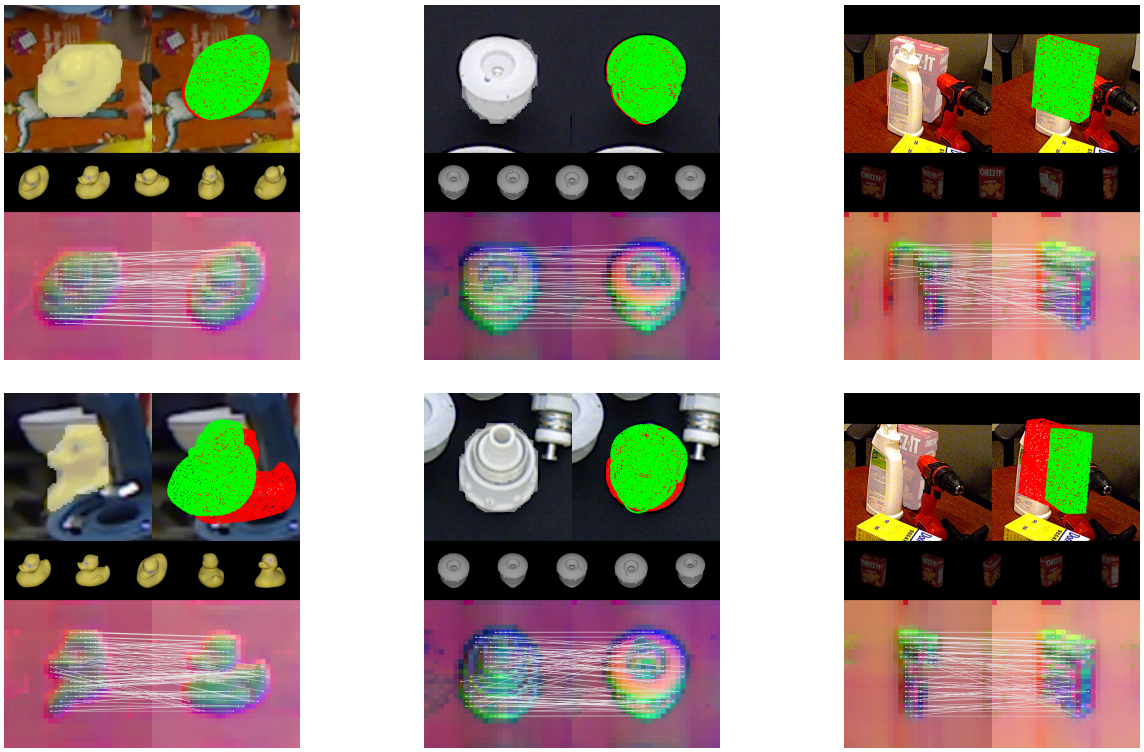

Figure 4.3: CroCoPose results on textured (LM-O), texture-less (T-LESS), and symmetric (YCB-V) objects. Top row: successful cases; bottom row: failure cases.

4.5 Ablation Study

#	Method	AR_{VSD}	AR_{MSSD}	AR_{MSPD}	AR
Feature extractor backbone					
1	FiT3DPose - layer 10	2.4	3.7	15.0	7.0
2	FiT3DPose - layer 9	20.6	25.9	48.7	31.8
3	FiT3DPose - layer 6	18.7	23.3	44.8	28.9
4	CroCoPose - decoder 10	11.3	12.4	25.7	16.5
5	CroCoPose - decoder 6	18.4	22.7	43.8	28.3
6	CroCoPose - encoder 18	5.2	5.8	15.4	8.8
7	CroCoPose - encoder 9	4.9	5.5	14.4	8.3
8	CroCoPose - decoder 6 + DPT module	15.3	18.6	38.9	24.4
Dimensionality reduction					
9	FiT3DPose - Yes PCA	19.3	24.6	47.4	30.4
10	FiT3DPose - No PCA	20.6	25.9	48.7	31.8
11	CroCoPose - Yes PCA	18.4	22.7	43.8	28.3
12	CroCoPose - No PCA	18.4	22.5	42.7	27.9
Templates generation					
13	FiT3DPose - $SO(3)$	18.3	22.3	32.2	24.3
14	FiT3DPose - upper hemisphere	20.6	25.9	48.7	31.8
15	CroCoPose - $SO(3)$	16.8	19.5	37.1	24.5
16	CroCoPose - upper hemisphere	18.4	22.7	43.8	28.3

Table 4.2: Ablation study results.

We conducted an ablation study on the LM-O dataset to evaluate the contributions of each component in the FiT3DPose and CroCoPose pipelines and to identify their optimal configurations. No pose refinement is applied in any variant. Results are reported in terms of AR_{VSD}, AR_{MSSD}, AR_{MSPD}, and overall AR. CNOS segmentation masks serve as the localization prior throughout. Table 4.2 (rows $1-4$) shows that FiT3DPose achieves the highest AR using output tokens from layer 9 of the FiT3D checkpoint with a DINOv2 ViT-B backbone. For CroCoPose (rows $5-10$), the best AR of 28.3% is obtained from decoder layer 6. We additionally experimented with incorporating the DPT module to enrich the extracted features with more explicit 3D structural information. However, this resulted in a reduced AR of 24.4%, indicating that the added complexity may not translate to improved localization performance in this context. Rows $11-14$ examine the effect of dimensionality reduction: PCA improves AR for CroCoPose but reduces it for FiT3DPose. Finally, rows $15-18$ compare two template generation strategies: one constrained to the upper hemisphere without in-plane rotations (see section 3.2.1), and another sampling the full $SO(3)$ space with in-plane rotations, as used in FoundPose.

Chapter 5

Discussion and Conclusion

As observed in fig. 4.2, FiT3DPose can handle objects with textures (the cat), without any texture (the industrial object), and those that exhibit symmetry (the can) under favorable conditions. In fact, failure cases reveal a critical limitation: under significant occlusion, FiT3DPose struggles to produce accurate results. These failures indicate that the model depends heavily on semantic information for patch matching, which becomes ambiguous when key visual regions are occluded. Consequently, the model fails to leverage the positional component of the feature descriptors robustly in such cases.
Despite this, an inspection of the matched patch descriptors from the query and the best-matching template images (bottom-left and bottom-right respectively) shows that some degree of positional information is still encoded. However, this information appears insufficient when compared to FoundPose, which is still based on DINOv2 as feature extractor backbone. A likely explanation for this discrepancy lies in the fine-tuning strategy adopted by FiT3D. The model was fine-tuned on broad, scene-level 3D reconstructions using Gaussian splatting, rather than on object-level data. As a result, the type of 3D inductive bias incorporated into the DINOv2 features may not be ideally suited for object-level pose estimation tasks. More specifically, pose estimation may benefit from a more object-centric fine-tuning that yields greater spatial precision. Moreover, the use of Gaussian Splatting for fine-tuning potentially dilute distinctive semantic features due to the "Gaussian bleeding" effect, where features from neighboring points are averaged, blurring object-specific details that are crucial for accurate pose inference.

In contrast, as observed in fig. 4.3, CroCoPose generally demonstrates a robust ability to handle object symmetries, lack of texture, and even moderate occlusions.

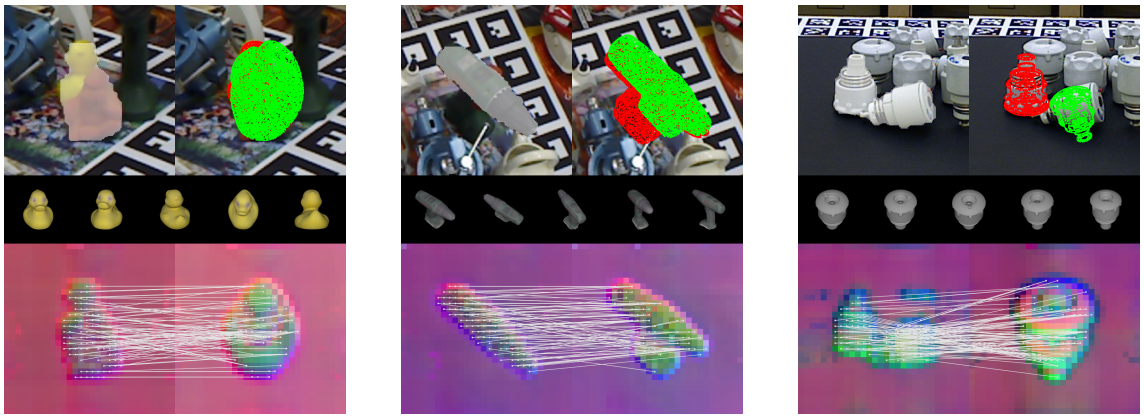

Figure 5.1: Examples of inaccurate CNOS segmentation masks that negatively impact pose estimation. Errors include merged object masks and partially missing object regions.

However, the extracted features are primarily semantic in nature, with minimal positional information. This reliance on purely semantic descriptors is a primary contributing factor to the performance degradation observed in table 4.1, and is further exemplified by the failure cases presented in the second row of fig. 4.3. In particular, the first and third failure cases illustrate how substantial occlusion can completely mislead the patch-matching process, leading to erroneous pose estimations. This issue is further corroborated by layer-wise analyses of CroCov2 shown in section 4.5, which reveal that, unlike DINOv2, CroCov2 fails to retain consistent positional information across its encoder, decoder, and DPT layers. This suggests that CroCov2 primarily encodes high-level semantic similarity, which is insufficient for precise pose alignment under challenging visual conditions.

Furthermore, an interesting case is the second failure example, in which the predicted pose is flipped by 180 degrees. This appears to be a direct consequence of the limited range of template orientations, which were generated only over the upper hemisphere of $SO(3)$. As a result, the retrieved templates lack coverage for certain rotations, highlighting the importance of using a full $SO(3)$ distribution of templates to ensure complete rotational coverage for accurate pose matching.

Finally, a notable and general limitation affecting both FiT3DPose and CroCo-Pose is their reliance on CNOS segmentation masks. As shown in fig. 5.1, several examples suffer from inaccurate or incomplete masks, which adversely affect pose estimation performance. For instance, in the first example, the CNOS mask erroneously segments both the chimpanzee and the duck objects together. Similarly, in

the third example the segmentation includes two different industrial objects. The second example shows a partial segmentation, where the drill's base and grip are entirely excluded from its mask. These segmentation inconsistencies introduce substantial noise into the pose estimation pipeline. This issue has also been recognized in prior work such as Co-Op [11], which instead employed a detector-free correspondence estimation framework for the coarse estimation stage. They showed that their learned object regions are more accurate than CNOS masks, highlighting the need to reduce reliance on segmentation masks in order to achieve more accurate pose estimation results.

In conclusion, this work has investigated the application of 2D foundation models for RGB-only train-free 6D pose estimation, revealing fundamental insights into the capabilities and limitations of current approaches. The experimental evaluation demonstrates that the choice of foundation model backbone is not arbitrary as different architectures exhibit dramatically different abilities to retain positional information alongside semantic understanding. While DINOv2-based approaches like FoundPose preserve strong spatial encoding that proves crucial for accurate pose estimation, models like CroCoV2 primarily capture high-level semantic similarity, which is insufficient for precise spatial alignment under challenging visual conditions. The comparative analysis between FiT3DPose and FoundPose further illuminates the critical importance of 3D awareness injection strategies. Despite FiT3DPose's explicit fine-tuning to incorporate 3D information into DINOv2 features, it consistently underperforms the vanilla DINOv2-based FoundPose. This counterintuitive result highlights that the nature and scope of fine-tuning are paramount: pose estimation demands object-centric training that preserves spatial precision, rather than broad scene-level approaches that may dilute distinctive spatial features through effects like "Gaussian bleeding". These insights point toward a critical need for future work to develop 2D foundation models that can simultaneously maximize positional information retention and 3D geometric awareness, enabling accurate and generalizable train-free RGB-only 6D pose estimation.

Bibliography

[1] Marc Alexa. Super-fibonacci spirals: Fast, low-discrepancy sampling of so(3). In 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 8281–8290, 2022.

[2] Philipp Ausserlechner, David Haberger, Stefan Thalhammer, Jean-Baptiste Weibel, and Markus Vincze. Zs6d: Zero-shot 6d object pose estimation using vision transformers, 2023.

[3] Andrea Caraffa, Davide Boscaini, Amir Hamza, and Fabio Poiesi. Freeze: Training-free zero-shot 6d pose estimation withÂ geometric andÂ vision foundation models. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, Computer Vision – ECCV 2024, pages 414–431, Cham, 2025. Springer Nature Switzerland.

[4] Xinke Deng, Yu Xiang, Arsalan Mousavian, Clemens Eppner, Timothy Bretl, and Dieter Fox. Self-supervised 6d object pose estimation for robot manipulation. In 2020 IEEE International Conference on Robotics and Automation (ICRA), pages 3665–3671, 2020.

[5] Tomáš Hodaň, Frank Michel, Eric Brachmann, Wadim Kehl, Anders Glent Buch, Dirk Kraft, Bertram Drost, Joel Vidal, Stephan Ihrke, Xenophon Zabulis, Caner Sahin, Fabian Manhardt, Federico Tombari, Tae-Kyun Kim, Jiří Matas, and Carsten Rother. Bop: Benchmark for 6d object pose estimation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, Computer Vision – ECCV 2018, pages 19–35, Cham, 2018. Springer International Publishing.

[6] Tomáš Hodaň, Martin Sundermeyer, Bertram Drost, Yann Labbé, Eric Brachmann, Frank Michel, Carsten Rother, and Jiří Matas. Bop challenge 2020 on 6d object localization. In Adrien Bartoli and Andrea Fusiello, editors, Computer Vision – ECCV 2020 Workshops, pages 577–594, Cham, 2020. Springer International Publishing.

[7] Tomas Hodan, Martin Sundermeyer, Yann Labbe, Van Nguyen Nguyen, Gu Wang, Eric Brachmann, Bertram Drost, Vincent Lepetit, Carsten Rother, and Jiri Matas. Bop challenge 2023 on detection, segmentation and pose estimation of seen and unseen rigid objects, 2024.

[8] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In 2017 IEEE International Conference on Computer Vision (ICCV), pages 1530–1538, 2017.

[9] Yann Labbé, Lucas Manuelli, Arsalan Mousavian, Stephen Tyree, Stan Birchfield, Jonathan Tremblay, Justin Carpentier, Mathieu Aubry, Dieter Fox, and Josef Sivic. Megapose: 6d pose estimation of novel objects via render & compare, 2022.

[10] Fabian Manhardt. Towards Monocular 6D Object Pose Estimation. PhD thesis, Technische Universität München, 2021.

[11] Sungphill Moon, Hyeontae Son, Dongcheol Hur, and Sangwook Kim. Co-op: Correspondence-based novel object pose estimation, 2025.

[12] Van Nguyen Nguyen, Thibault Groueix, Georgy Ponimatkin, Vincent Lepetit, and Tomas Hodan. Cnos: A strong baseline for cad-based novel object segmentation. In 2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), pages 2126–2132, 2023.

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

[14] Van Nguyen Nguyen, Yinlin Hu, Yang Xiao, Mathieu Salzmann, and Vincent Lepetit. Templates for 3d object pose estimation revisited: Generalization to new objects and robustness to occlusions, 2022.

[15] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024.

[16] Evin Pınar Örnek, Yann Labbé, Bugra Tekin, Lingni Ma, Cem Keskin, Christian Forster, and Tomas Hodan. Foundpose: Unseen object pose estimation withÂ foundation features. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, Computer Vision – ECCV 2024, pages 163–182, Cham, 2025. Springer Nature Switzerland.

[17] Caner Sahin, Guillermo Garcia-Hernando, Juil Sock, and Tae-Kyun Kim. Instance- and Category-Level 6D Object Pose Estimation, pages 243–265. Springer International Publishing, Cham, 2019.

[18] Dave Shreiner. OpenGL Programming Guide: The Official Guide to Learning OpenGL, Versions 3.0 and 3.1. Pearson Education, 2009.

[19] Philippe Weinzaepfel, Thomas Lucas, Vincent Leroy, Yohann Cabon, Vaibhav Arora, Romain Brégier, Gabriela Csurka, Leonid Antsfeld, Boris Chidlovskii, and Jérôme Revaud. Croco v2: Improved cross-view completion pre-training for stereo matching and optical flow, 2023.

[20] Yuanwen Yue, Anurag Das, Francis Engelmann, Siyu Tang, and Jan Eric Lenssen. Improving 2d feature representations by 3d-aware fine-tuning, 2024.

[21] Guangyao Zhai, Dianye Huang, Shun-Cheng Wu, HyunJun Jung, Yan Di, Fabian Manhardt, Federico Tombari, Nassir Navab, and Benjamin Busam. Monograspnet: 6-dof grasping with a single rgb image. In 2023 IEEE International Conference on Robotics and Automation (ICRA), pages 1708–1714, 2023.