ÉCOLE NATIONALE DES PONTS ET CHAUSSÉES

IP PARIS

THÈSE DE DOCTORAT

de l'École Nationale des Ponts et Chaussés

Pose Estimation for Novel Rigid Objects

École doctorale N°532, Mathématiques et Sciences et Technologies de l'Information et de la Communication (MSTIC)

Spécialité de doctorat : Informatique

Thèse préparée au sein de l'équipe A3SI (Imagine, École Nationale des Ponts et Chaussés) du Laboratoire d'Informatique Gaspard Monge (LIGM)

Thèse soutenue le 19 décembre 2024, par

Van Nguyen NGUYEN

Composition du jury :

Josef SIVIC Directeur de recherche, CTU Prague	*Président*
Markus VINCZE Professeur, TU Wien	*Rapporteur*
Benjamin BUSAM Chercheur, TU Munich	*Rapporteur*
Dima DAMEN Professeur, University of Bristol / Google DeepMind	*Examinatrice*
Slobodan ILIC Chercheur, TU Munich / Siemens	*Examinateur*
Vincent LEPETIT Professeur, ENPC	*Directeur de thèse*

Ce manuscrit est dédié à ma grande-mère Thi Truong NGUYEN.

Abstract

Object pose estimation is an important computer vision problem as it has great impact in many applications such as robotics, augmented reality, and autonomous driving. While existing supervised object pose estimation methods have achieved remarkable performance, they heavily rely on extensive training data specific to each target object, and cannot generalize to novel objects. In this thesis, we address this problem of generalization, and propose scalable methods for detecting, segmenting, and estimating the 6D pose of novel objects in RGB images. Our methods require only the 3D model or a single reference image of test objects for inference, making them suitable for practical scenarios in real-world applications.

Our methods are grounded on the concept of "templates", which are 2D views of target 3D objects. The idea of using templates for object pose estimation is not new: the first work on this seems to be Murase's and Nayar's in 1995. We nevertheless show that, thanks to recent advances in machine learning, in particular on unsupervised image features, templates can have very interesting properties to generalize to unseen objects efficiently: They do not require training while adapting to new objects; they can be matched extremely fast; they can be robust to occlusions thanks to high dimensional local image features.

Our first method is CNOS, a simple yet powerful method for segmenting novel objects in RGB images from their 3D models. CNOS generates segmentation proposals from the input image using Segment Anything and matching them with templates rendered from the 3D models and represented with the foundation features of DINOv2. CNOS significantly outperforms existing methods on multiple datasets and has been awarded as the best method for 2D detection/segmentation of unseen object in the BOP challenge 2023. CNOS is also used as the default detection method for both model-based, model-free unseen object pose estimation at the BOP challenge 2024.

We then reconsider in details existing template matching methods for pose estimation and analyse their limitations when dealing with novel objects. We introduce a new approach that matches the input testing image with templates from their 3D models. Our approach is trained on a limited number of reference objects with contrast learning and generalizes well to novel objects.

We further extend this work in a method we call GigaPose, a novel approach that is significantly faster and more robust against input segmentation errors. GigaPose samples templates only on two degrees of freedom (DoFs) for estimating out-of-plane rotation, then uses feature matching to estimate the remaining four DoFs. GigaPose seamlessly integrates with existing refinement methods. It achieves state-of-the-art results for RGB-based methods on the standard BOP benchmark while is 35 times faster comparing to existing methods for the coarse pose estimation stage.

Finally, we present NOPE, a simple method to estimate novel object pose from a single reference image. Our approach takes a single image of a new object as input and predicts the relative pose of this object in new images without prior knowledge of the object's 3D model. We achieve this by training a model to directly predict discriminative embeddings that resemble templates for viewpoints surrounding the object. This prediction is done using a simple U-Net architecture with attention and conditioned on the desired pose, which yields extremely fast inference. We compare our approach to state-of-the-art methods and show it outperforms them both in terms of accuracy and robustness on both synthetic and real-world datasets.

Résumé

L'estimation de pose d'objets est un problème important en vision par ordinateur, car elle a un grand impact sur de nombreuses applications telles que la robotique, la réalité augmentée et la conduite autonome. Bien que les méthodes existantes d'estimation de pose d'objets supervisées aient atteint des performances remarquables, elles dépendent beaucoup de données d'entraînement spécifiques à chaque objet et ne peuvent pas généraliser à de nouveaux objets qui n'ont pas été vus pendant l'entraînement. Dans cette thèse, nous abordons ce problème de généralisation et proposons des nouvelles méthodes pour détecter, segmenter et estimer la pose 6D pour des nouveaux objets dans des images RGB. Nos méthodes n'utilisent que le modèle 3D ou une seule image de référence des objets de test pour l'inférence, permettant des scénarios pratiques dans des applications réelles.

Nos méthodes sont basées sur le concept de "templates", qui sont des vues 2D d'objets 3D. L'idée d'utiliser des templates pour l'estimation de la pose d'objets n'est pas nouvelle : le premier travail à ce sujet est celui de Murase et Nayar en 1995. Cependant, nous montrons que, grâce aux récents progrès en apprentissage automatique, en particulier sur les caractéristiques d'images non supervisées, les templates peuvent avoir des propriétés très intéressantes pour se généraliser efficacement à des objets jamais vus auparavant : ils ne nécessitent pas d'entraînement pour s'adapter à de nouveaux objets ; ils peuvent être appariés extrêmement rapidement ; ils peuvent être robustes aux occultations grâce à des caractéristiques locales d'image.

Les principales contributions de cette thèse sont les suivantes. Premièrement, nous introduisons CNOS, une méthode simple mais efficace pour segmenter de nouveaux objets dans des images RGB à partir de leurs modèles 3D. Notre méthode génère des hypothèses de segmentation à partir de l'image d'entrée en utilisant Segment Anything et les compare avec des "templates", ici des images rendues des modèles 3D en utilisant les caractéristiques DINOv2. CNOS surpasse de manière significative les méthodes existantes sur plusieurs jeux de données et a été récompensée comme la meilleure méthode pour la détection/segmentation 2D d'objets inconnus dans le BOP Challenge 2023. CNOS est aussi utilisée comme la méthode de détection par défaut pour le BOP challenge 2024.

Deuxièmement, nous revisitons les méthodes de correspondance de templates existantes et démontrons leurs limites lorsqu'il s'agit de nouveaux objets. Nous introduisons ensuite une nouvelle approche qui associe l'image de test d'entrée avec des templates de leurs modèles 3D. Notre approche est entraînée sur un petit nombre d'objet de référence avec un apprentissage par contraste et se généralise bien à de nouveaux objets.

Troisièmement, nous introduisons GigaPose, une nouvelle approche qui est significativement plus rapide et plus robuste contre les erreurs de segmentation en entrée. GigaPose échantillonne des templates uniquement sur deux degrés de liberté (DoFs) pour estimer la rotation hors-plan, puis utilise la correspondance de caractéristiques pour estimer les quatre DoFs restants. GigaPose peut s'intégrer parfaitement avec les méthodes de raffinement existantes et atteindre des résultats à la pointe de la technologie sur les jeux de données standards BOP tout en étant 35 fois plus rapide que les méthodes existantes dans la phase d'estimation de pose grossière.

Quatrièmement, nous présentons NOPE, une méthode simple pour estimer la pose d'objet nouveaux à partir d'une seule image de référence. Notre méthode prend

une seule image d'un nouvel objet comme entrée et prédit la pose relative de cet objet dans de nouvelles images sans connaissance préalable du modèle 3D de l'objet. Nous y parvenons en entraînant un modèle à prédire directement des embeddings discriminatifs pour des points de vue entourant l'objet. Cette prédiction est effectuée en utilisant une architecture U-Net simple avec attention et conditionnée par la pose souhaitée, ce qui permet une inférence extrêmement rapide. Nous comparons notre approche aux méthodes éxistant et montrons qu'elle les surpasse en termes de précision et de robustesse sur des jeux de données synthétiques et réels.

Acknowledgements

I would like to thank everyone who has supported me on this scientific and living abroad adventure. The successful completion of this thesis would not have been possible without your help and inspiration.

Vincent, thank you for always being a great and very cool advisor. I feel extremely lucky and could not have hoped for a better mentor. Discussing with you was always enjoyable and full of interesting research directions. Your continuous guidance and encouragement through COVID, good and bad times, on regular days or during deadlines, have been key to the success of this journey. I also learned a lot and was greatly inspired by your expertise in research and fast-paced but rigorous writing. Thank you!

Thibault, thank you for your great help and collaboration. I am grateful to have had you in the projects. Discussing with you despite our 9-hour timezone difference has always been fun and very helpful.

Mathieu, Ko, thank you for your support and for welcoming me into your teams at EPFL and Kyoto University. Living for a few months in beautiful Switzerland and Japan are among the best memories of this journey that I will always cherish.

Pierre, thank you for hosting me on the Surreal team and for sharing your insights both as a researcher and as an exceptional human being. Being at Meta and working alongside so many great researchers and on Aria glasses is something I could not have imagined at the beginning of this PhD.

Tomáš, first, thank you for creating BOP; it has saved not only my life but also those of many other students and researchers in the field. Second, thank you for your help and inviting me to the BOP team. I could not be more excited to be back at working Meta with you and Christian.

Michaël, Xi, Yang, thank you so much for welcoming me to the lab from the internship to the PhD, and for teaching me a lot during the early stages.

Thank you to all my friends, labmates, colleagues at ENPC, EPFL, Kyoto University, Meta, NVIDIA, ENS Paris-Saclay, INSA Toulouse, Hue Val de Loire K6, Nhà 170, whom I met in France, Switzerland, Japan, USA, Vietnam, or elsewhere. While I would like to list all your names here, I must refrain to avoid the risk of exposing myself in case of forgetting some of you. No matter what, we have shared enjoyable discussions, travels, coffee breaks, lunches, dinners, beers, football, and more together. Your positive influence, directly or indirectly, has impacted me in many different ways.

Alexis, merci d'être toujours mon vrai ChatGPT depuis le début de mon aventure en France. Ton soutien inconditionnel pour m'apprendre à parler français, à m'intégrer à la culture française, à trouver un stage ainsi que pour m'accompagner jusqu'au diplôme à l'INSA et me motiver à faire le MVA, tout cela je te le dois.

Jacques, merci d'être toujours notre meilleur ami pour Dong, Toan et moi, que ce soit à Toulouse ou à Paris, et surtout de nous avoir logés et nourris à Toulouse pendant la période de COVID.

Cuối cùng, con/anh/em muốn cảm ơn gia đình, những người quan trọng nhất trong cuộc đời con/anh/em. Cảm ơn em, Vy đã luôn bên anh lúc vui cũng như lúc buồn. Cảm ơn ba mẹ, Bin, chị Ngọc, ông mệ, đặc biệt là mệ Ngoại luôn là nguồn động lực, người động viên và nuôi nấng con nhưng lúc quan trọng nhất trên chặng đường học tập. This thesis is dedicated to you!

Contents

Chapter 1

Introduction

1.1 Problem statement

The goal of this thesis is to develop methods for detecting, segmenting, and estimating the pose of novel rigid objects in RGB images without retraining. The pose of a rigid object is defined by 6 degrees of freedom (DoFs), composed by a 3D rotation and a 3D translation relative to a reference coordinate frame, located, e.g., at the camera. Because of these 6 DoFs, the object pose is also called “6D object pose" in the literature.

Existing supervised methods on 6D object pose estimation [81, 115, 77, 130, 78, 157, 108, 58, 71, 141] have shown impressive performance; however, they heavily rely on large amounts of annotated training data of target objects. Introducing new objects unseen during training requires a significant effort to synthesize or annotate data and retrain the model, thereby limiting their practicality in industrial applications. For instance, in a logistics warehouse, it may be impractical to retrain the pose estimation method for every new product.

To address this issue, category-level pose estimation methods [143, 21, 76, 89] have been proposed, which can handle unseen instances from the same categories as those in the training data. Nonetheless, these methods still struggle with completely unknown object categories. Another potential solution is to train keypoint-based approaches to estimate generic keypoints for all objects from different categories [165]. However, these methods require well-defined keypoints on 3D models, which are generally sensitive to object appearances and shape variations.

In this thesis, we aim to address this problem of generalization by developing scalable methods that can handle previously-unseen objects, thus saving both training and data capture time. We define three stages for our methods:

- **Training stage**: The method is provided a set of RGB-D training images showing annotated objects with ground-truth 6D poses and 3D CAD models. The images can be captured in real-world environments or synthetically generated using softwares such as BlenderProc [26]. This stage typically takes several hours or days.

- **Onboarding stage**: The method is then provided a 3D CAD model or a reference image of test objects that were not seen during training. For this stage, the method can spend few minutes on a single GPU, e.g., up to 5 minutes in BOP challenge [51]. The time is measured from the point right after the raw data

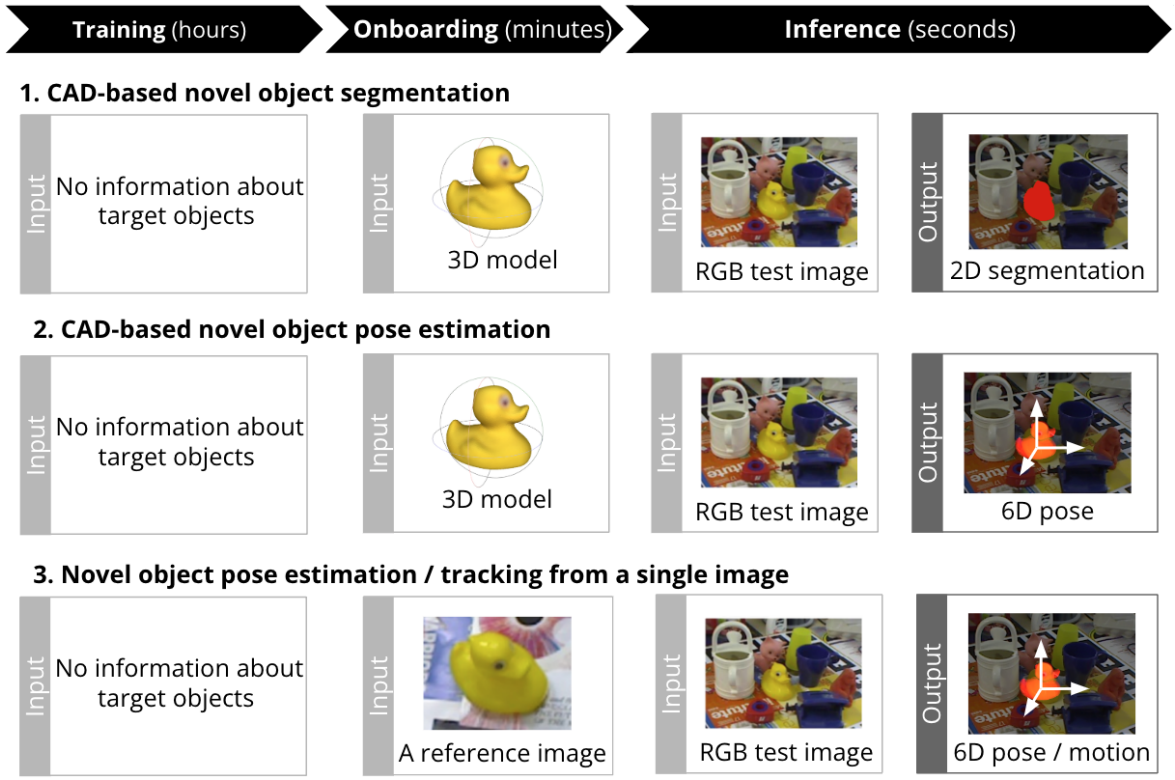


Figure 1.1: **Overview of the settings considered in this thesis.** During the training stage, we assume that the methods do not have any prior information about the target objects. Given a CAD model or a single reference image of test objects for a very short "onboarding stage" within few minutes, our goal is to segment the object, or estimate its 6D pose, or track its 6D motion during the inference.

(CAD models or reference images) is loaded to the point when the method is ready to run the inference.

- **Inference stage**: The method receives a real-world RGB image unseen during training and outputs a 6D pose.

We consider several practical testing scenarios where objects and their categories are not seen during training, only a 3D CAD model or a reference image is available at the onboarding and inference stages as shown in Figure 1.1.

1.2 Applications

Object pose estimation has great impact in many applications such as robotics, augmented reality, and autonomous driving. We discuss each of these applications in details below.

- **Robotics**: Two main downstream applications of object pose estimation in robotics are object grasping and navigation, as shown in Figure 1.2a. For instances, in warehouses, or assembly systems, the robot needs to detect objects and estimates their objects in order to grasp them. Similarly, the robot needs to

localize all objects in the 3D scene to have a representation of the environment, and avoid obstacles when navigating.

- **Augmented reality (AR)**: AR enables us to seamlessly integrate virtual objects into the real world, commonly represented by images. To achieve this, synthetic elements must be accurately rendered and aligned with the real-world images, where object pose estimation plays an important role. We show in Figure 1.2b an example of an AR application using FoundationPose [147].

- **Autonomous driving**: A self-driving car benefits from advances in 3D object pose estimation since it requires accurate object detection and pose estimation for a vehicle to navigate in the 3D space without human assistance. We show an example of 3D object detection using Cube R-CNN [10] in Figure 1.2c.

1.3 Challenges

While detecting, segmenting, and estimating 6D poses of novel rigid objects is important, well-motivated problem, there are, however, several challenges that need to be addressed:

- **Generalization to novel objects**: The scalability of object pose estimation methods is an important factor for most industrial applications. However, learning-based approaches, commonly used in the industry, have a significant limitation: they cannot generalize to new objects that are not seen during training. Handling novel objects is not only the main goal but also the main challenge in this thesis.

- **Robustness to domain gap**: Capturing real images of training objects under different lighting, illumination conditions and annotating them with 6D object poses requires considerable human effort. To address this issue, we use a similar practical approach used in object pose estimation for known objects–synthesizing the training data using rendering engines, such as BlenderProc [26]. In particular, we use the synthetic images by rendering large-scale CAD model databases, such as ShapeNet [16] or Google Object Scan [28]. Being able to adapt from synthetic to real-world data is thus also important in this task. We show in Figure 1.3a examples of synthetic training images and real-world testing images.

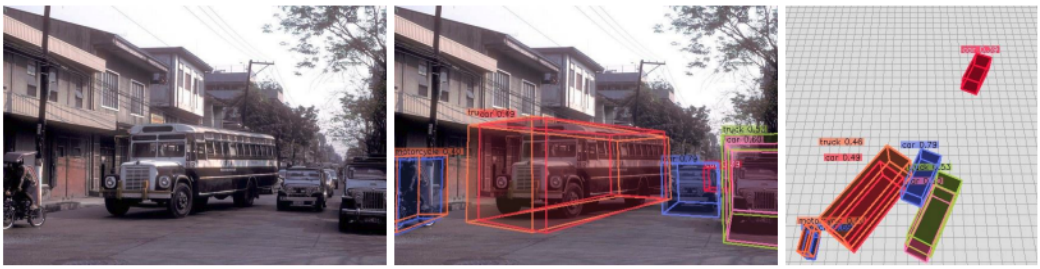
- **Robustness to clutter and partial occlusions:** Object pose estimation methods often fail in cluttered environments and when objects partially occluded by other objects. This is because the irrelevant information from the background or from other objects can confuse the method. We show an examples of this challenge in Figure 1.3b.

- **Robustness to viewpoint and illumination changes**: Changes in the viewpoint and illumination conditions can significantly affect the appearance of objects. Developing algorithms that are robust to these variations is important for consistent performance across different environments. We show an example of this challenge in Figure 1.3c.

(a) (b)

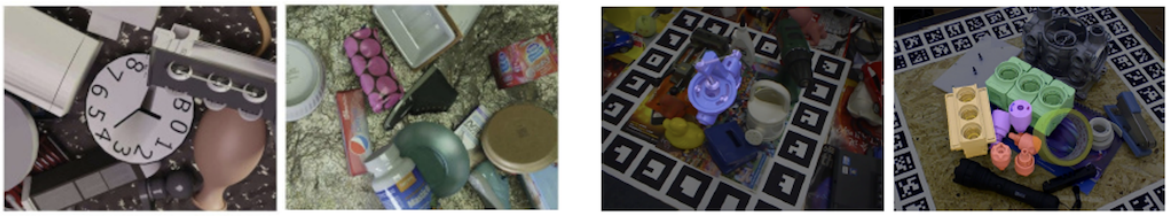
(a) **Examples of robotics applications:** (a) grasping objects in warehouses [32], (b) robot navigation systems [42].

(b) **Example of AR applications:** Given an input image sequence, we can replace a book (bottom right) with an AR wooden labyrinth game [147].



(c) **Example of applications in autonomous driving:** Given an input RGB image, we can use Cube R-CNN [10], a 3D object detection method to detect the cars in the streets and their 3D pose.

Figure 1.2: **Examples of object pose estimation applications** in (a) robotics, (b) augmented reality (AR), and (c) autonomous driving.

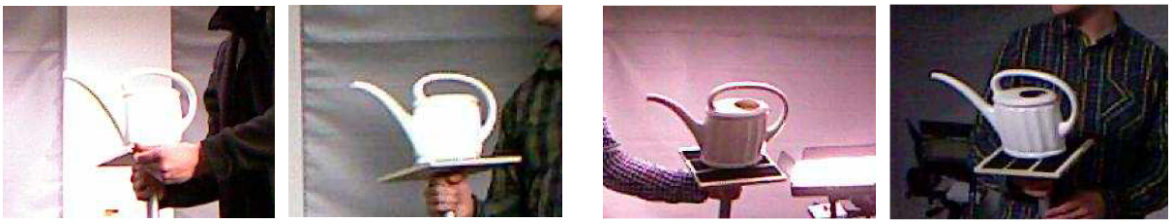


(i) (ii)

(a) **Challenges of generalization to novel object and domain gap.** We show (i) synthetic training images generated using CAD models from ShapeNet [16] or Google Object Scan [28] datasets, generated by BlenderProc [26], (ii) real-world testing images from popular datasets Linemod [46] and T-LESS [54]. This figure shows both synthetic-real domain gap and challenge of generalization on novel objects.

(i) (ii)

(b) **Challenges of cluttered background and occlusions**: (i) cluttered environment in Linemod [46], (ii) heavy occlusions for testing objects highlighted in red circle in T-LESS [54].



(c) **Challenges of illumination and viewpoints.** We show four different images of the same object under different illumination conditions from TUD-L dataset [55].

(i) (ii)

(d) **Challenges of textureless objects**. We show examples of (i) textured objects in YCB-V [150] and HOPE [134] datasets, (ii) textureless objects in T-LESS [54] and ITODD [30] datasets.

Figure 1.3: **Challenges of object pose estimation for novel objects:** (a) generalization to unseen objects and domain gap, (b) cluttered background and occlusions, (c) illumination and viewpoints, and (d) textureless objects.

- **Robustness to the lack of texture**: Common objects without discriminative textures can be found in households, offices, and industrial environments. However, these objects pose challenges for methods that rely on 2D local features because feature detectors often struggle to detect reliable descriptors on these objects. We compare textured objects and textureless objects in Figure 1.3d.

1.4 In defense of templates for object pose estimation

When aiming to solve the challenges raised by pose estimation of novel objects as detailed above, we discovered that "templates", which are 2D views of target 3D objects are a very useful concept.

The idea of using templates for object pose estimation is not new: Murase and Nayar in 1995 [99] relied on templates coded in the form of small grayscale images. However, templates are often seen as slow and not robust to partial occlusions. This is in fact now a misconception thanks to the advance of hardware and of machine learning. They do provide a strong answer to the challenges raised in Section 1.3:

Generalization to novel objects. Templates allow to naturally generalize to novel objects, by extending the set of templates. By using discriminative learning, we can represent templates with discriminative image features. Matching input images to templates can be seen as a Nearest Neighbor classifier, which is a very strong classification method when the number of prototypes is large enough. Moreover, computing the templates is in practice extremely fast, as it is only a matter of rendering views, computing their image features, and storing them.

Templates can thus also be seen as "short term memory", in the sense introduced by LeCun in his "Path Towards Autonomous Machine Intelligence" [156]. Templates allow to very quickly learn novel objects by storing them. However, the number of templates grows linearly with the number of objects, and simply storing these templates could become intractable at some point. In a complete system, one could imagine training a deep network, which scales better with the number of objects, once time is available for training.

Robustness to domain gap. To generate the templates, we use standard rendering techniques in the case of our CNOS and GigaPose methods, and image prediction by a deep network in the case of our NOPE method. In practice, the input images differ to some extend with the generated templates, because of illumination differences and the domain gap between real and synthetic images. These differences could interfere with the matching of the templates.

Instead of representing templates as regular images as in [99], we represent them as image features computed with a composition of unsupervised learning and discriminative learning. Unsupervised learning gives us robustness to domain gaps. Discriminative learning improves the accuracy of the matching.

Robustness to clutter and partial occlusions. Robustness to clutter is obtained thanks to discriminative learning.

More complicated to achieve is robustness to partial occlusions. It is well known that image intensity correlation is very sensitive to partial occlusions. We represent

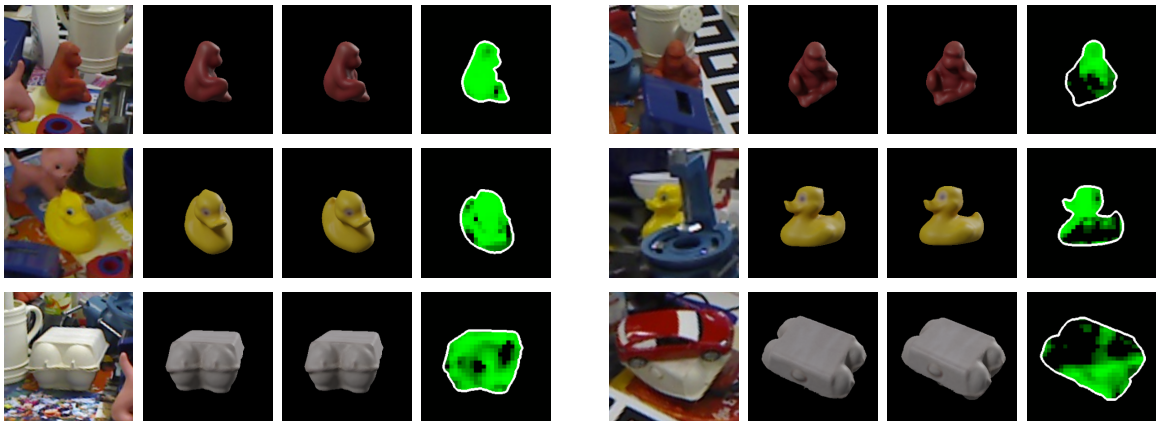


Figure 1.4: **Robustness against occlusions of template matching approaches for object pose estimation.** We show examples on Occlusion-LINEMOD without occlusion (left) and with occlusion (right). The results demonstrate that using local feature vectors minimizes the impact of partial occlusion on the evaluation of similarity between a template and an image when applying our template matching method [104] as discussed in Section 1.4.

Figure 1.5: **Challenges of object pose estimation for novel objects:** (a) generalization to unseen objects and domain gap, (b) cluttered background and occlusions, (c) illumination and viewpoints, and (d) textureless objects.

however templates and input images not with intensities but with relatively large local feature vectors. Similarity between a template and an input image is thus computing as the sum of the dot products between high dimensional vectors. If such a vector is corrupted, because of a partial occlusion for example, the dot product tends to be close to 0: Two random vectors tend to be orthogonal to each other. Thus, a partial occlusion does not penalize much the evaluation of similarity between a template and an image.

Robustness to the lack of texture. One of the initial motivations for using templates in object pose estimation is that templates capture the appearance of an object as a whole. This contrasts with approaches based on local features such as interest points. As such, templates work better than these approaches on untextured objects.

It should also be noted that computing the similarity between an input image and the templates is very fast, as it can be done in batches. This contrasts with much slower approaches such as RelPose [158], which need to apply a deep network multiple times—once for each discretized pose.

1.5 Thesis outline

This thesis is organized as follows:

- Chapter 2. We review existing methods for 6D object pose estimation and related tasks for both seen and unseen objects.

- Chapter 3. We present CNOS, a method that can segment novel objects in RGB images from only their CAD models. This method can then be used to discard the rest of the image and to provide an input 2D bounding box to the methods introduced afterwards in the thesis. CNOS has been awarded as the best method for 2D detection/segmentation of unseen object in the BOP challenge 2023. CNOS is also used as the default detection method for BOP challenge 2024.
- Chapter 4. We revisit templates for object pose estimation and introduce a novel method that can generalize to novel objects and robust to occlusions.
- Chapter 5. We further extend the work presented in Chapter 4 with GigaPose. GigaPose combines templates and 2D-to-2D correspondences to significantly improve the pose accuracy while reducing the required number of templates and being 35× faster comparing to existing methods in coarse pose estimation stage.
- Chapter 6. Finally, we present NOPE, a novel method we developed for estimating 6D pose of novel objects from a single reference RGB image instead of a complete 3D model. NOPE is able to provide a distribution over the 3D rotation space rather than a single pose as this task is inherently ambiguous.
- Chapter 7. We conclude the thesis and suggests topics for future work.

1.6 Contributions

This thesis covers the following peer-reviewed accepted publications (ordered by years):

[104] **Van Nguyen Nguyen**, Yinlin Hu, Yang Xiao, Mathieu Salzmann, Vincent Lepetit. *Templates for 3D Object Pose Estimation Revisited: Generalization to New Objects and Robustness to Occlusions*. Computer Vision and Pattern Recognition (CVPR), 2022.

Abstract: We present a method that can recognize new objects and estimate their 3D pose in RGB images even under partial occlusions. Our method requires neither a training phase on these objects nor real images depicting them, only their CAD models. It relies on a small set of training objects to learn local object representations, which allow us to locally match the input image to a set of "templates", rendered images of the CAD models for the new objects. In contrast with the state-of-the-art methods, the new objects on which our method is applied can be very different from the training objects. As a result, we are the first to show generalization without retraining on the Linemod and Linemod-Occluded datasets. Our analysis of the failure modes of previous template-based approaches further confirms the benefits of local features for template matching. We outperform the state-of-the-art template matching methods on the Linemod, Linemod-Occluded and T-LESS datasets.

[102] **Van Nguyen Nguyen**, Thibault Groueix, Georgy Ponimatkin, Vincent Lepetit, Tomas Hodan. *CNOS: A Strong Baseline for CAD-based Novel Object Segmentation*. International Conference on Computer Vision Workshops (ICCVW), 2023.

Abstract: We propose a simple yet powerful method to segment novel objects in RGB images from their CAD models. Leveraging recent foundation models, Segment Anything and DINOv2, we generate segmentation proposals in the input image and match them against object templates that are pre-rendered using the CAD models. The matching is realized by comparing DINOv2 `cls` tokens of the proposed regions and the templates. The output of the method is a set of segmentation masks associated with per-object confidences defined by the matching scores. We experimentally demonstrate that the proposed method achieves state-of-the-art results in CAD-based novel object segmentation on the seven core datasets of the BOP challenge, surpassing the recent method of Chen *et al.* [17] by absolute 19.8% AP.

[103] **Van Nguyen Nguyen**, Thibault Groueix, Mathieu Salzmann, Vincent Lepetit. *GigaPose: Fast and Robust Novel Object Pose Estimation via One Correspondence*. Computer Vision and Pattern Recognition (CVPR), 2024.

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

[101] **Van Nguyen Nguyen**, Thibault Groueix, Georgy Ponimatkin, Yinlin Hu, Renaud Marlet, Mathieu Salzmann, Vincent Lepetit. *NOPE: Novel Object Pose Estimation from a Single Image*. Computer Vision and Pattern Recognition (CVPR), 2024.

Abstract: The practicality of 3D object pose estimation remains limited for many applications due to the need for prior knowledge of a 3D model and a training period for new objects. To address this limitation, we propose an approach that takes a single image of a new object as input and predicts the relative pose of this object in new images without prior knowledge of the object's 3D model and without requiring training time for new objects and categories. We achieve this by training a model to directly predict discriminative embeddings for viewpoints surrounding the object. This prediction is done using a simple U-Net architecture with attention and conditioned on the desired pose, which yields extremely fast inference. We compare our approach to state-of-the-art methods and show it outperforms them both in terms of accuracy and robustness.

All the software, pre-trained models and datasets developed during this thesis are

available at https://www.github.com/nv-nguyen under open-source licenses. Besides, other publications are not explicitly discussed in the thesis:

[100] **Van Nguyen Nguyen**, Yuming Du, Yang Xiao, Michaël Ramamonjisoa, Vincent Lepetit. *PIZZA: A Powerful Image-only Zero-Shot Zero-CAD Approach to 6DoF Tracking*. International Conference on 3D Vision (3DV), 2022 (Oral).

[1] Arslan Artykov, Antoine Guedon, Clementin Boittiaux, **Van Nguyen Nguyen**, Vincent Lepetit. *MAGIC-GS: Monocular Articulated Generic Object Reconstruction with Gaussian Splatting*. In submission, 2024

[2] Guillaume Astruc, Nicolas Dufour, Ioannis Siglidis, Constantin Aronssohn, Nacim Bouia, Stephanie Fu, Romain Loiseau, **Van Nguyen Nguyen**, Charles Raude, Elliot Vincent, Lintao XU, Hongyu Zhou, Loic Landrieu. *Open Street View - 5M: The Many Roads to Global Visual Geolocation*. Computer Vision and Pattern Recognition (CVPR), 2024.

[51] Tomas Hodan, Martin Sundermeyer, Yann Labbe, **Van Nguyen Nguyen**, Gu Wang, Eric Brachmann, Bertram Drost, Vincent Lepetit, Castern Rother, Jiri Matas. *BOP Challenge 2023 on Detection, Segmentation and Pose Estimation of Unseen Rigid Objects*. Computer Vision and Pattern Recognition Workshops (CVPRW), 2024.

Besides the publications mentioned above, I have actively contributed to the research community as the main co-organizer of the BOP Challenge 2024 [1], which captures the state of the art in 6D object pose estimation. I also actively maintain the evaluation methodology BOP Toolkit [2] and the BOP datasets on HuggingFace [3].

[1] https://bop.felk.cvut.cz/challenges/bop-challenge-2024/
[2] https://github.com/thodan/bop_toolkit
[3] https://huggingface.co/datasets/bop-benchmark/datasets

Chapter 2

Related Work

This chapter provides an overview of methods for the detection, segmentation, pose estimation, and novel view synthesis of rigid objects, including those applied to both seen and novel objects. Pose estimation of rigid objects from images is one of the oldest problems in computer vision, with a long history in the field, dating back to Roberts' work in 1963 [118], and the advent of Deep Learning marked a turning point in the history of 3D pose estimation from images [74].

We therefore divide our review into four main sections. First, we briefly discuss classical methods that do not rely on Deep Learning in Section 2.1. We then review recent learning-based methods on pose estimation and related tasks for seen objects in Section 2.2. Finally, we review methods applied for novel objects, including CAD-based methods in Section 2.3, and CAD-free methods, which relax the requirement of CAD models in Section 2.4.

2.1 Classical methods

This section discusses traditional, non-learning-based methods for object pose estimation. It is structured around the image representations used by these methods. These representations often depends on the types of test objects and scenes these methods consider.

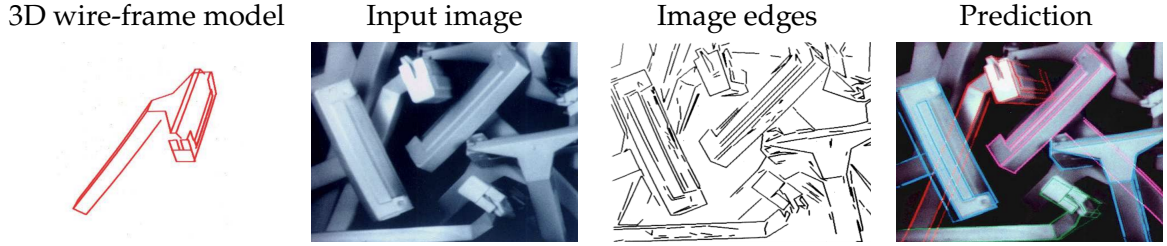


Figure 2.1: **In [86], David G. Lowe represents the input image as a set of edges**. These edges are matched with the contours of the 3D wireframe object model to detect the objects and recover their 6D poses. Figure from [86]. See 2.1.1

2.1.1 Edge-based features

In the early work [118], Roberts assumes that objects can be constructed as known simple 3D wireframe models, composed of primitives representing edges on the

objects. These primitives are detected in images by analyzing intensity gradients, for example, using Sobel filters, which are invariant over different viewpoints.

Given these detected primitives, we can define a set of matches between the input image and 3D wireframe model, then use a RANSAC-like strategy [86] to recover the initial object pose. Specially, for each match, an initial object pose is computed, we then measure the errors between the projection of the 3D model and the location of detected primitives. By searching over the set of possible matches, we find the optimal initial pose estimate. The pose is then further refined using Newton-based optimization with all matches. We illustrate these approaches with [86] in Figure 2.1. Similar edge-based methods [86, 4, 11, 60] were proposed to improve the performance.

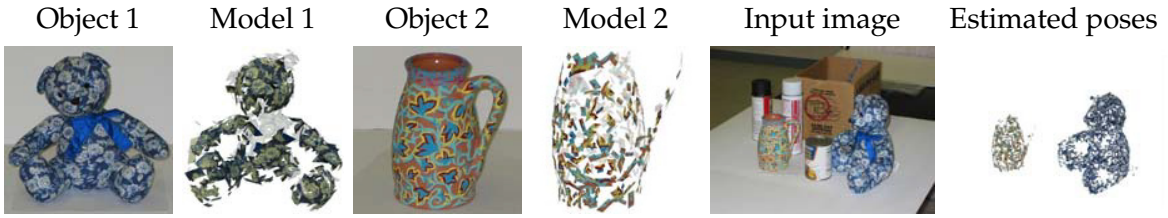


Figure 2.2: **Ponce *et al.* [111] represent an object using small, registered patches in 3D space.** This representation is automatically generated from a limited set of unordered training images. The 3D registration enables estimating the 6D object pose by matching model patches with those detected in the input image. Figure from [111]. See Section 2.1.2.

2.1.2 2D features

Edges are invariant to different viewpoints, they however discard the rich information in image brightness, color pattern of testing objects in the input image. 2D feature methods address this problem by modeling objects as sets of 3D patches associated with a local image descriptor as well as the position of the 3D patch within the object's coordinate system. These descriptors are invariant under geometric and photometric changes.

At test time, given a new image of the object, local invariant regions are detected in the image, are then matched against the 2D features registered previously. Finally, the 6D object poses are estimated from the established 2D-3D correspondences, typically using a PnP-RANSAC algorithm [36, 75]. This approach can be used with other types of local features, such as SIFT [87], MSER [92], and SURF [6]. We show in Figure 2.2 an example of this approach with [111].

2.1.3 3D features

If depth measurements are available, 2D features methods can be extended to 3D feature methods to detect and describe repeatable and distinctive local features on 3D surfaces. These method assume that the 3D object mesh models are available. First, these methods extract features from the 3D models and organize them into a database. Then, these methods match them against same features extracted from the input point cloud calculated from the depth image and camera intrinsics. Examples of 3D local features are SHOT [122], KPQ [94], ISS [162], and MeshDoG [135] while the

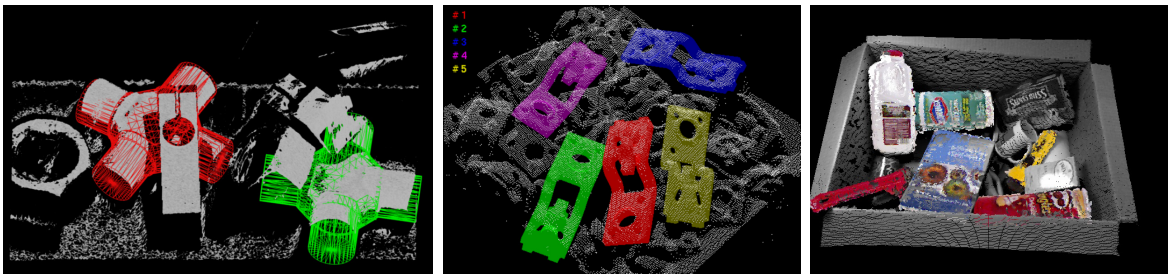


Figure 2.3: **Pose estimation using Point Pair Features (PPF) [31, 23, 22]**. The results show accurately align 3D models with the input point clouds calculated from depth image. Figure from [52]. See Section 2.1.3

most popular approach is Point Pair Features (PPF) [31]. PPF builds upon the concept of surflet pairs [139] by matching oriented point pairs between a test scene's point cloud and an object model [31] as shown in Figure 2.3. Several similar works [68, 29, 7, 47, 137, 23, 22] were introduced to improve the performance.

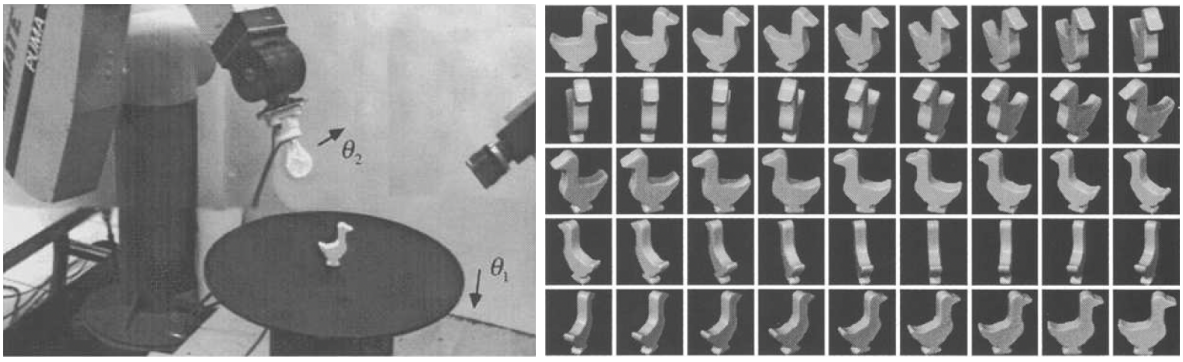


Figure 2.4: **Pose estimation using image templates [99].** An instrumented setup composed of a robot and a turntable (a) is used to capture image templates of an object under different viewpoints. Figure from [99]. See Section 2.1.4.

2.1.4 Template matching

Some methods compare prototype image representations, known as templates, against corresponding regions of the input image. The templates depict an object under various conditions, including different viewpoints and illumination settings, and are obtained by capturing the object in a controlled environment as shown in Figure 3.3. Additionally, the templates are automatically annotated with 6D object poses.

When a template is detected, its associated 6D pose can be used as the estimated pose, which can then be further refined. This approach requires a large number of templates and has been addressed in several works [99, 13, 66].

Template matching methods are still sensitive to a cluttered background and partial object occlusions as demonstrated in [99, 73]. We will further discuss and address these limitations later in this thesis.

2.2 Pose estimation for seen objects

With the advent of Deep Learning, most 6D object pose estimation and tracking methods have shifted towards relying on image features learned by neural networks [74]. In this section, we first review methods on instance detection, segmentation used in pose estimation (Section 2.2.1). We then discuss supervised pose estimation methods including template matching methods (Section 2.2.1), correspondence-based methods (Section 2.2.3), and direct pose estimation methods (Section 2.2.4).

2.2.1 Instance detection and segmentation

Performing object pose estimation typically involves two main steps: (1) detecting and segmenting target objects in the input image, and (2) estimating their 6D object poses from detected regions. While most methods focus on the second step, developing a method for the first step is also important as 2D detection/segmentation is generally an easier problem than 6D pose estimation, and it allows us to focus only on a specific region in the input image, significantly reducing the computation cost and run-time.

Since Mask-RCNN [44], used in CosyPose [71], showed promising performance in 2019 for instance segmentation of seven core-datasets of BOP challenge [56], most state-of-the-art pose estimation methods based on these two steps have been developed, and the problem of 2D detection/segmentation had been omitted by using Mask R-CNN [44] or GDRNPP [141]. To reduce computational costs, some popular methods focus only on 2D detection, such as FCOS [132], used in PFA [57] and YOLOX [39], used in GDRNPP [141]. Similar to pose estimation methods, while these supervised detection/segmentation methods have achieved impressive results, they rely heavily on large training datasets and require several hours or days to train for each target object.

We will further discuss and propose a novel method to address the problem of generalization later in the thesis.

2.2.2 Template matching

One of the first methods that exploits the power of Deep Learning for object pose estimation method is the template-matching approach proposed by Wolhart and Lepetit [148]. In their method, a convolutional neural network (CNN) is used to project the input image into a learned discriminative representation space. The CNN is trained using a triplet loss that separates two types of pairs: (1) descriptors of the same object in similar poses to be close in the learned space, and (2) different objects or different poses to be far away. During training, synthetic images are rendered from CAD object models at various viewpoints created by vertices on an isosphere. At inference time, retrieving the closest synthetic templates to the real input image in the learned representation space enables object pose estimation as shown in Figure 2.5.

In this context, Balntas *et al*. [5] showed how to obtain more discriminative representations by combining triplet loss with a regression loss. Similarly, Sundermeyer *et al*. [128] trained an auto-encoder to reconstruct object images augmented with domain randomization to learn representation space invariant to lighting, background.

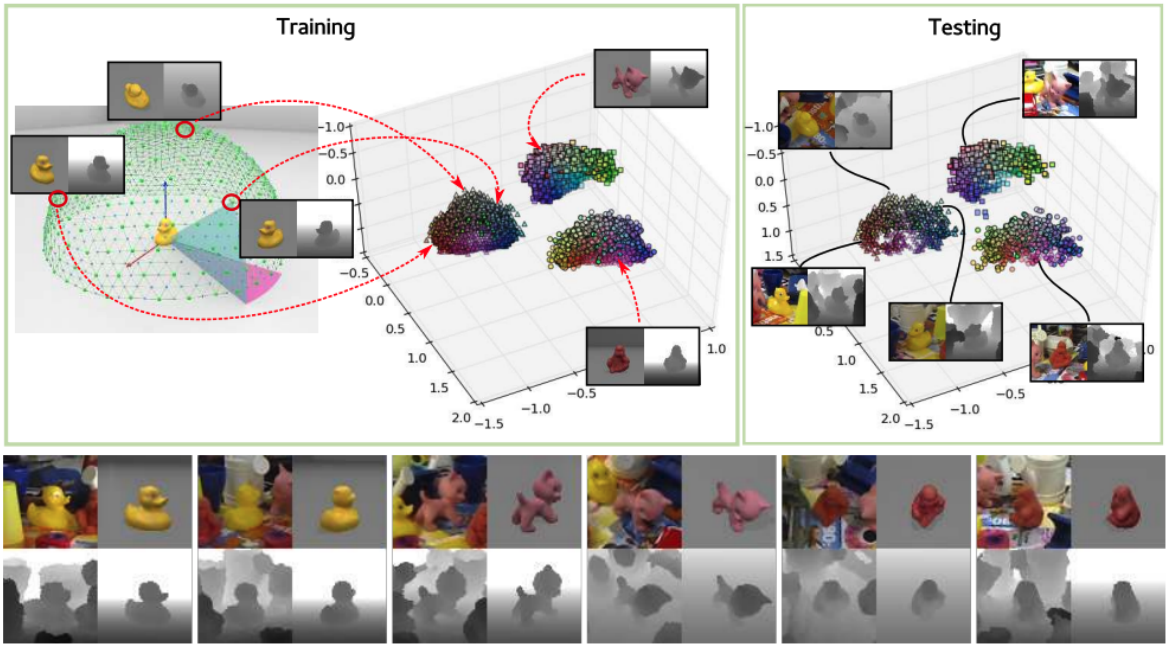


Figure 2.5: **Template-matching object pose estimation proposed by Wolhart and Lepetit [148].** During the training, images of different objects are mapped to well-separated descriptors while images of the same object are mapped to descriptors that capture the geometry of the corresponding poses (top left). New images are then mapped to locations corresponding to the object and 3D pose, even in the presence of clutter background (top right). At testing time, the method returns the correct template correspond to input RGB-D images via nearest neighbor search in the descriptor space (bottom). Figure from [148]. See Section 2.2.2

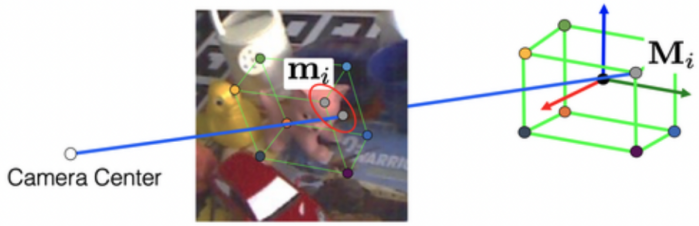


Figure 2.6: **Correspondence-based pose estimation BB8 [115]** where we first predict the 2D reprojections of some 3D points and then use a PnP algorithm on these 2D-3D correspondences to recover the object pose. Figure from [74]. See Section 2.2.3

2.2.3 Correspondence-based object pose estimation

A classical approach to solving the 6D object pose estimation problem is to establish 3D-to-2D or 3D-to-3D correspondences and then compute the pose with a PnP-RANSAC algorithm. This approach is popular in object pose estimation thanks to its robustness against outliers. One of the first learning-based correspondence-based methods is BB8 [115], which uses deep networks to regress local 2D-3D correspondences. Specifically, they first learn to segment objects in the image with a two-level coarse-to-fine object segmentation, then apply a deep network within an image window centered on the detected object to predict the 2D reprojections of the corners of

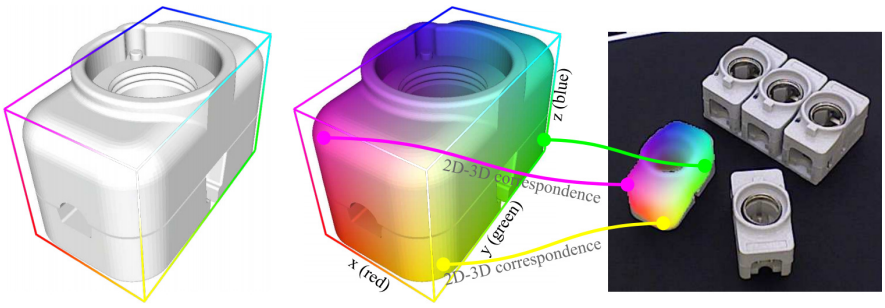


Figure 2.7: **Dense correspondences in Pix2Pose [108]**. The 3D points on the model surface (left) are mapped to colored coordinates (middle). The 3D object coordinates are predicted for densely sampled pixels of the input image (right) to establish 2D-3D correspondences. The 6D object poses are then estimated from the correspondences by using RANSAC. The image is taken from [108]. See Section 2.2.3

the 3D object's bounding box. Figure 2.6 shows the 3D object's bounding box corners (M_i) and their 2D reprojections in the image (m_i) used in BB8.

Instead of predicting sparse 2D-3D correspondences, some recent methods propose to predict dense correspondences using an encoder-decoder network such as DPOD [157] or Pix2Pose [108]. These methods predict the 3D coordinates of each pixel in a target image relative to the object's coordinate system. Specifically, DPOD represents 3D object coordinates via a UV map, which is a 2-channel image with values ranging from 0 to 255. Similarly, Pix2Pose regresses not only 3D object coordinates but also a confidence value based on prediction error, allowing for efficient filtering of matches at inference time and speeding up pose estimation. We show an example of dense correspondences used in Pix2Pose [108] in Figure 2.7.

Other relevant works in correspondence-based methods is PVNet [109], Single-stage [58], and EPOS [53]. PVNet regresses pixel-wise unit vectors pointing to the 2D reprojections of 3D control points and uses these vectors to vote for the final reprojections using RANSAC as shown in Figure 2.8. Single-stage proposes a deep architecture that takes as input clusters of 2D locations and regresses the 3D pose. EPOS represents objects by compact surface fragments to predict 3D location of the pixel on the predicted fragment.

2.2.4 Direct pose estimation

This category directly estimates the 3D translation and 3D rotation of the object given an image using learned features from deep neural networks. We distinguish between two approaches within this category.

The first approach directly estimates the absolute pose from an input image. For instance, in Figure 2.9, we show SSD6D [65] which extends single-shot object detector SSD [81] to predict the 6D pose for each detection.

In contrast, the second approach estimates relative poses, commonly starts with an initial pose estimate and iteratively refines the pose using a render-and-compare strategy. A relevant example of this second approach is BB8 [115] as shown in Figure 2.10. In BB8, a deep network is trained to improve the prediction of the 2D projections by comparing the input image and a rendering of the object for the initial

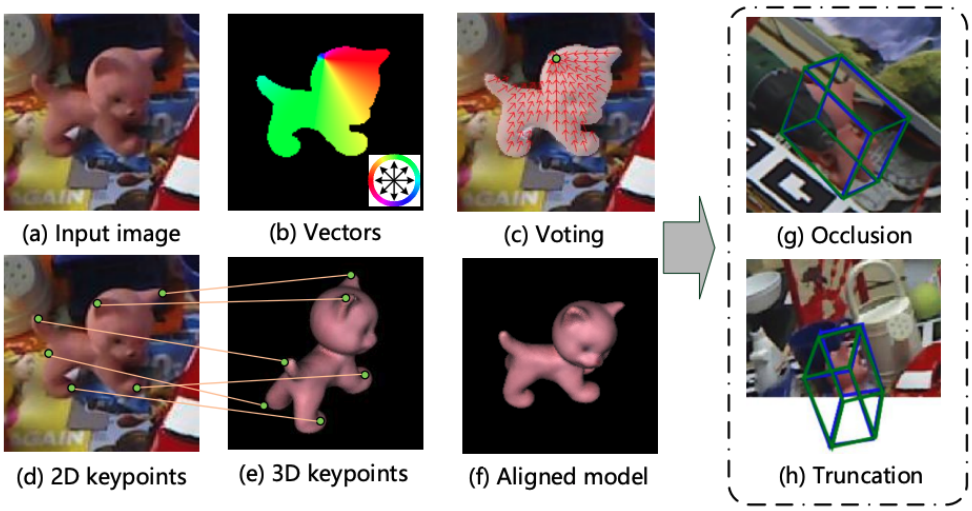


Figure 2.8: **Dense 2D-to-3D correspondences and voting network used in PVNet [109].** PVNet predicts unit vectors pointing to keypoints for each pixel, as shown in (b), and localize 2D keypoints in a RANSAC-based voting scheme, as shown in (c). The proposed method is robust to occlusion (g) and truncation (h), where the green bounding boxes represent the ground truth poses and the blue bounding boxes represent the predictions. The image is taken from [109]. See Section 2.2.3

pose estimate. Similarly, DeepIM introduces a novel parametrization of the pose update that does not use intermediate keypoints and can be trained end-to-end. CosyPose [71] builds upon this parametrization of DeepIM, incorporating a novel disentangled loss from CDPN [78], rotation6D [166], and strong data augmentation that leads to more stable training and improved performance, achieving state-of-the-art results for the BOP challenge 2020 [56].

Overall, direct pose estimation methods have an advantage in relying solely on learning-based components, allowing them to improve their performance by scaling up the training datasets.

2.3 CAD-based pose estimation for novel objects

This section provides an overview of learning-based methods that can be applied to novel objects without re-training, and only rely on the CAD model of test objects. We first discuss object detection, segmentation methods applied for novel objects (Section 2.3.1). We then discuss each group of pose estimation methods, including template matching methods (Section 2.3.2), correspondence-based methods (Section 2.3.3), and direct pose estimation methods (Section 2.3.4).

2.3.1 Instance detection and segmentation

Object segmentation methods traditionally focus on scenarios known as “closed-world" settings, where the training and test sets share the same object classes. Nevertheless, recent observations by Du *et al.* [34, 33] suggest that class-agnostic instance segmentation networks can effectively generalize to previously unseen object classes.

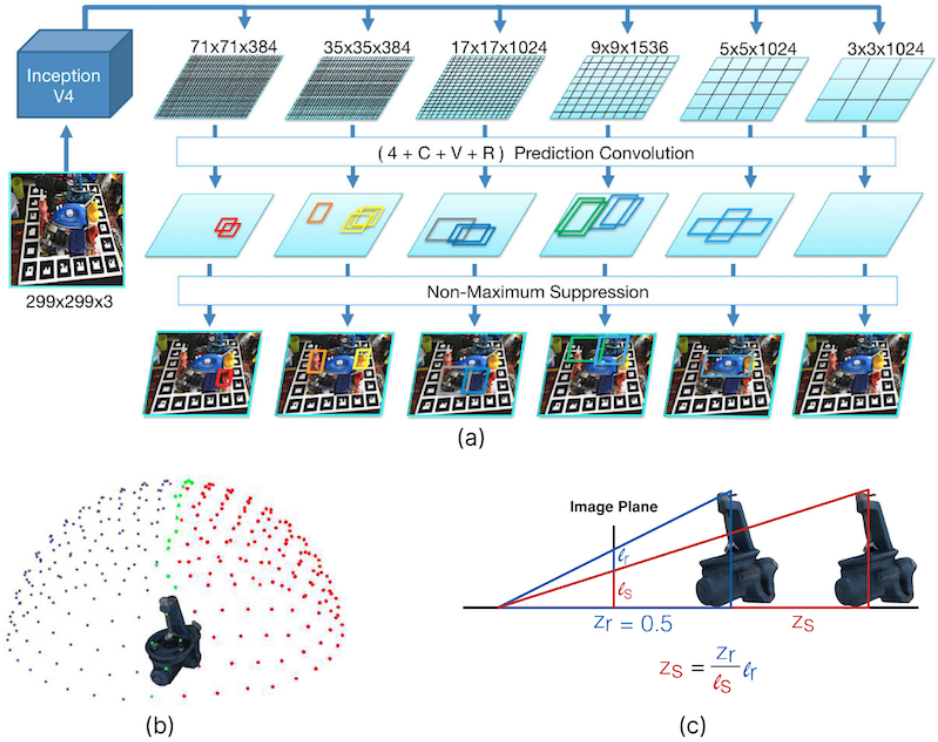


Figure 2.9: **Direct pose estimation in SSD6D [65]** A CNN is used to directly detect the position, size, and orientation of objects in an image. The figure (b) illustrates the discretization of the space of possible object orientations. The simple geometric reasoning depicted in the figure (c) allows for the recovery of the z-axis translation of the object pose given the size of its 2D bounding box. Image taken from [70]. See Section 2.2.4

Building upon this insight, Zhao *et al.* [161] leverage saliency detection models to solve the novel class discovery task in 2D segmentation.

Commonly used in robotics, UOIS-Net [155] uses a two-stage approach to segment novel objects. It operates on the depth channel of captured RGB-D images to generate object instance center votes and assembles them into rough initial masks. These masks are subsequently refined using the RGB channels. Xiang *et al.* [151] also propose an RGB-D based method that uses learned feature embeddings and applies a mean shift clustering algorithm to discover and segment unseen objects.

To avoid using depths, Durner et al. [35] use horizontal correlation to extract disparity RGB-based features and segment novel objects from stereo RGB images. It is worth noting that UOIS-Net [155], Xiang et al. [151], and Durner et al. [35] are RGB-D or stereo RGB approaches, while in this thesis, we focus the detection, segmentation of unseen objects from only CAD models or a single reference RGB image, which is more applicable.

Recently, Shugurov *et al.* [123] introduced OSOP that can be used to segment novel objects from the CAD model and show the promising results on BOP datasets [51]. However, in their method, detector modules is built upon features of templates, rendered from CAD model of each object, and requires a feed-forward for each testing object considered. Therefore, its complexity is linear to the number of testing objects.

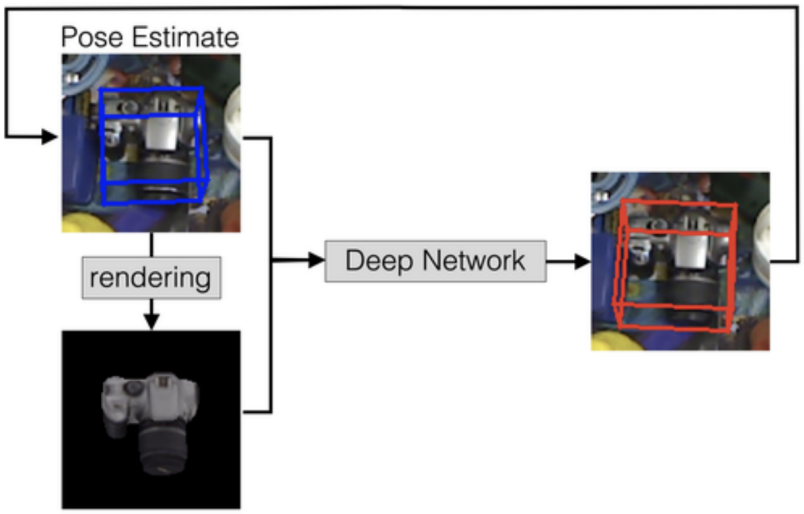


Figure 2.10: **Direct pose estimation via iterative render-and-compare in BB8 [115].** Given a first pose estimate, shown by the blue bounding box, BB8 generates a color rendering of the object. A network is trained to predict an update for the object pose given the input image and this rendering, to get a better estimate shown by the red bounding box. The same strategy is applied multiple times iteratively. The image is taken from [74]. See Section 2.2.4

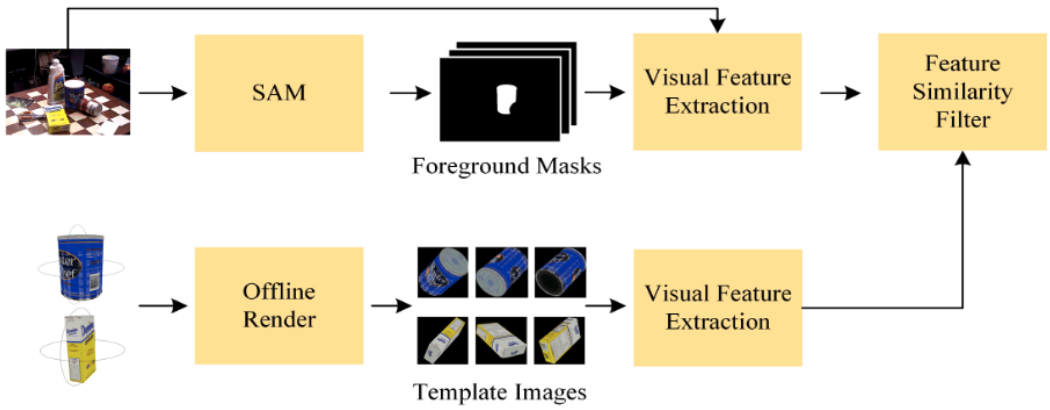


Figure 2.11: **CAD-based novel object segmentation in [17].** Their method first uses offline rendering to generate reference images and Segment Anything [69] to generate proposal mask candidates, which are then classified using visual cues from ImageBlind [40]. Image reproduced from [17]. See Section 2.3.1

More recently, Segment Anything (SAM) [69] has introduced a powerful foundation model for image segmentation capable of segmenting all objects in a given RGB image. Chen *et al.* [17] use SAM to extract object proposals, which are then combined with visual clues extracted by ImageBlind [40], enables novel object segmentation from only CAD models, as shown in Figure 2.11. While their method shows promising results, there is still significant gap comparing to supervised methods such as Mask R-CNN [44] used in CosyPose [71].

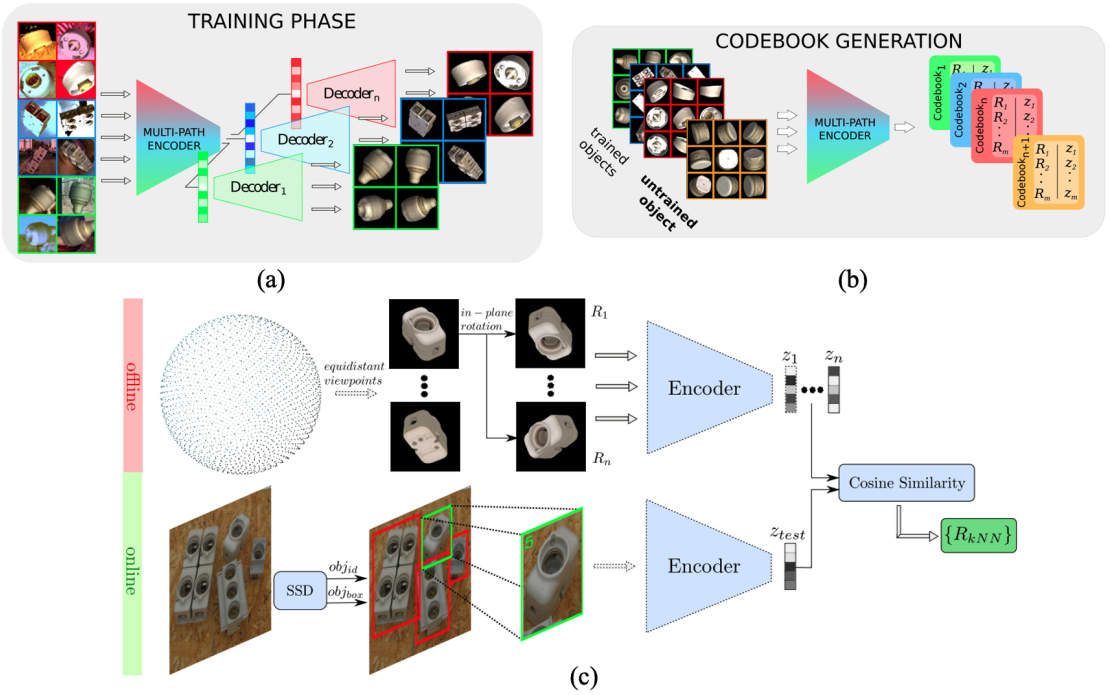


Figure 2.12: **Template matching pose estimation for novel objects in MultiPath [127].** (a) During training one encoder is shared among all objects, while each decoder reconstruct views of a single object. (b) This turns the encoder into a viewpoint-sensitive feature extractor, that generates expressive encodings for multiple trained and even untrained objects. (c) Template-matching pose estimation during the inference. Top: creating a codebook from the encodings of discrete synthetic object views; bottom: object detection and 3D orientation estimation using the nearest neighbor(s) with highest cosine imilarity from the codebook. Image reproduced from [127]. See Section 2.3.2

2.3.2 Template matching

As discussed in previous section, Wolhart and Lepetit [148] proposed to learn discriminative representations of templates, which are images of objects associated with the corresponding 3D poses. Pose estimation could then be achieved by matching the input image against these templates in an image-retrieval manner.

Building upon this concept, Balntas *et al*. [5] then showed how to obtain more discriminative representations to generalize on novel objects. While the ability to consider novel objects from their 3D models is the motivation for this thesis, it was only superficially demonstrated.

MultiPath [127] proposed an extension of Implicit [128] with a novel architecture with multiple decoders to adapt to different object types that can generalize to novel objects as shown in Figure 2.12.

While their results indeed show this generalization, the testing objects must remain similar to the training ones. As a consequence, this method has been demonstrated only on the T-LESS dataset [54], which depicts different kinds of electrical appliances that bear strong visual similarities.

Recently, MegaPose [72] has proposed a two-stage approach for estimating 6D pose of novel objects, as depicted in Figure 2.13. In the first stage, MegaPose uses

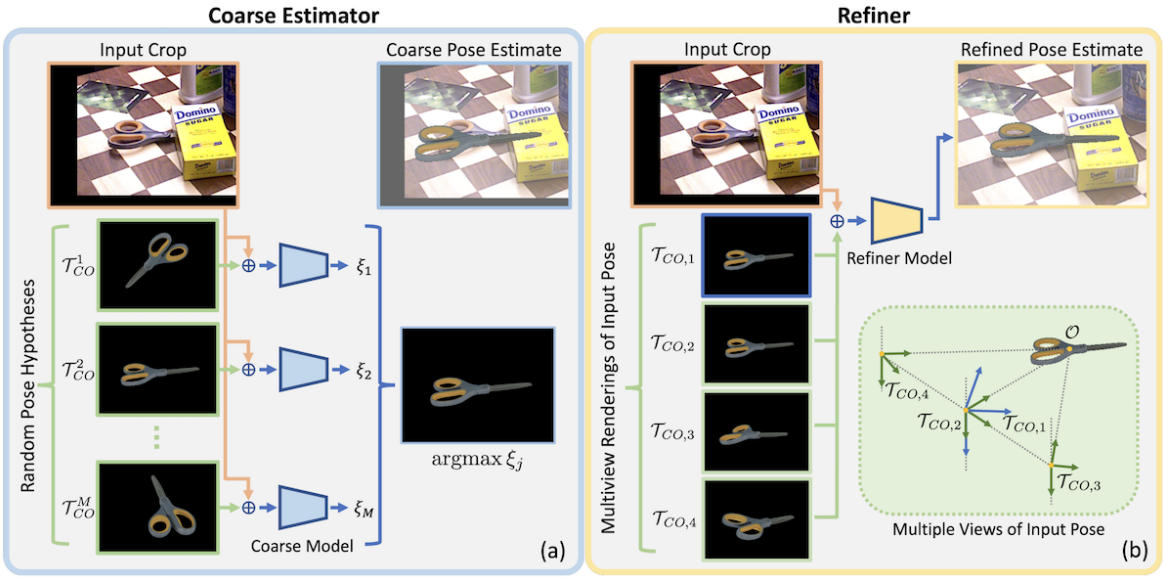


Figure 2.13: **Two-stage render-and-compare pose estimation for novel objects in [72]**: (a) Left: Given a cropped input image, the coarse module renders object templates in multiple poses, then classifies which template best matches the observed image, (b) Right: Given an initial pose estimate from coarse network, the refiner renders the objects at the estimated pose (blue axes) along with 3 additional viewpoints (green axes) to predict an updated pose estimate. Figure from [72]. See Section 2.3.2

templates in a render-and-compare manner to find a coarse pose. Specifically, unlike previous methods that use nearest neighbors search [148, 5, 127], their method uses a network that takes a test image-template pair as input and outputs a similarity score. This feed-forwarding is applied for all templates, and the template with the highest estimated scores is considered as the prediction. The second stage involves another deep network that iteratively refines the prediction from the first stage, similar to DeepIM [77]. Since MegaPose requires a feed-forward for each pair input image-template, which makes its runtime time-consuming, requiring more than 1.6 seconds per input detection on a single GPU V100.

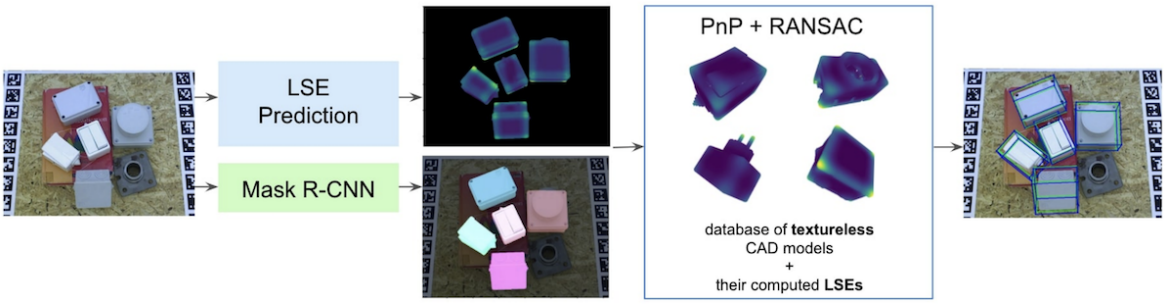


Figure 2.14: **Correspondence-based pose estimation for novel objects in [110]**. Given an input RGB image, their method first predicts Local Surface Embeddings (LSEs) for each pixel that we match with the LSEs of 3D points on the CAD models, then uses a PnP algorithm and RANSAC to estimate the 3D poses from these correspondences. Image taken from [110]. See Section 2.3.3

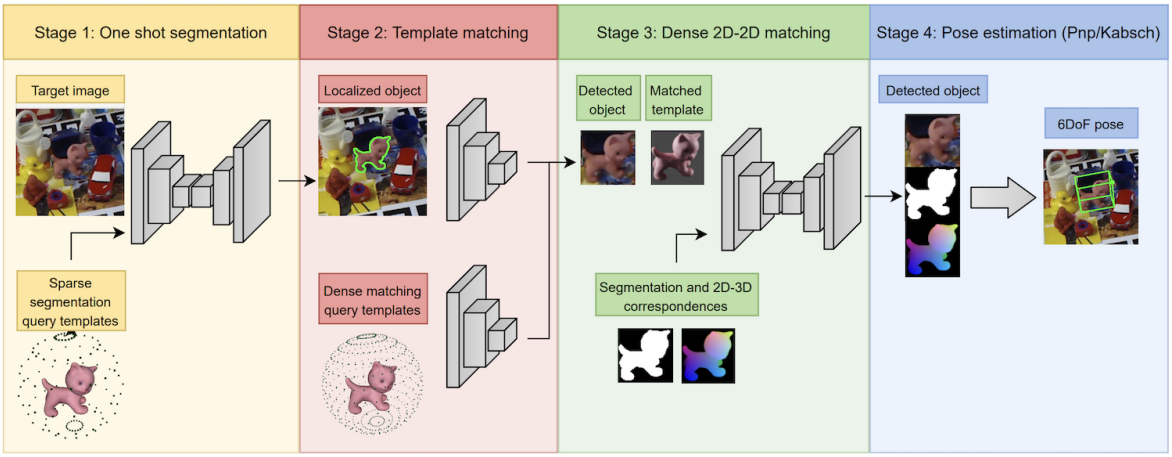


Figure 2.15: **Dense correspondence-based method in OSOP [123]**. Their method is divided into four main stages: 1) One-shot object localization conditioned on the 3D model. 2) Initial viewpoint estimation by template matching. 3) Dense 2D-2D matching between the image patch and the matched template. 4) 6 DoF pose estimation with PnP+RANSAC or Kabsch+RANSAC. Image taken from [123]. See Section 2.3.3

2.3.3 Correspondence-based object pose estimation

Pitteri *et al*. [110] introduce Local Surface Embeddings (LSEs), a method for estimating the 6D pose of new objects from their CAD models. LSEs are calculated from CAD models, capturing local geometry at each 3D point on the object surface while being translation-invariant and rotation-invariant. Given the 3D model of target object, they first calculate a set of LSE. Their method then uses a U-Net to calculate similar embeddings from input RGB images. The U-Net is trained on a small set of training objects and generalizes on novel objects. Given two sets of LSEs, their method can then match CAD models to input images and establish 2D-3D correspondences. Finally, RANSAC is used to estimate the 6D object pose as shown in Figure 2.14.

Since this matching process is performed independently for each 3D point, it has many ambiguities and a combinatorial matching cost, resulting in frequent failures. Another limitation of LSEs is that it is only efficient for objects with prominent corners. Consequently, this method has been demonstrated primarily on the T-LESS dataset.

Two other relevant correspondence-based approaches for novel objects are OSOP [123] and ZS6D [3]. Both methods initially render multiple templates from CAD models, predict 2D-to-2D correspondences, then convert them into 2D-to-3D correspondences as the pose of templates is known, and finally use RANSAC-PnP [75] to recover the 6D object pose. The main difference between these two approaches is on how they predict the correspondences: OSOP use a deep network to predict dense 2D-to-2D correspondences while ZS6D uses self-supervised vision transformers [15] to find a set of sparse 2D-to-2D correspondences via nearest neighbor search. We show in Figure 2.15 the overview of OSOP [123].

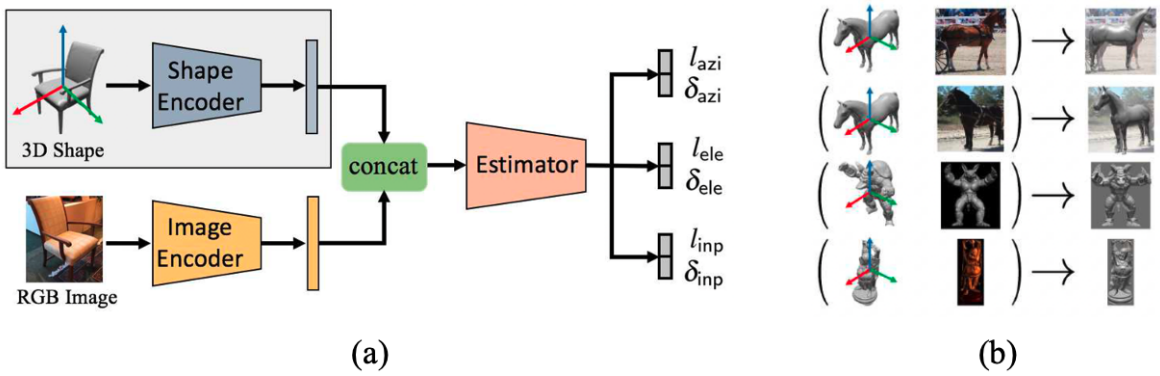


Figure 2.16: **Direct pose estimation in PosefromShape [154].** (a) A network is used to predict the orientation of a novel object in an RGB image on diverse training data including many categories. (b) The trained model can be tested on novel objects such as horses or figurines. Image reproduced from [154]. See Section 2.3.4

2.3.4 Direct pose estimation

Learning-based direct pose estimation methods can be scaled up to handle novel objects by using large-scale training data. Built upon this simple idea, PosefromShape [154] proposed a method to regress the 3D rotation of an object, using a shape encoder to encode a novel object's appearance and coordinate system as shown in Figure 2.16.

Similarly, DeepIM [77] has demonstrated that its method can generalize to unseen categories when trained on a wide range of categories in ModelNet. While this experiment of DeepIM was conducted only on synthetic datasets, shows that pose estimation models can effectively generalize to unknown categories by learning diversity from various categories during training.

The refined network in MegaPose [72] has demonstrated a similar observation: by training a deep network on large-scale datasets, they showed that their method can refine accurately 6D pose estimates of novel objects.

2.4 CAD-free pose estimation for novel objects

Previous sections have discussed CAD-based generalizable methods, which are relevant for warehouse or factory settings where CAD models of target objects are often available. However, this setting still has limitations in real-world scenarios since it requires expensive devices to scan 3D models of test objects.

To overcome this issue, we discuss in this section methods that do not require a 3D model. First, we review category-level methods in Section 2.4.1. We then discuss in Section 2.4.2 methods for 6D object tracking that do not require CAD models but rely on temporal constraints. Finally, we review methods based on multi-view images in Section 2.4.3.

Figure 2.17: **Category-level pose estimation with normalized object coordinates in [143]**. The normalized object coordinate space, which is contained within a unit cube, displays a 3D space for each category. Within this shared space, all instances of a given category, such as cameras, are aligned in terms of orientation, while their sizes are normalized to fit within the unit cube. This allows for direct prediction of 2D-3D correspondences for novel objects that belong to a known category. As shown in (b), example categories include bowls, cameras, cans, laptops, and mugs. Image taken from [70]. See Section 2.4.1.

2.4.1 Category-level methods

One way to avoid retraining on new object instances and CAD-model requirement is to consider object categories, and train a model on target categories that will generalize to new instances of these categories [165, 143, 21, 76, 89]. We show in Figure 2.17 a relevant example from NOCS [143].

While such an approach can be useful in some applications, such as scene understanding, in many others, the new objects do not belong to a known category. By contrast, this thesis focuses on developing methods that can generalize to new objects with no similarity in shape to the objects used to train the initial model.

2.4.2 Temporal 6D pose tracking

Instead of relying on a single image for absolute pose estimation, tracking methods exploit temporal information. This problem has been formulated under a deep learning framework, where a network is trained to regress the pose difference between image pairs extracted from RGB [25, 12] or RGB-D videos [37, 38, 164, 140, 145]. Within the context of generalize object pose tracking, [12, 145, 38] has shown that their methods can be applied to new objects. We show an example with BundleTrack in Figure 2.18. However, their method still require 3D information in the form of a CAD model [12] or depth images [145, 38].

2.4.3 Multi-view image-based methods

A few recent methods such as LatentFusion [107], Gen6D [83], OnePose [126] address the problem of not having the 3D models for the target objects by first acquiring and registering reference RGB images taken from multiple viewpoints. LatentFusion [107]

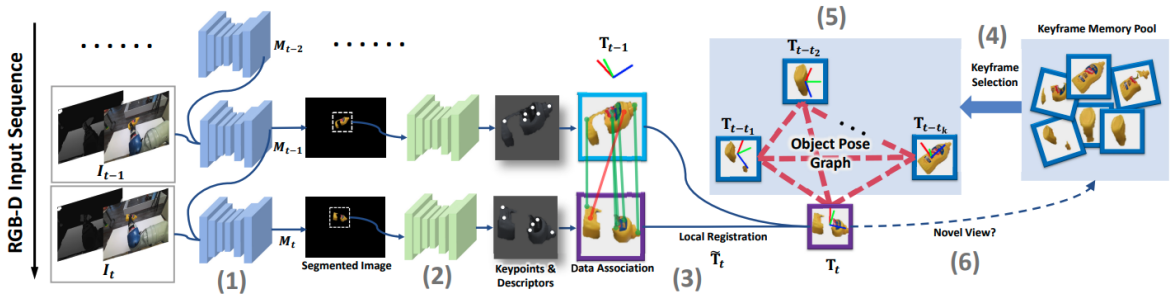


Figure 2.18: **6D pose tracking for novel objects in BundleTrack [145].** Their method is composed of six stages: 1) an image network is used to segment object masks; (2) another network detects keypoints and corresponding descriptors; (3) keypoints are matched and coarse registration is used on consecutive frames to estimate an initial relative poses; (4) keyframes are selected and form a pose graph; (5) online pose graph optimization outputs a refined, spatiotemporally consistent pose; and (6) the latest frame is included in the memory pool if it represents a novel view, enriching diversity. Image taken from [145]. See Section 2.4.2.

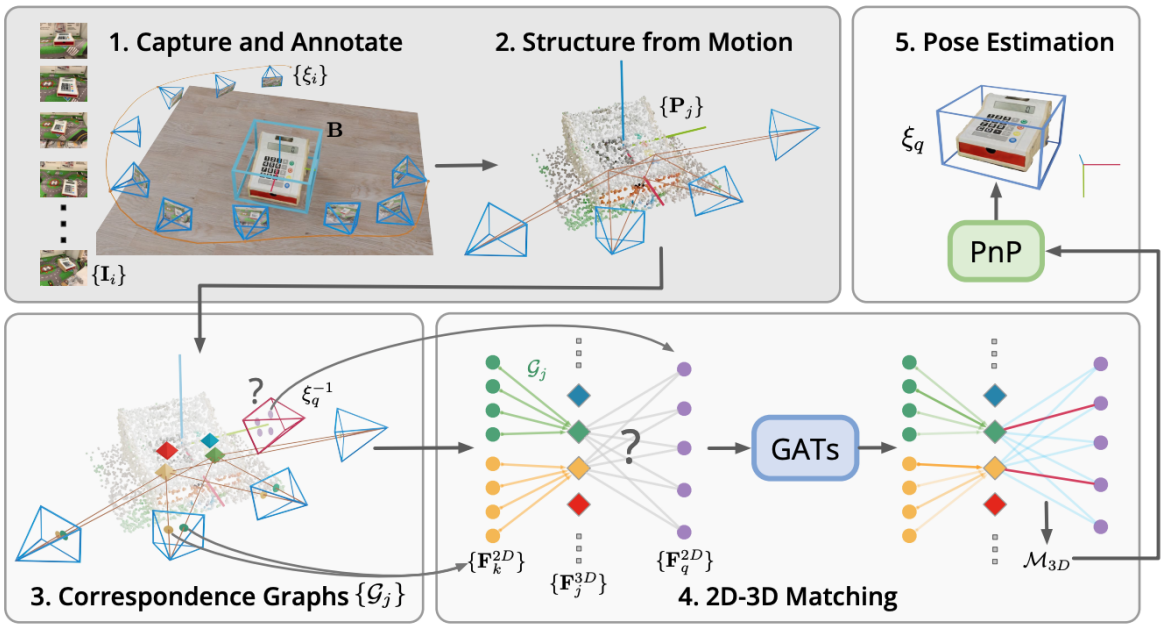


Figure 2.19: **CAD-free novel object pose estimation in OnePose [126].** Their method is composed of five stages for estimating 6D pose of novel objects: 1) For each object, a video scan with RGB frames and annotated camera poses; (2) a Structure from Motion (SfM) reconstructs a sparse point cloud; (3) The correspondence graphs are built during SfM, which represent the 2D-3D correspondences in the SfM map; (4) 2D descriptors are aggregated to 3D descriptors to generate 2D-3D matches; (5) Finally, the object pose is computed by solving the PnP problem. The image is taken from [126]. See 2.4.3

also requires an additional training phase to obtain a 3D latent representation, while OnePose [126] necessitates optimization with Structure-from-Motion to establish cor-

respondence graphs as shown in Figure 2.19. While these methods show promising results, it is noteworthy that capturing the reference images still demands human expertise and some time for capture and registration.

Chapter 3

CNOS: A Strong Baseline for CAD-based Novel Object Segmentation

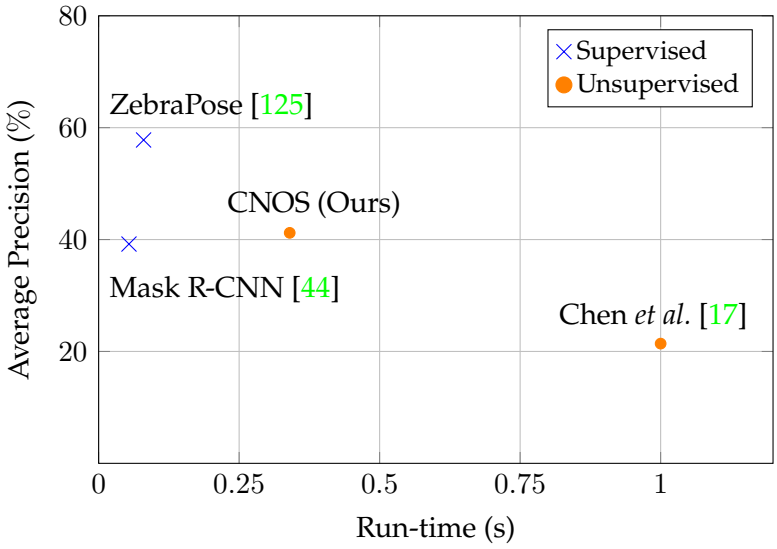


Figure 3.1: **Performance on the seven core datasets of BOP challenge [51].** Our method, CNOS, relies on FastSAM [160] for generation of segmentation proposals and on DINOv2 [106] for visual description. CNOS outperforms the unsupervised method of Chen *et al.* [17] and even the supervised method Mask R-CNN [44], which was trained on tens of thousands of images per BOP dataset and used in CosyPose [71]. Similarly to Mask R-CNN [44], the runtime of CNOS is dominated by the proposal stage.

The work presented in this chapter was initially presented in:

[102] **Van Nguyen Nguyen**, Thibault Groueix, Georgy Ponimatkin, Vincent Lepetit, Tomas Hodan. *CNOS: A Strong Baseline for CAD-based Novel Object Segmentation*. International Conference on Computer Vision Workshops (ICCVW), 2023.

As discussed in Section 2, performing object pose estimation typically involves two main steps: (1) the target objects are detected/segmented in the input image, and (2) the 6D object poses are then estimated from the detected regions [51]. Recent works such as MegaPose [72] introduced effective CAD-based object pose estimation methods. However, these methods mainly focus on the second step and require input 2D bounding boxes, which restricts their applicability to scenarios where precise 2D bounding boxes are available.

To address the gap, in this chapter, we propose a simple method for object detection and segmentation that only requires CAD models of the target objects. The method is dubbed **CNOS** for **C**AD-based **N**ovel **O**bject **S**egmentation.

In CNOS, new objects are onboarded by rendering their CAD models and describing each rendered template by the DINOv2 `cls` token [106]. Given an RGB input image, segmentation proposals are extracted from the image by Segment Anything (SAM) [69] or Fast Segment Anything (FastSAM) [160] and matched against the templates based on the similarity between their DINOv2 `cls` tokens. Rendering the templates takes less than 2 seconds per CAD model which is much faster than retraining of supervised methods, which typically requires several hours. The choice of DINOv2 for measuring the similarity between templates and proposals is mainly motivated by its ability to effectively address the domain gap between real and synthetic images. We also demonstrate that photo-realistic rendering techniques of BlenderProc [26], which require approximately 1 second to render an image, can be leveraged to further mitigate this domain gap and enhance accuracy. Experiments on the seven core datasets of the BOP challenge [51] demonstrate the state-of-the-art performance of CNOS.

As shown in Figure 3.1, CNOS outperforms the recent unsupervised method for CAD-based segmentation by Chen *et al.* [17] and even Mask R-CNN [44], a supervised method that was trained on tens of thousands of images per BOP dataset and used in CosyPose [71].

3.1 Method

In this section, we provide a detailed description of our three-stage approach for CAD-based novel object segmentation. We first describe the onboarding stage in Section 3.1.1, where we extract visual descriptors from renderings of the CAD models. In Section 3.1.2, we explain the proposal stage, which involves obtaining all possible masks and their descriptors. Finally, in Section 3.1.3, we discuss the matching stage, where object masks are retrieved and labeled based on visual descriptors of their CAD models.

3.1.1 Onboarding stage

In the onboarding stage, we render a set of RGB synthetic templates and extract their visual descriptors using DINOv2 [106]. To ensure robust object segmentation under different orientations, we render CAD models under 42 viewpoints as shown in Figure 3.3. These 42 viewpoints are defined by the icosphere primitive of Blender [1] which has been shown in [104] to provide well-distributed view coverage of CAD

[1] `bpy.ops.mesh.primitive_ico_sphere_add()`

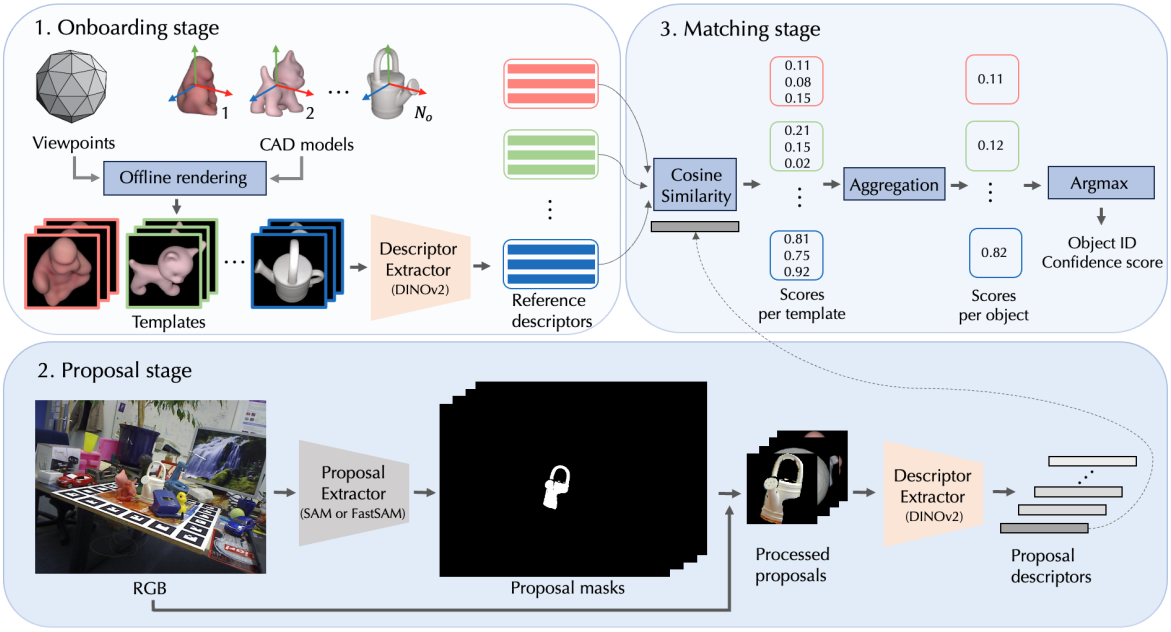


Figure 3.2: **CNOS overview.** Given CAD models of the target objects, the objects are onboarded by (i) rendering a set of templates showing the models from different viewpoints, and (ii) describing the templates by the DINOv2 `cls` token (Section 3.1.1). At inference time, segmentation proposals are generated from the input RGB image using SAM or FastSAM (Section 3.1.2), and the proposals are matched against the templates by comparing their DINOv2 `cls` tokens (Section 3.1.3).

models for robust template matching. Additionally, we experiment with denser viewpoints by dividing each triangle of the icosphere into four smaller triangles. The rendering process results in a total of $N_o \times N_v$ templates, where N_o is the number of CAD models and N_v is the number of viewpoints. We then crop the templates with the ground-truth bounding boxes and use the DINOv2 `cls` tokens as their visual descriptors $\mathbf{D_r}$ of size $N_o \times N_v \times C$. By default, we use $N_v = 42$ and $C = 1024$.

3.1.2 Proposal stage

For each testing RGB image, we use SAM [69] or FastSAM [160] with a default configuration to generate a set of N_p unlabeled proposals, where each proposal i is defined by a mask M_i. N_p is not fixed and varies depending on the input RGB image.

To compute the visual descriptor for each proposal i, we first remove the background of the input image using the corresponding mask M_i. Subsequently, we crop the image using the model bounding box derived from M_i. Since each proposal mask M_i has a different bounding box size, parallel processing becomes unfeasible. To overcome this, we add a simple image processing step including scaling and padding in order to resize all proposals to a consistent size of 224×224. This standardization enables efficient parallel processing of proposals in a single batch. Then, we extract the DINOv2 `cls` tokens from the processed proposals and use them as their visual descriptors $\mathbf{D_p}$ of size $N_p \times C$.

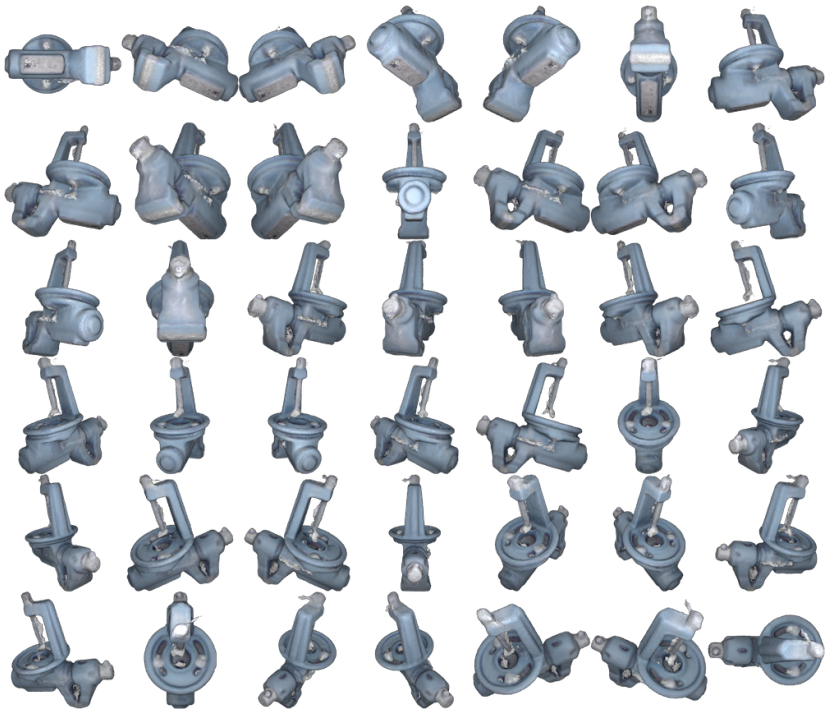


Figure 3.3: **Visualization of templates for the "benchwise" object from LM [46] rendered with Pyrender [113].** 42 templates were rendered from viewpoints defined by the icosphere [104].

3.1.3 Matching

The goal of the matching stage is to assign each proposal i an object ID o_i and a confidence score s_i. To this end, we compare each proposal descriptor in $\mathbf{D_p}$ with each template descriptor in $\mathbf{D_r}$ using the cosine similarity. This comparison step produces a similarity matrix of size $\mathrm{N_p} \times \mathrm{N_o} \times \mathrm{N_v}$.

View aggregation. By aggregating the similarity scores over all $\mathrm{N_v}$ templates for each CAD model, we obtain a matrix of size $\mathrm{N_p} \times \mathrm{N_o}$. This matrix represents the similarity between each proposal p_i and each CAD model. We experiment with different aggregation functions, such as Mean, Max, Median, and Mean of top k highest, noted Mean_k, and find that Mean_k yields the best results.

Object ID assignment. To assign the object ID o_i and confidence score s_i to each proposal, we simply apply the argmax and max functions on the similarity matrix $\mathrm{N_p} \times \mathrm{N_o}$ over the $\mathrm{N_o}$ objects. This yields a similarity matrix of size $\mathrm{N_p}$ defining the confidence score for $\mathrm{N_p}$ proposals.

Output. At the end of the matching stage, we obtain a set of labeled proposals, where each proposal is defined as $\{\mathrm{M}_i, \mathrm{o}_i, \mathrm{s}_i\}$, where M_i is the modal mask (*i.e.*, a mask covering the visible part of the object surface [51]), o_i is the object ID, and s_i is the confidence score. Some of these proposals may still be incorrectly labeled. To address this, it is possible to apply a threshold δ on the confidence score threshold. The figures in Section 3.2.2 show CNOS's segmentation results with $\delta = 0.5$.

3.2 Experiments

In this section, we describe the experimental setup (Section 3.2.1), compare our method with previous works [17, 44, 125] on the seven core datasets of the BOP challenge [51] (Section 3.2.2), and conduct an ablation study focused on the accuracy under different aggregating functions and different numbers of rendering viewpoints, and on the run-time (Section 3.2.3). Finally, we discuss the use of CNOS in a pipeline for 6D pose estimation of novel objects (Section 3.2.4) or in CAD-free novel object segmentation.

3.2.1 Experimental setup

Datasets. We evaluate our method on the test set of seven core datasets of the BOP challenge [51]: LineMod Occlusion (LM-O) [8], T-LESS [54], TUD-L [50], IC-BIN [27], ITODD [30], HomebrewedDB (HB) [64] and YCB-Video (YCB-V) [150]. In total, the datasets include 132 different objects shown in cluttered scenes with occlusions. The objects are of verious types: textured or untextured, symmetric or asymmetric, household or industrial.

Evaluation metric. We evaluate our method using the Average Precision (AP) metrics, following the COCO metric and the BOP challenge evaluation protocol [51]. The AP metric is calculated as the mean of AP values at different Intersection over Union (IoU) thresholds ranging from 0.50 to 0.95 with an increment of 0.05.

Baselines. We compare our method with Chen *et al.* [17], the most relevant work to ours. They use a three-stage CAD-based object segmentation approach, incorporating SAM [69] for image segmentation and ImageBlind [40] for visual descriptor extraction. Their use of 72 templates per CAD model resulted in the best performance according to their paper. Additionally, we compare our method with two relevant supervised methods from the BOP challenge [51]: Mask R-CNN [44], which was trained on real or synthetic training images specific to each dataset and used in CosyPose [71], and ZebraPose [125], which is currently the state-of-the-art for this task in the BOP challenge.

Implementation details. For the proposal stage, we use the default ViT-H SAM [69] or the default FastSAM [160], which has demonstrated promising results in terms of run-time efficiency. For extracting visual descriptors, we use the default ViT-H model of DINOv2 [106].

To further evaluate the performance of our method, we conducted a comparison using two sets of templates. The first set of templates was generated using Pyrender [113] from 42 pre-defined viewpoints. It is worthy to note that Pyrender computes the Direct Illumination and it is extremely fast, takes on average 0.026 second per image. The second set of templates comprised 42 realistic rendering templates selected from the available synthetic images of the PBR-BlenderProc4BOP training set provided in the BOP challenge. These realistic templates were specifically chosen to closely match the orientations of the 42 predefined viewpoints in the first set. Since the PBR-BlenderProc4BOP training images possibly have occlusions, we chose only images

			BOP Datasets							
	Method	Rendering	LM-O	T-LESS	TUD-L	IC-BIN	ITODD	HB	YCB-V	Mean
Supervised	1 MaskRCNN [44] (Synth)	-	37.5	51.7	30.6	31.6	12.2	47.1	42.9	36.2
	2 MaskRCNN [44] (Real)	-	37.5	54.4	48.9	31.6	12.2	47.1	42.9	39.2
	3 ZebraPose [125] (Synth)	-	50.6	62.9	51.4	37.9	36.1	64.6	62.2	52.2
	4 ZebraPose [125] (Real)	-	50.6	70.9	70.7	37.9	36.1	64.6	74.0	57.8
Unsupervised	5 Chen *et al.* [17]	-	17.6	9.6	24.1	18.7	6.3	31.4	41.9	21.4
	6 CNOS (SAM)	Pyrender[113]	33.3	38.3	35.8	27.2	14.8	45.9	57.6	36.1
	7 CNOS (SAM)	BlenderProc [26]	39.6	39.7	39.1	28.4	28.2	48.0	59.5	40.4
	8 CNOS (FastSAM)	BlenderProc [26]	39.7	37.4	48.0	27.0	25.4	51.1	59.9	41.2

Table 3.1: **Comparison of our method with [44, 125, 17] on the seven core datasets of the BOP challenge [51].** MaskRCNN and ZebraPose are retrained specifically on these objects with synthetic renderings of the CAD model (noted as "Synth") or real images of the object (noted as "Real"). We classify them as "supervised" in contrast with [17] and our method which we call "unsupervised" because it requires no retraining for novel objects. We report the AP metric (higher is better) using the protocol from [51]. We highlight in blue the best supervised method and in yellow the best unsupervised method. Our method not only significantly outperforms [17] under the same settings but also surpasses the supervised method MaskRCNN, highlighting its ability to generalize.

where the target objects are fully visible. The templates of target objects are finally obtained by making the background black using the ground-truth mask and cropping regions with the ground-truth bounding box.

In order to maintain a consistent run-time across all datasets, we resize the images while preserving their aspect ratio. Specifically, we ensure that the width of each input RGB testing image is fixed at 640 pixels. All our experiments were conducted on a single V100 GPU.

3.2.2 Comparison with the state of the art

In Table 3.1, we show that CNOS outperforms Chen *et al.* [17] by a significant margin of absolute 19.8% AP. Furthermore, despite not being trained on the testing objects of the BOP datasets, our method surpasses the performance of Mask R-CNN [44] used in CosyPose [71], which was specifically trained on these objects. This highlights the generalization capability of our method.

We qualitatively found that the generated segmentation proposals usually include ones that are very well aligned with the target object instances, and that most mistakes are due to erroneous DINOv2-based classification of the proposals. Improving the proposal classification would be crucial to close the gap between CNOS and supervised state-of-the-art approaches such as ZebraPose.

We show in Figure 3.4 qualitative results of our method on LM-O [8], HB [64] and YCB-V [150] datasets.

Figure 3.4: **Qualitative results on LM-O [8], HB [64] and YCB-V [150].** The first column shows the input CAD models. In cases where there are more than 16 models, we only show the first 16 to ensure better visibility. The second column show the input RGB image and the last column depicts the detections produced by our method CNOS with confidence scores greater than 0.5. Interestingly, in the last row, even though the segmentation proposals in CNOS (SAM) and CNOS (FastSAM) are very similar, their final labels differ for a few objects. This inconsistency arises from DINOv2-based classification of the proposals as discussed in Section 3.2.2

3.2.3 Ablation study

Segmentation model	Descriptor model	AP	Run-time (s)
FastSAM-s	ViT-s	32.1	0.18
FastSAM-s	ViT-l (default)	33.8	0.25
FastSAM-x (default)	ViT-s	38.0	0.27
FastSAM-x (default)	ViT-l (default)	39.7	0.33

Table 3.2: **Ablation study of different FastSAM segmentation models [160] and DINOv2 descriptor models [106] on LM-O.**

Model size vs. run-time. We present the results for FastSAM and DINOv2 using various base models in Table 3.2, highlighting the trade-off accuracy-runtime.

Rendering. Table 3.1 demonstrates the performance of our method using two types of rendering: Pyrender [113] in row 6 and BlenderProc [26] in row 7. The results indicate that incorporating realistic rendering significantly reduces the domain gap between synthetic and real images, yielding a 4.3% improvement in the AP metric.

Method	Viewpoint density	
	Coarse (42)	Dense (162)
CNOS (SAM)	**39.6**	39.5
CNOS (FastSAM)	**39.7**	39.7

Table 3.3: **Ablation study on the number of viewpoints on LMO dataset [8].** The denser viewpoints are created by subdividing each triangle of the icosahedron (used to create coarse viewpoints) into four smaller triangles.

Number of viewpoints. As shown in Table 3.3, using more viewpoints does not bring any improvement compared to the coarse viewpoints. This can be explained by the fact that the current set of 42 coarse viewpoints already provides sufficient coverage of the 3D objects.

Run-time. In Table 3.4, we present the average run-time of each stage in our method for a given CAD model. In the onboarding stage, the average rendering time for one image with Pyrender[113] is 0.026 second while with BlenderProc [26] is around 1 second per image on a single V100 GPU. It is important to note that the onboarding stage is performed once for each CAD model. In terms of run-time, the onboarding stage is clearly bottlenecked by the generation of templates, while the proposal stage is currently bottlenecked by the segmentation algorithm.

Method	Run-time (second)		
	Onboarding	Proposal	Matching
CNOS (SAM, Pyrender)	1.22	1.58	0.13
CNOS (FastSAM, Pyrender)	1.22	0.22	0.12
CNOS (FastSAM, PBR)	42.1	0.22	0.12

Table 3.4: **Run-time.** We report the runtime of each stage of our method on a single V100 GPU. The runtime of the onboarding stage includes both the rendering time and the visual descriptor extraction time for each CAD model.

3.2.4 Discussion

Our intention was originally to use the DINOv2 `cls` token not only to recognize the object but also to estimate its initial pose that could be refined in a subsequent step. However, as illustrated in Figure 3.5, this approach did not yield successful results, as the DINOv2 `cls` token seems to carry sufficient information about the object identity but not about the object pose.

3.3 Conclusion

In this chapter, we presented a simple yet powerful method for novel object segmentation solely based on their CAD models, without the need of any training. The

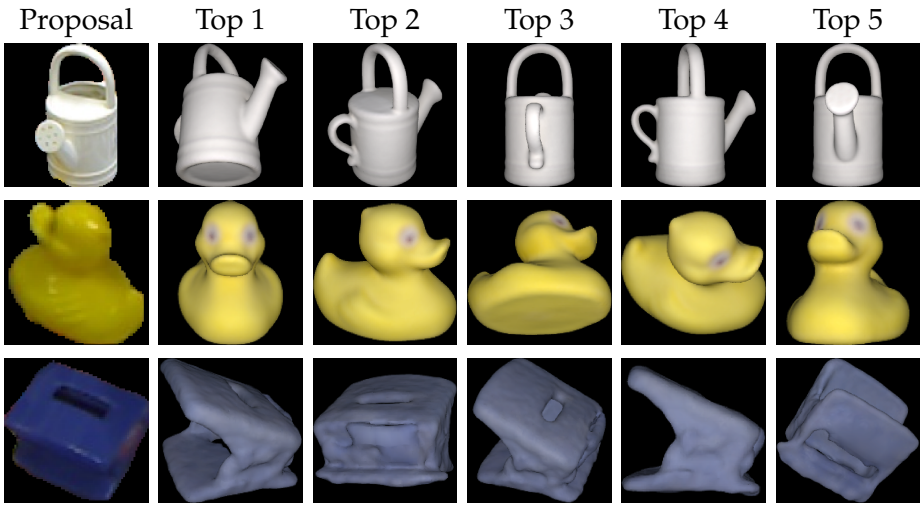


Figure 3.5: **Visualization of the nearest neighbors.** We show proposals along with five retrieved templates with the most similar DINOv2 `cls` tokens. The retrieved templates correspond to the same object but to poses that do not match the pose in the proposals – this suggests that the DINOv2 `cls` token can be effectively used to recognize the objects, but not to estimate the pose.

method achieves a surprisingly high accuracy, comparable to previous supervised methods trained on large-scale annotated datasets. We hope that CNOS will serve as a standard baseline for CAD-based novel object segmentation and will be employed as the initial stage of novel object pose estimation pipelines.

Chapter 4

Templates for 3D Object Pose Estimation Revisited: Generalization to New Objects and Robustness to Occlusions

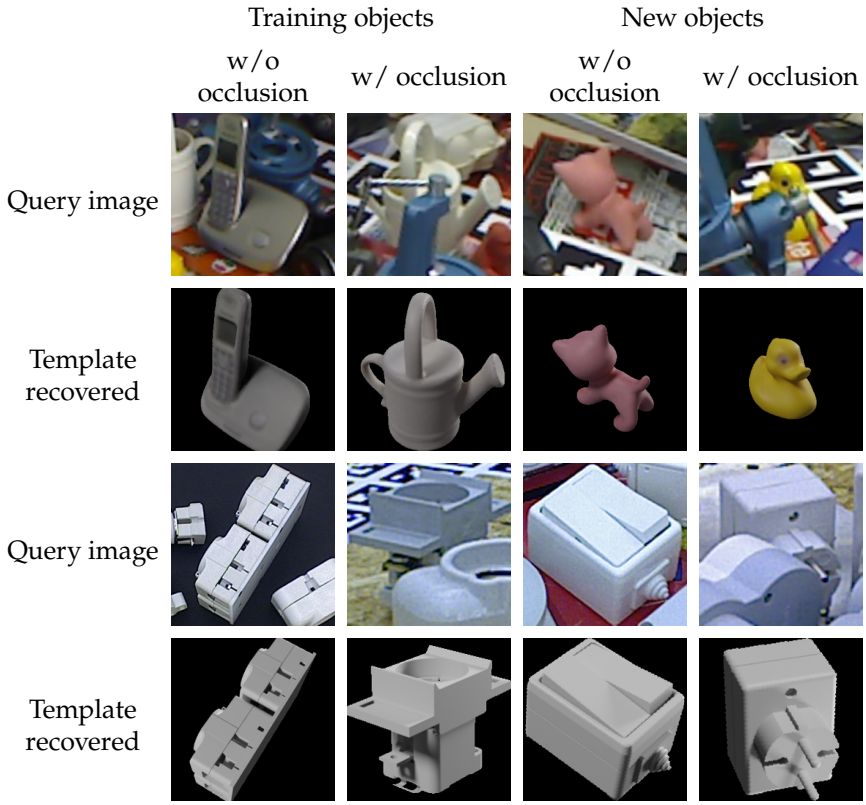


Figure 4.1: Our method can estimate the 3D pose of new objects in query images by matching them with templates created from their 3D models. **These new objects can be very different from the ones, and can be partially occluded in the query images.**

The work presented in this chapter was initially presented in:

[104] **Van Nguyen Nguyen**, Yinlin Hu, Yang Xiao, Mathieu Salzmann, Vincent Lepetit. *Templates for 3D Object Pose Estimation Revisited: Generalization to New Objects and Robustness to Occlusions*. Computer Vision and Pattern Recognition (CVPR), 2022.

As discussed in Section 1.4, template-based approaches, such as [148, 5] offer the promise of generalizing to arbitrary new objects by learning an image embedding used to match the input image to a series of templates generated from their CAD models. Unfortunately, their use with new objects has been demonstrated only anecdotally, and we show in our experiments that these methods struggle in this challenging scenario, particularly in the presence of occlusions. We indeed notice that the global representations used in [148, 5] to compare the input image to the CAD-generated templates have two limitations. First, they generalize poorly to new objects in the presence of a cluttered background, and result in inaccurate pose estimation even for uniform background. Furthermore, they are ill-suited to handle occlusions.

These observations motivate us to keep the 2D structure of the images for a template-based approach. More precisely, given a small set of training objects, we learn local features that can be used to reliably match real images and synthetic templates. Relying on local features allows us to discard the background: While the object's mask in the input image is not available at run-time, we can use the template's mask, thus solving the first limitation of global representations. Note that using the template's mask to instead remove the background in the real image before computing the image global representation requires us to recompute the input image representation for each template, which would result in very slow matching.

As will be shown in this chapter, using local features also results in much more accurate poses. This can be explained by the fact that we do not use pooling operations, which remove critical information about the poses, especially for new objects. Finally, yet another advantage is that our method can be robust to partial occlusions. To do so, we introduce a measure to evaluate the similarity between two images that explicitly takes into account the object's mask in the template and the possible occlusions in the query image.

We demonstrate the benefits of our approach on the Linemod [46], Linemod-Occluded [8], and T-LESS [54] datasets. It consistently outperforms previous works [148, 5, 128, 127] on new objects by a large margin.

4.1 Contrastive learning

Given a collection of images, contrastive learning aims to learn an embedding space where similar images are close to each other while dissimilar ones are far apart. [48, 149, 105, 131, 43, 18] leverage unlabeled images and strong data augmentation to learn powerful image features that achieve results competitive with those of supervised learning on various downstream tasks.

In our context, PoseContrast [153] exploits a form of contrastive learning, leveraging the pose labels to learn a pose-aware embedding space for class-agnostic 3D object pose estimation. One limitation of PoseContrast is that different objects can be mixed with each other in the embedding space, thus making it impossible to recognize the correct object instance from the input image. Moreover, like PosefromShape [154], PoseContrast [153] does not attempt to recognize the object.

By contrast, [148, 5] rely on contrastive learning to learn an embedding space that is variant to both the object pose and the object instance. To this end, they rely on a triplet loss for learning object-discriminative features, together with a pairwise loss

for pose-discriminative features. Similarly, we use contrastive learning to extract a discriminative feature representation, but we show that the InfoNCE [105] loss is the most simple and effective choice. Our experiments also show that most of the performance of our method in terms of generalization and robustness to occlusions come from our use of local representations.

4.2 Method

Our goal is to recognize new objects in color images and predict their 3D poses. We do this by matching the color image of the object with a set of templates. A template is a rendered image of a 3D model in some 3D pose. For each new object, the template set contains many templates, rendered from different views sampled around its 3D model. As the templates are annotated with the object's identity and pose, the method returns the identity and pose of the template most similar to the input image.

The challenge then is to measure the similarity between templates and input images. This should be done reliably despite that no real images of the new objects have been seen beforehand, the objects can be partially occluded, the lighting differs between the templates and the real images, and the object's background is cluttered in the real images.

In this work, motivated by the better repeatability and robustness to occlusions of local representations compared to global ones, we measure the similarity between an input image and a template based on local image features extracted using a deep model. We train this model using pairs made of a real image and a synthetic image from a small set of training objects. Note that these training objects can be very different in appearance from the new objects.

We start this section with an analysis of the limits of global representations in Section 4.2.1. We then detail in Section 4.2.2 our training procedure. It relies on a similarity measure that compares the local features of real images and synthetic templates. At run-time, we use an extended version of this similarity function that explicitly estimates which local features in the input image are occluded and discards them. We discuss this in Section 4.2.3. Finally, we detail how we generate the templates in Section 4.2.4.

4.2.1 Motivation and analysis

We present in this section two experiments that point out the main drawbacks of global representations in template matching when working with unseen objects.

Cluttered background. A first drawback of global representations is their poor ability to represent unseen objects on cluttered backgrounds. To show this, we plot in Figure 4.2 the t-SNE visualization [136] of the global representations learned by [5] and local representations learned by our method for real images of training and new objects of the Linemod [46] dataset.

The first column of Figure 4.2 shows that both representations manage to cluster the images of each training object together, despite the fact that the images of the objects are captured with a cluttered background. The second column shows that

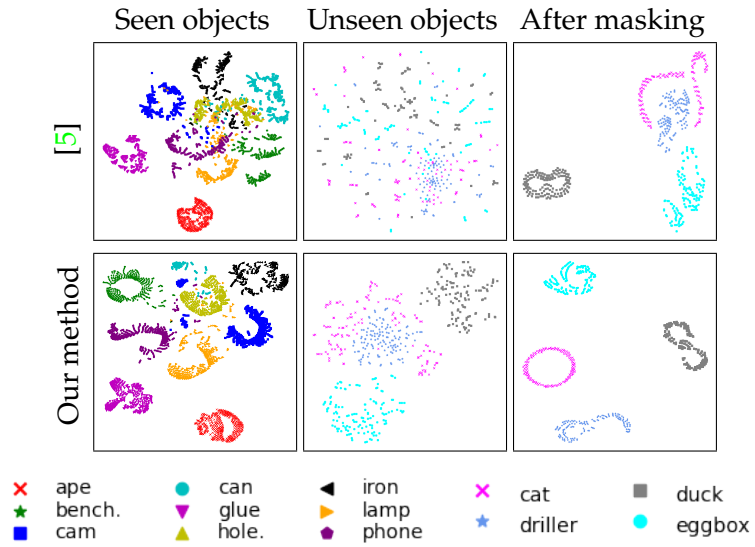


Figure 4.2: **Understanding the influence of background on different image representations,** with T-SNE visualizations of the image representations learned by [5] (first row) and by our method (second row) for real images of Linemod objects. For a given column, all the plots have the same scale for comparison.

global representations of [5] cannot disentangle the images of unseen objects, while our representations can. To better understand the reason behind this, we remove the background in the images by replacing it with a uniform color using the ground-truth object masks. As shown in the third column, the representations are now disentangled. This shows the influence of the background on the global representations for unseen objects, and that our representations are robust to cluttered backgrounds.

Pose discrimination. A second drawback of global representations is their poor reliability when matching the real image of an unseen object with the synthetic template for the corresponding 3D pose, even when the object identity is known and the background is uniform. This can be explained by the fact that the pooling layers remove important information. This information loss appears to be compensated by the rest of the architecture for the training objects, but this compensation does not generalize to unseen objects.

To show this, we visualize in Figure 4.3 the correlation between pose distances and representation distances for unseen objects, as done in [148, 5]. While both representations result in a strong correlation for training objects, this correlation is lost when considering unseen objects for the global representations but not for ours. Even without background, the correlation is still very low for global representations [5].

4.2.2 Framework

In each training iteration, we sample N positive pairs, where pair i is composed of a real image $\mathcal{Q}_i$ depicting a training object and of a synthetic template $\mathcal{T}_i$ of the same object in a similar 3D pose. Following [148], we deem the two viewpoints similar if the angle between them is less than 5 degrees. All the pairs composed by a real image and a synthetic image of different objects or dissimilar poses (larger than 5 degrees) are defined as negative pairs.

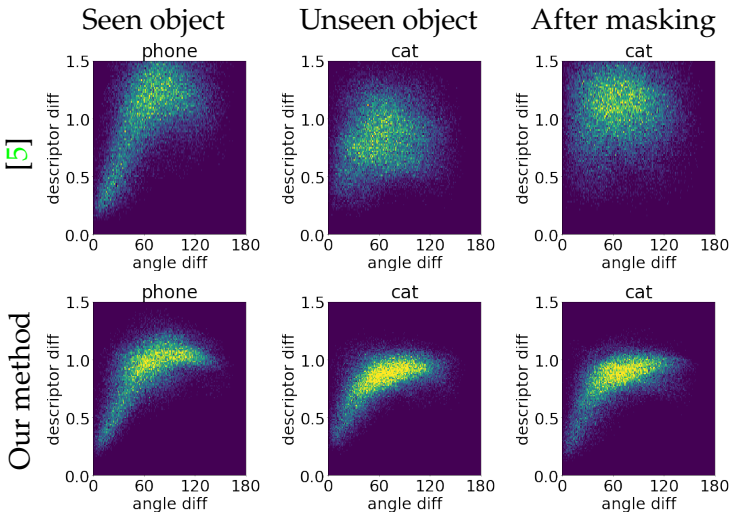


Figure 4.3: **Visualization of the correlation between pose distances and representation distances,** for understanding the discriminative power of different image representations for pose retrieval. First row is for [5], second row is for our method. Please see Section 4.2.1.

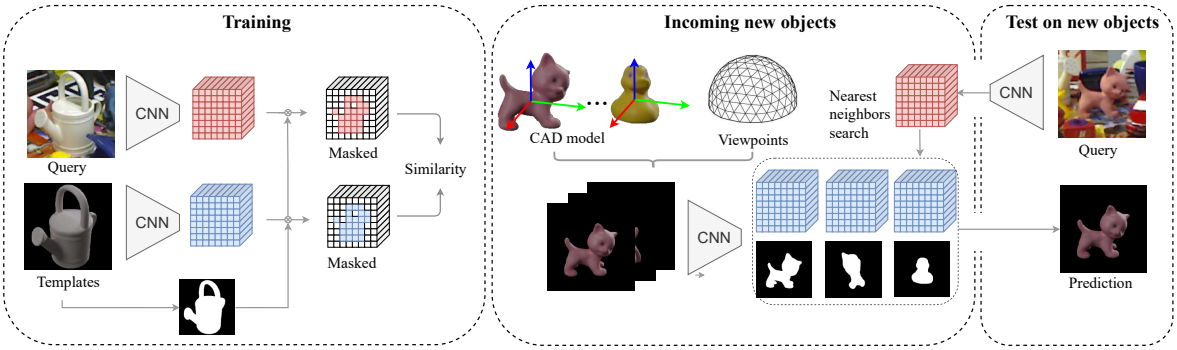


Figure 4.4: **At training time**, we use pairs made of a real image and a synthetic template to train a network to compute local features, from which the similarity between the two images can be predicted. **At run-time**, we apply this network to images of objects not seen during training to compute their local features. We can then retrieve the object pose by matching the image against the database of templates.

Triplet loss. [148] proposed a metric learning approach based on the intuition that the distance between feature descriptors for positive pairs should be closer in the learnt embedding space than negative pairs. To learn this property, [148] used a training loss $\mathcal{L} = \mathcal{L}_{triplet} + \mathcal{L}_{pair}$ where:

- $\mathcal{L}_{triplet}$ is the triplet term, which allows the network to learn features such that the distance in the learned embedding space between the positive pairs $\Delta_+^{(i)}$ is lower than the distance between the negative pairs $\Delta_-^{(i)}$ within the limits of the margin m. This triplet term is defined as

$$\mathcal{L}_{triplet} = \sum_{i=1}^{N} \max\left(0, 1 - \frac{\Delta_+^{(i)}}{\Delta_-^{(i)} + m}\right) \tag{4.1}$$

- $\mathcal{L}_{pair} = \sum_{i=1}^{N} \Delta_+^{(i)}$ is the pairwise term, to minimise distances between two images of identical poses but different viewing conditions.

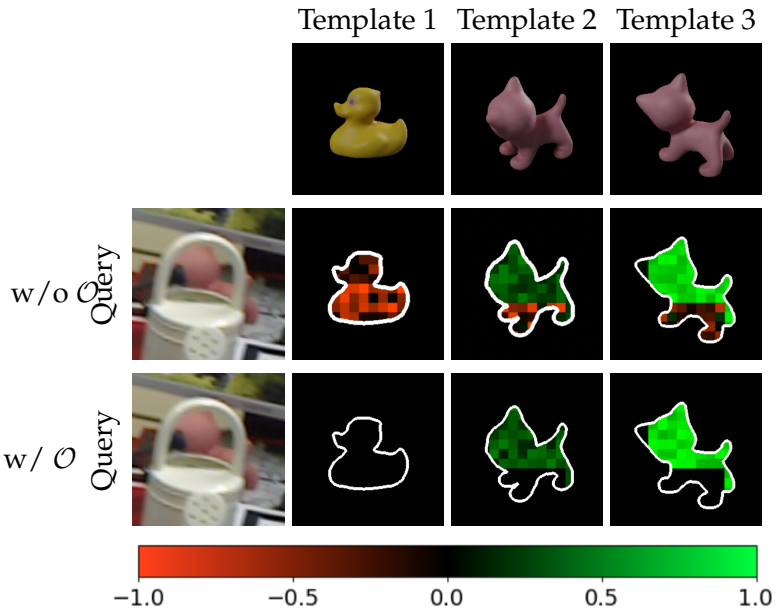


Figure 4.5: **Illustration of feature similarity** when not using the occlusion mask $\mathcal{O}$ (second row) and when using it. As discussed in Section 4.2.3, using $\mathcal{O}$ allows "turning off" the possible occluded local features in the similarity score.

[5] made an extension of this work by proposing a triplet loss which focuses only on learning object-discriminative features while using a pairwise loss to learn an embedding space analogous to the pose differences.

While these two losses work well, we experimentally show that the recent standard contrast loss InfoNCE [105] is the most simple and effective choice.

InfoNCE loss. For each real image $\mathcal{Q}_i$, we also create $N-1$ negative pairs by combining it with synthetic templates $\mathcal{T}_k$ of other pairs in the current batch, with $1 \leq k \leq N, k \neq i$. Altogether, this yields N positive pairs and $(N-1) \times N$ negative pairs for each batch. We train our model to maximize the agreement between the representations of samples in positive pairs, while minimizing that of negative pairs with the InfoNCE loss function [105]:

$$\mathcal{L} = -\sum_{i=1}^{N} \log \frac{\exp\left(\text{sim}(\mathbf{q}_i, \mathbf{t}_i)/\tau\right)}{\sum_{k=1}^{N} 1_{[k \neq i]} \exp\left(\text{sim}(\mathbf{q}_i, \mathbf{t}_k)/\tau\right)}, \tag{4.2}$$

where $\text{sim}(\mathbf{q}, \mathbf{t})$ measures the similarity between the local image features $\mathbf{q}$ and $\mathbf{t}$ computed by the deep model for real image $\mathcal{Q}$ and template $\mathcal{T}$, and $\tau = 0.1$, is a temperature parameter. As shown in Figure 4.4, $\mathbf{q}$ and $\mathbf{t}$ retain a grid structure and are 3-tensors. In practice, their dimensions depend on the size of the input image, ranging from $25 \times 25 \times C$ to $28 \times 28 \times C$, with $C = 16$.

Local feature similarity. While previous works on contrastive learning [105, 131, 96, 14, 43, 20, 18, 19] focused mostly on image classification and define the similarity metric $\text{sim}(.,.)$ using a global representation of the two images, we found such a representation to only classify well either known objects or images with a clean background, as discussed in Section 4.2.1. To effectively handle new objects and

complex backgrounds, we use a metric based on a pairwise comparison of the local features in $\mathbf{q}$ and $\mathbf{t}$. Specifically, we define

$$\text{sim}(\mathbf{q}, \mathbf{t}) = \frac{1}{|\mathbf{m}_\mathcal{T}|} \sum_l \mathbf{m}_\mathcal{T}^{(l)} \mathcal{S}\left(\mathbf{q}^{(l)}, \mathbf{t}^{(l)}\right) , \tag{4.3}$$

where $\mathcal{S}$ is a local similarity metric, $\mathbf{m}_\mathcal{T}$ is a 2D binary visibility mask for template $\mathcal{T}$, and index l indicates a 2D grid location. $\mathbf{q}^{(l)}$ and $\mathbf{t}^{(l)}$ are thus local features of dimension C. Considering the template mask allows us to discard the background in the real image. Note that the mask does not account for possible occlusions in the real image as it corresponds to the object's silhouette in the template. Occlusions will be considered in the next subsection. As a local similarity metric $\mathcal{S}$, we use the cosine similarity:

$$\mathcal{S}\left(\mathbf{q}^{(l)}, \mathbf{t}^{(l)}\right) = \frac{\mathbf{q}^{(l)}}{||\mathbf{q}^{(l)}||_2} \cdot \frac{\mathbf{t}^{(l)}}{||\mathbf{t}^{(l)}||_2} , \tag{4.4}$$

We empirically observed that measuring the similarity as the opposite of the L1 and L2 norms of the differences yields the same performance as the cosine similarity.

4.2.3 Run-time and robustness to occlusions

At run-time, given a real query image $\mathcal{Q}$, we retrieve the most similar template. To be robust to occlusions that can occur in the query image, we modify $\text{sim}(\mathbf{q}, \mathbf{t})$ as:

$$\text{sim}^*(\mathbf{q}, \mathbf{t}) = \frac{1}{|\mathbf{m}_\mathcal{T}|} \sum_l \mathbf{m}_\mathcal{T}^{(l)} \mathcal{O}^{(l)} \mathcal{S}\left(\mathbf{q}^{(l)}, \mathbf{t}^{(l)}\right) , \tag{4.5}$$

where $\mathcal{O}^{(l)} = 1_{\mathcal{S}(\mathbf{q}^{(l)}, \mathbf{t}^{(l)}) > \delta}$ with δ a threshold applied to the cosine similarity to "turn off" the occluded local features as shown in Figure 4.5. In practice, we set this threshold $\delta = 0.2$ through ablation study. Note that Equation 4.5 can be written as the element-wise product $\odot$ and can be computed efficiently with:

$$\text{sim}^*(\mathbf{q}, \mathbf{t}) = \frac{1}{|\mathbf{m}_\mathcal{T}|} (\mathbf{m}_\mathcal{T} \odot \mathcal{O} \odot \mathcal{S}) . \tag{4.6}$$

4.2.4 Template creation

On Linemod [46] and Linemod-Occluded [8] datasets, we follow the protocol of [148] to sample the synthetic templates. More precisely, the viewpoints are defined by starting with a regular icosahedron and recursively subdividing each triangle into 4 smaller triangles. After applying this subdivision two times and removing the lower half-sphere, we end up with 301 templates per object.

On T-LESS [54], we follow the protocol of [127] by using a dense regular icosahedron with 2 562 viewpoints and 36 in-plane rotations for each rendered image. Altogether, this yields 92 232 templates per object. Besides, we also show our results with a coarser regular icosahedron with 642 viewpoints, which results 21 672 templates per object. We use BlenderProc [26] to generate templates with realistic rendering for both settings.

4.2.5 Projective distance estimation

As done in [128, 127], we estimate 3D translation in the query image from the retrieved template and the input bounding box as detailed in Section 3.6.2 of [129]. More precisely, given known camera intrinsic of both real sensor $K_{\mathcal{Q}}$ and of the synthetic view $K_{\mathcal{T}}$, we estimate the coordinate of the translation in Z-axis $z_{\mathcal{Q}}$ of real image:

$$\hat{z}_{\mathcal{Q}} = z_{\mathcal{T}} \times \frac{||B_{\mathcal{T}}||_2}{||B_{\mathcal{Q}}||_2} \times \frac{f_{\mathcal{Q}}}{f_{\mathcal{T}}} \tag{4.7}$$

where $||B_{(.)}||_2$ is the diagonal of the bounding box and $||f_{(.)}||_2$ is the focal length.

Then, we can estimate the vector to transform from the object center in the synthetic image $\mathcal{T}$ to the query image $\mathcal{Q}$:

$$\Delta\hat{t} = \hat{z}_{\mathcal{Q}} K_{\mathcal{Q}}^{-1} B_{\mathcal{Q},c} - z_{\mathcal{T}} K_{\mathcal{T}}^{-1} B_{\mathcal{T},c} \tag{4.8}$$

where $B_{(.),c}$ is the bounding box centers in homogeneous coordinates.

Finally, the 3D translation in the query image $\hat{t}_{\mathcal{Q}}$ can be estimated as :

$$\hat{t}_{\mathcal{Q}} = t_{\mathcal{T}} + \Delta\hat{t} \tag{4.9}$$

where $t_{\mathcal{T}}$, the translation from camera to object center in the synthetic view $\mathcal{T}$.

4.3 Experiments

In this section, we first describe the experimental setup (Section 4.3.1). Then, we compare quantitatively and qualitatively our method with previous works [148, 5, 128, 127] on both seen and unseen objects of the Linemod (LM) [46], Linemod-Occluded (LM-O) [8] and T-LESS [54] datasets (Section 4.3.2). Finally, we provide an ablation study for investigating the effectiveness of our method with different parameters and failure cases of our method (Sections 4.3.3 and 4.3.4).

4.3.1 Experimental setup

Data processing. For the LM and LM-O datasets, as there are no standard splits to evaluate the robustness of RGB-based methods on unseen objects, we propose three different splits created from the order of the object ids. The new, or unseen objects for each of these splits are:

- Split #1: **Ape**, Benchvise, Camera, **Can**;
- Split #2: **Cat**, **Driller**, **Duck**, **Eggbox**;
- Split #3: **Glue**, **Holepuncher**, Iron, Lamp, Phone.

The other objects from LM are used for training the model. The objects with names in **bold** in the lists above often are occluded in LM-O. Note that LM-O is only used for testing, as we do not need to see occlusions during the training time. Moreover, to understand the performance gap between objects that are seen or unseen during the training, we also evaluate the methods on seen objects. To do so, we keep 10% of the real images of training objects under unseen poses for testing purposes. Table 4.1 details the different splits.

On T-LESS [54], we follow the evaluation protocol of [127] by training only on objects 1-18 under randomized backgrounds of SUN397 [152] and testing on the complete T-LESS primesense test set.

Split	Training	Seen LM	Seen LM-O	Unseen LM	Unseen LM-O
#1	9 954	981	6 832	4 848	2 377
#2	9 928	981	4 490	4 874	4 719
#3	8 850	872	7 096	6 061	2 113

Table 4.1: **Dataset splits for LM and LM-O.** For each split, we provide the numbers of test instances in the training set and in four test sets.

Evaluation metrics. For the LM and LM-O datasets, the pose error is measured by the angle between the two positions on the viewing half-sphere. We also treat the "Eggbox" and "Glue" objects as symmetric around the z-axis as done in [148, 5].

In the case of known object pose estimation, the recognition score is almost 100% on LM and LM-O. Previous works [148, 5] that focused on known objects thus only evaluate the pose error without considering whether the retrieved object is actually correct. In the case of unseen objects, we found that retrieving correctly both pose and class is important as the model can still get correct poses but from another object. Therefore, we propose using the Acc15 metric, which measures how often the pose error is less than 15 degrees *and* the predicted object class is correct.

As most objects in T-LESS [54] are symmetric, we report the recall under the Visible Surface Discrepancy (err_{vsd}) metric at $\text{err}_{vsd} < 0.3$ with tolerance $\tau = 20mm$ and $> 10\%$ object visibility as done in [128, 127]. Unless otherwise stated in previous works [128, 127], only templates of the same object are used at testing time (in other words, the class of the object is assumed to be known before testing). Please note that for the evaluation on the T-LESS dataset, we also predict the translation by using the same formula "projective distance estimation" of SSD-6D [65] as done in [128, 127]. This translation is deduced from the retrieved template and the input bounding box of query image as discussed in Section 4.2.5.

Implementation details. For a fair comparison, in the evaluation on LM and LM-O, we consider two different backbones: (i) "Base" – the simple backbone used in [148, 5]; (ii) ResNet50 – the standard backbone used in recent contrastive learning methods [43]. We reimplemented [148, 5] to get quantitative results in both seen and unseen objects. Our implementations get very similar performance when evaluated on the same data as the original papers on seen objects (see Table 4.2), validating our reimplementation.

We also follow [148, 5] when testing with the "Base" backbone by using the same input image of size 64×64. While testing with ResNet50, we use a larger input size of 224×224. In both settings, we slightly change the architecture by removing all the pooling, FC layers and then replace them by two 1×1 convolution layers to output the desired local feature of size 16. As done in [148, 5], we use the ground-truth pose to crop the input image at the center of objects and do not consider in-plane rotation. On the T-LESS dataset, we use the same backbone ResNet50 and crop the input image with ground-truth bounding box as done in [128, 127].

For both evaluations, we train our networks using Adam with an initial learning rate of 1e-2 for the "Base" backbone and of 1e-4 for ResNet50. Training takes less than 5h for all splits on a single V100 GPU when training on LM [46] and around 12h when training on T-LESS [54].

Method	Backbone	Features	Loss	Seen LM				Seen LM-O				Unseen LM				Unseen LM-O			
				#1	#2	#3	Avg.	#1	#2	#3	Avg.	#1	#2	#3	Avg.	#1	#2	#3	Avg.
[148]	Base	Global	[148]	87.0	83.1	85.1	85.0	19.2	23.1	15.0	19.1	13.2	15.5	18.2	15.2	9.3	5.1	5.1	6.5
[148]	Base	Global	InfoNCE	95.2	95.3	95.4	95.3	19.6	25.3	16.1	20.3	13.3	17.0	20.5	16.9	8.2	6.4	6.7	7.1
[5]	Base	Global	[5]	89.2	85.4	83.3	86.3	18.3	21.9	17.6	19.5	14.1	16.3	19.7	16.7	8.2	7.5	7.6	7.8
[5]	Base	Global	InfoNCE	96.3	95.2	96.5	96.0	18.3	23.1	15.8	19.1	11.5	17.7	17.2	15.5	7.1	6.5	6.5	6.7
Ours	Base	Local	[148]	84.8	85.5	86.3	85.5	50.1	51.3	42.2	47.9	69.6	63.2	46.2	59.7	35.3	34.3	44.2	37.9
Ours	Base	Local	InfoNCE	95.6	96.9	92.0	94.8	68.9	71.0	57.7	65.8	78.8	82.5	64.1	75.1	42.2	57.1	59.8	53.0
[148]	ResNet	Global	InfoNCE	98.8	96.9	98.8	98.1	66.7	73.2	62.7	67.5	42.2	43.7	49.4	45.1	22.3	22.5	45.9	29.9
[5]	ResNet	Global	InfoNCE	96.9	97.1	94.5	96.1	63.6	71.8	58.9	64.7	39.9	44.9	48.3	44.3	15.5	21.8	50.2	29.1
Ours	ResNet	Local	InfoNCE	**99.3**	**99.0**	**99.2**	**99.1**	**77.3**	**84.1**	**76.8**	**79.4**	**94.4**	**97.4**	**88.7**	**93.5**	**71.4**	**72.7**	**85.3**	**76.3**

Table 4.2: **Comparison of our method with [148] and [5]** on seen and unseen objects of LM and LM-O under the three different splits detailed at the beginning of Section 4.3.1. We report Acc15 ↑, the accuracy of predicting correctly the object identity *and* its pose with an error less than 15 degrees. We are on par on the "easy" case and outperform them by a large margin on the 3 other configurations. Using the InfoNCE loss rather than the loss from [5] brings some improvement, but the main improvement comes from our approach based on local features.

4.3.2 Comparison with the state of the art

Results on Linemod and Linemod-Occluded. Table 4.2 presents the results of our method compared with previous work [148, 5]. With either the "Base" or ResNet50 backbones, our method based on local feature similarities achieves the best overall performance in almost all settings compared to previous methods that compute the feature similarity between global image representations. While [148, 5] explored carefully designed pairwise and triplet losses for learning an embedding space that is both object-discriminative and pose-discriminative, we find that using the InfoNCE loss as defined in InfoNCE [105] boosts the performance of all methods, in particular for our method based on local feature similarities.

When the objects are occluded, the accuracy of [148, 5] drops to below 70% for training objects, while our method can still maintain a relatively high accuracy. This shows the robustness of local image features rather than global image representations that are much more strongly affected by the occlusions. Furthermore, the prediction accuracy of our method on unseen objects is clearly higher than that of previous methods, regardless of the objects being occluded or not. This indicates that matching based on local features is not only robust to occlusions, but also generalizes better to unseen objects. More importantly, this improvement on unseen objects holds still in the presence of occlusions.

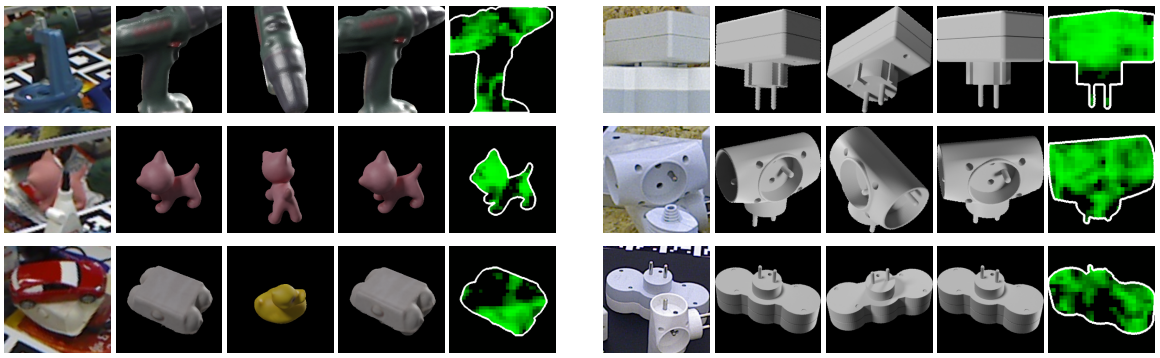
Results on T-LESS In Table 4.3, we shown that our proposed approach outperforms the state-of-the-art methods [128, 127] on the T-LESS dataset by a large margin on both seen and unseen objects. While [127] carefully designed single-encoder-multi-decoder network that allows sharing a latent space for all objects and having each decoder only reconstructs views of a single object, we find that using our method and InfoNCE loss is much more simple but also boosts significantly the performance in the same setting.

We show in Figure 4.6 our qualitative comparisons with previous methods [5,

Method	Number templates	Recall VSD		
		Obj. 1-18	Obj. 19-30	Average
Implicit [128]	92K	35.60	42.45	38.34
MultiPath [127]	92K	35.25	33.17	34.42
Ours	92K	**58.62**	**58.44**	**58.54**
Ours	21K	59.14	56.91	58.25

Table 4.3: **Comparison with [128, 127]** on seen objects (obj. 1-18) and unseen objects (obj. 19-31) of T-LESS using the protocol from [127]. Our method significantly outperforms [128, 127] in the same setting.

127] on Linemod-Occluded [8] and T-LESS [54] datasets.



Query GT [5] Ours Similarity Query GT [127] Ours Similarity

Figure 4.6: **Qualitative results on unseen objects** of Linemod-Occluded [8] (left) and T-LESS [54] (right). Our method retrieves the correct template and pose while [5, 127] fails on unseen objects, particularly in the presence of occlusion.

4.3.3 Ablation study

We present several ablation evaluations on Linemod [46] and Linemod-Occluded [8].

Effectiveness of feature masking. Table 4.4 shows the effectiveness of using the template masks $\mathcal{M}$ in Equation 4.6 for unseen objects. Removing $\mathcal{M}$ results in a dramatic degradation for our method on all the three splits.

Influence of the threshold δ. Table 4.5 shows the influence of the threshold δ in Equation 4.5 for estimating the occlusion mask $\mathcal{O}$. Using $\mathcal{O}$ brings improvements on large objects ("Can", "Driller", and "Eggbox"). This can be explained by the fact that the occlusions can be very large in LM-O, especially on small objects, as shown in Figure 4.7.

Influence of the local feature dimensions. Figure 4.8 shows the pose error as a function of the dimension C of the local features and of the resolution of the feature maps and masks $\mathcal{M}$. While C is not a critical value, the resolution is more important,

	Ape	Can	Cat	Driller	Duck	Egg*	Glue*	Hole.	Avg
[148]	16.6	28.0	1.5	8.2	11.5	68.8	67.7	22.1	29.9
[5]	12.6	18.4	9.0	16.7	7.8	53.7	60.3	40.1	29.1
Ours	**53.8**	**89.7**	**45.1**	**84.4**	**87.2**	**76.9**	**89.9**	**83.3**	**76.3**
w/o $\mathcal{M}$	13.3	1.0	10.0	1.0	80.1	7.0	80.0	1.0	24.1

Table 4.4: **Effectiveness of $\mathcal{M}$.** Comparison of [148, 5] and our method with and without using the template mask $\mathcal{M}$ in the computation of the similarity. Using $\mathcal{M}$ allows discarding the cluttered background and brings significant improvement on occluded unseen objects.

Threshold δ	-0.3	-0.2	-0.1	0	0.1	0.2	0.3	w/o $\mathcal{O}$
Ape	54.1	53.7	54.6	**54.7**	54.0	53.8	53.6	53.3
Can	82.2	89.2	89.1	89.7	89.4	89.7	**89.8**	84.9
Cat	46.7	**47.5**	46.1	45.5	46.1	45.1	46.5	45.1
Driller	83.6	**84.5**	**84.5**	83.8	84.4	84.4	**84.5**	81.5
Duck	87.1	87.1	**87.8**	86.7	87.3	87.2	87.0	87.3
Egg*	76.3	75.2	74.1	75.3	75.1	**76.9**	76.2	72.6
Glue*	89.3	83.5	83.9	90.1	89.5	89.9	89.6	**90.2**
Holep.	83.9	**85.9**	83.6	82.9	83.4	83.3	82.5	81.8
Avg	75.4	75.8	75.4	76.0	76.1	**76.3**	76.2	74.5

Table 4.5: **Influence of threshold δ of Equation 4.5.** Predicting occlusion mask $\mathcal{O}$ with threshold $\delta = 0.2$ results on the best performance, particularly on large objects.

Figure 4.7: The “Cat" object is often barely visible in the test images of Linemod-Occluded, resulting in large errors.

Dataset	Number templates	Features creation	Memory	Run-time	
				CPU	GPU
Linemod	1.204	0.5 min	28 MB	0.15 s	7.8×10^{-3} s
T-LESS	21.672	6 min	544 MB	0.84 s	8.2×10^{-3} s

Table 4.6: **Average run-time** of our method on a single GPU V100 and CPU Intel Xeon.

as higher resolution allows discarding the background more precisely. Furthermore, this hyperparameter has a much stronger influence on the performance on the unseen objects compared to the seen objects.

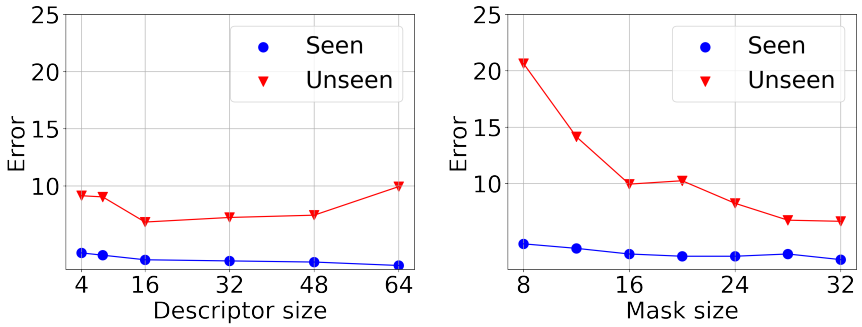


Figure 4.8: **Influence of the local feature dimension C and of the resolution of the local features and masks.** Using a good resolution is much more important than using high-dimensional local features as this allows discarding background more precisely when computing the similarity score.

Run-time. Table 4.6 provides run-times on CPU and GPU.

4.3.4 Failure cases

When evaluated on Linemod-Occluded, both our method and [148, 5] fail on the "Cat" object. As shown in Figure 4.7, this object is small and particularly heavily occluded in this dataset.

4.4 Conclusion

In this chapter, we have presented an efficient approach to 3D object recognition and pose estimation that can generalize to new objects without the need for retraining and that is robust to occlusions. Our analysis has shown that a global representation, which discards the grid structure of images, is not robust to clutter and results in inaccurate pose predictions. Our method, based on local representations, has much better properties and can be made robust to occlusions. We hope that our analysis and our new approach will guide the development of more practical systems.

Chapter 5

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

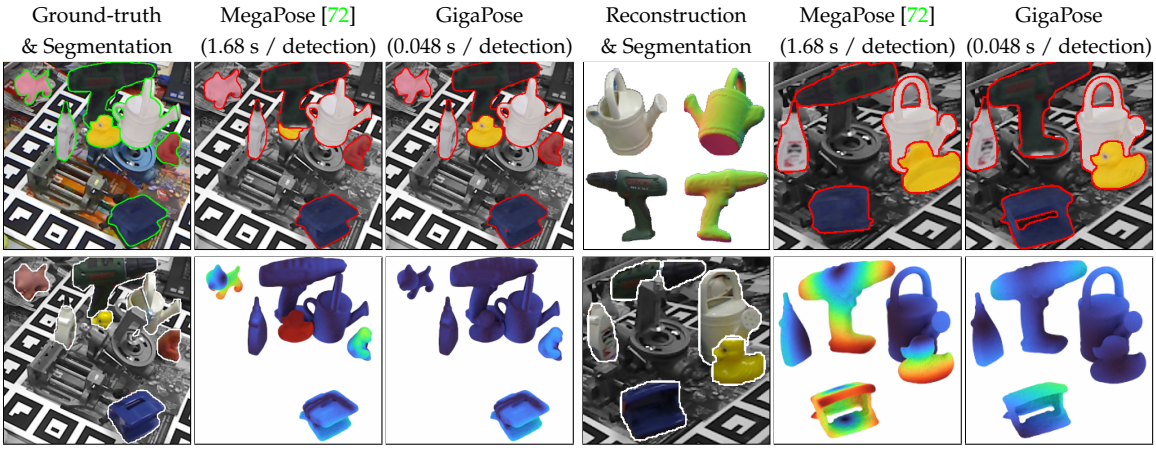


Figure 5.1: **Comparison of our method GigaPose with MegaPose [72].** GigaPose is (i) more robust to noisy segmentation, often due to occlusions, (ii) more accurate with 3.5 % average precision improvement on the BOP benchmark [51], and (iii) significantly faster with a speed up factor of 35× per detection for coarse object pose estimation stage (0.048 s vs 1.68 s). Left example compares the results using accurate 3D models, while the right example shows the results with 3D models predicted from a single image by Wonder3D [84]. The bottom row shows the input segmentation, and the depth error heatmap of each detected object with respect to the ground truth pose, i.e the distance between each 3D point in the ground-truth depth map and its position with the predicted pose (legend: 0 cm 10 cm).

The work presented in this chapter was initially presented in:

[103] **Van Nguyen Nguyen**, Thibault Groueix, Mathieu Salzmann, Vincent Lepetit. *GigaPose: Fast and Robust Novel Object Pose Estimation via One Correspondence*. Computer Vision and Pattern Recognition (CVPR), 2024.

As discussed in Section 2, most of current pose estimation approaches involve two main steps: object detection and segmentation, pose estimation including coarse pose estimation, and refinement. While object detection and segmentation has been recently addressed by CNOS [102], introduced in Chapter 3, refinement has been also addressed effectively with render-and-compare approaches [72, 133]. However, existing solutions to coarse pose estimation still suffer from low inference speed and sensitivity to segmentation errors. We thus focus on this step in this chapter.

The low inference speed stems from how existing coarse pose estimation methods rely on templates [104, 159, 72]. Among them, MegaPose [72] has been widely adopted and integrated into various pipelines, notably the BOP challenge-winner GenFlow [97]. However, the complexity of MegaPose is linear in the number of templates, as it matches the input images against the templates by running a network on each image-template pair. As a result, the methods based on MegaPose require more than 1.6 seconds *per detection*.

Sensitivity to detection and segmentation errors, often due to occlusions, is a common issue for template-based approaches [104, 72]. As illustrated in Figure 5.1, the segmentation of occluded objects such as the "duck" (left example), results in a scale and translation mismatch when cropping the test image and templates. Additionally, the erroneous segments may include noisy signal from the other objects or the background, which results in numerous outlier matches between the input image and the templates.

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

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

To estimate the remaining 4 DoFs—in-plane rotation and 3D translation decomposed into 2D translation and 2D scale, we rely on patch correspondences between the input image and the template candidates. Given a template candidate, we match its local features with those of the input image, which gives us 2D-2D point correspondences. Instead of simply exploiting the matched point coordinates and use a PnP algorithm [98] to estimate the pose as done in previous works [115, 58, 59], we also exploit their appearances: We show that it is possible to predict the in-plane rotation and relative scale between the input image and the template from local features computed at the matched points. The remaining 2D translation is obtained from the positions of these matched points, allowing the estimation of the four DoFs from a single correspondence. To robustify this estimate, we combine this process with RANSAC.

We experimentally demonstrate that our balance between the use of templates

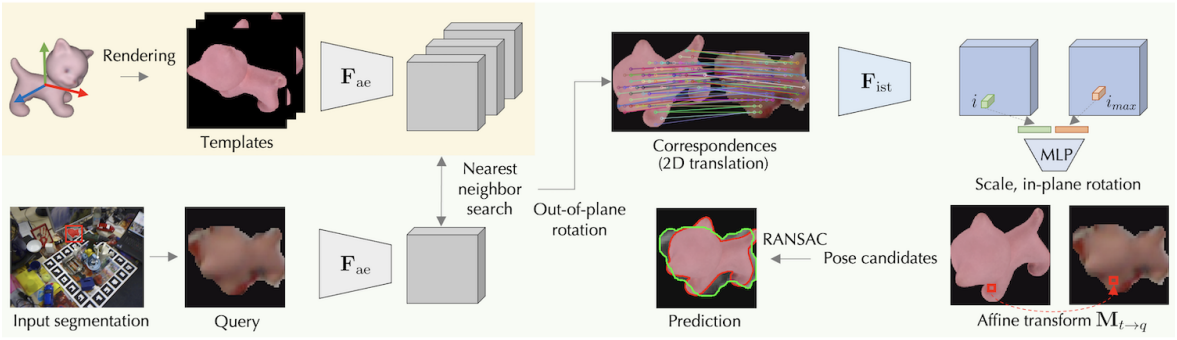


Figure 5.2: **Overview.** We first onboard each novel object by rendering 162 templates, spanning the spectrum of out-of-plane rotations. We also extract dense features using $\mathbf{F}_{ae}$ from each of the templates. At runtime, given a query image segmented with CNOS [102], we process it (by masking the background, cropping on the segment, adding padding then resizing), and extracting features with $\mathbf{F}_{ae}$. We retrieve the nearest template to the segment using the similarity metric detailed in Section 5.1.3. Further, 2D scale and in-plane rotation are computed from a single 2D-2D correspondence using $\mathbf{F}_{ist}$ and two lightweight MLPs. The 2D position of the correspondences also gives us the 2D translation which is used with 2D scale, in-plane rotation to create the affine transformation $\mathbf{M}_{t\to q}$, mapping the nearest template to the query image. This enables us to recover the complete 6D object pose from a single correspondence. Finally, we use RANSAC to robustly find the best pose candidate. Onboarding takes 11.5 seconds per object and inference takes 48 milliseconds per detection on average.

and patch correspondences effectively addresses the two issues in coarse pose estimation. Indeed, our method relies on a sublinear nearest-neighbor template search, successfully addressing the low inference speed issue with a speedup factor of 35× per detection compared to to MegaPose [72]. Furthermore, the two steps of our method are particularly robust to segmentation errors.

We also demonstrate that GigaPose can exploit a 3D model reconstructed from a single image by a diffusion-based model [84, 82, 80, 114, 112, 142] instead of an accurate CAD model. Despite the inaccuracies of the predicted 3D models, our method can recover an accurate 6D pose as shown on Figure 5.1. This relaxes the need for CAD models and makes 6D pose object detection much more convenient.

5.1 Method

Figure 5.3 provides an overview of GigaPose. Given the 3D model of an object of interest, we render templates and extract their dense features using a Vision-Transformer (ViT) model $\mathbf{F}_{ae}$. Then, given an input image, we detect the object of interest and segment it using an off-the-shelf method CNOS [102]. GigaPose extracts dense features from the input image at the object location using $\mathbf{F}_{ae}$ again. We select the template most similar to the input image using a similarity metric based on the dense features, detailed in Section 5.1.3. This gives us the azimuth and elevation of the camera.

To estimate the remaining DoFs, we look for corresponding patches between the input image and its most similar template. From one such pair of patches, we can directly predict two additional DoFs: the 2D scale s and the in-plane rotation

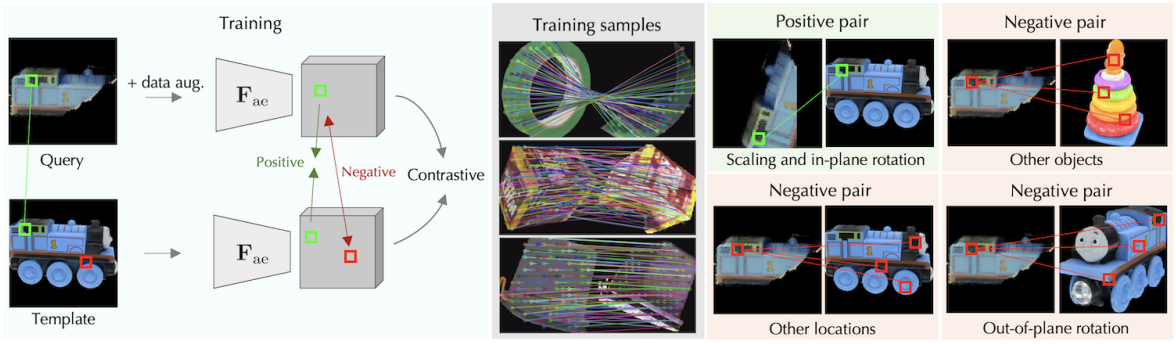


Figure 5.3: **Contrastive training of $\mathbf{F}_{\text{ae}}$.** We use pairs made of a query image and a template to train a network using local contrastive learning as detailed in Section 5.1.3. Middle: Training samples provided by [72], and the 2D-2D correspondences created from ground-truth 3D information used to generate positive and negative pairs. Right: We seek local features that vary with the out-of-plane rotation, but are invariant to in-plane rotation and scaling. Thus, positive pairs are made of corresponding patches under scaling and in-plane rotation changes, and negative pairs are made of corresponding patches under different out-of-plane rotations, patches that do not correspond, or that come from different objects.

α, by feeding two lightweight MLPs the features for the two patches extracted by another feature extractor denoted as $\mathbf{F}_{\text{ist}}$. Note that the features extracted by $\mathbf{F}_{\text{ae}}$ are not suitable here, as they discard information about scale and in-plane rotation by design. The image locations of the corresponding patches also give us directly the last two DoFs: the 2D translation $(\mathbf{t}_x, \mathbf{t}_y)$. From the scale and 2D translation, we can estimate the 3D translation. We use a RANSAC framework and iterate over different pairs of patches to find the optimal pose. We detail the training of $\mathbf{F}_{\text{ist}}$ and the MLPs, and the RANSAC scheme in Section 5.1.4.

5.1.1 Generating templates

In contrast to other approaches [104, 159], we do not generate templates for both in-plane and out-of-plane rotations, as this yields thousands of templates. Instead, we decouple the 6 DoFs object pose into out-of-plane rotation, in-plane rotation, and 3D translation (2D translation and scale). Given the out-of-plane rotation, finding the scale and in-place rotation is indeed a 2D problem only. We thus create much less templates and push the estimation of the other DoFs to a later stage in the pipeline (see Section 5.1.4). In practice, we use 162 templates. These are generated from viewpoints defined in a regular icosphere which is created by subdividing each triangle of the icosphere primitive of Blender into four smaller triangles. This has been shown in previous works [102, 3, 104] to provide well-distributed view coverage of CAD models.

5.1.2 Generating ground-truth 2D-to-2D correspondences

We use the training sets provided by the BOP challenge [51], originally sourced from MegaPose [72] to create the 2D-to-2D correspondences for training. These datasets consist of 2 million images and are generated from CAD models of Google Scanned

Objects [28] and ShapeNet [16] using BlenderProc [26].

For each 2D location i—the 2D center for a patch of size 14×14 of the query image—we aim to identify its corresponding location i^* in the nearest template. We achieve this through a straightforward re-projection process. For each 2D center in the query image, we first calculate its 3D counterpart using the query depth map and camera intrinsics. We then transform this 3D point into the camera view of the nearest template using the ground-truth relative pose, and re-project this 3D point into the template using template camera intrinsics. If the re-projected 2D location falls inside the template mask, we identify the nearest patch i^* among all patches within the template mask, as the corresponding location for the input query patch i. We reverse the roles of the query and the template, then use the same process to establish 2D-to-2D correspondences for each patch of the template.

Additionally, since our goal is to close the domain gap between real images and synthetic renderings, we apply color augmentation including: Gaussian blur, contrast, brightness, colors and sharpness filters from the Pillow library [24] along with random cropping and in-plane rotation to the input pairs . We show typical training samples in the middle part of Figure 5.3.

5.1.3 Predicting azimuth and elevation

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

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

We achieve this with a local contrastive learning scheme. The main difficulty lies in defining the positive and negative patch pairs. Figure 5.3 illustrates our training procedure. We construct batches of B image pairs $(\mathcal{Q}_k, \mathcal{T}_k)$, such that the query $\mathcal{Q}_k$ is a rendering of a 3D object in any pose, and the template $\mathcal{T}_k$ is another rendering of that object with the same out-of-plane rotation but different in-plane rotation, scale, and 2D translation. Because we have access to the ground-truth 2D-to-2D correspondences as detailed in Section 5.1.2, we can create positive and negative pairs for the training.

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

For a location i in the query feature map $\mathbf{q}_k$, we denote by i^* the corresponding location in the template feature map. We arrange the query patches $(\mathbf{q}_k^i)$ and their

corresponding patch ($\mathbf{t}_k^{i^*}$) in a square matrix such that the diagonal contains the positive pairs, and all other entries serve as negative pairs. For each pair $(\mathcal{Q}_k, \mathcal{T}_k)$, we thus obtain $|\mathbf{m}_{\mathcal{Q}_k}|$ positive pairs and $|\mathbf{m}_{\mathcal{Q}_k}| \cdot (|\mathbf{m}_{\mathcal{Q}_k}| - 1)$ negative pairs.

To improve the efficiency of contrastive learning, we use additional negative pairs from a query image $\mathcal{Q}_k$ and a template $\mathcal{T}_{k'}$, where $k' \neq k$ in the current batch. This process results in total in $\left(\sum_{k=1}^{B} |\mathbf{m}_{\mathcal{Q}_k}|\right)^2 - \sum_{k=1}^{B} |\mathbf{m}_{\mathcal{Q}_k}|$ negative pairs for the current batch. We train $\mathbf{F}_{\text{ae}}$ to align the representations of the positive pairs while separating the negative pairs using the InfoNCE loss [105]:

$$\mathcal{L}_{\text{out}} = -\sum_{k=1}^{B} \sum_{i=1}^{|\mathbf{m}_{\mathcal{Q}_k}|} \ln \frac{e^{S(\mathbf{q}_k^i, \mathbf{t}_k^{i^*})/\tau}}{\sum_{(k',i') \neq (k,i^*)} e^{S(\mathbf{q}_k^i, \mathbf{t}_{k'}^{i'})/\tau}} , \tag{5.1}$$

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

Since the positive pairs of patches have different scales and in-plane rotations, our network $\mathbf{F}_{\text{ae}}$ learns to become invariant to these two factors, as demonstrated in our experiments. We initialize our feature extractor $\mathbf{F}_{\text{ae}}$ as DINOv2 [106] pretrained on ImageNet, because it has proven to be highly effective in extracting features for vision tasks.

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

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

$$i_{\text{max}} = \underset{j|\mathrm{m}_{\mathcal{T}}^j > 0}{\arg\max}\, S\left(\mathbf{q}^i, \mathbf{t}^j\right) . \tag{5.2}$$

where $j|\mathrm{m}_{\mathcal{T}}^j > 0$ refers to a subset of indices j for which $\mathrm{m}_{\mathcal{T}}^j > 0$.

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

$$\text{sim}(\mathbf{q}, \mathbf{t}) = \frac{1}{|\mathrm{m}_{\mathcal{Q}}|} \sum_i \mathrm{m}_{\mathcal{Q}}^i S\left(\mathbf{q}^i, \mathbf{t}^{i_{\text{max}}}\right) . \tag{5.3}$$

We compute this score for all templates $\mathcal{T}_k$ $(1 \leq k \leq 162)$ and find the top-K candidates yielding the most similar out-of-plane rotations. This nearest neighbor search is very fast and delivers results within 48 milliseconds. In practice, we experiment with $K = 1$ and $K = 5$. For the latter, the final template is selected by the RANSAC-based estimation detailed in Section 5.1.4 below.

5.1.4 Predicting the remaining DoFs

Once we have identified the template candidates, we seek to estimate the remaining 4 DoFs, i.e., in-plane rotation, scale, and 2D translation, which yield the affine

transformation $\mathbf{M}_{t\to q}$ transforming each template candidate $\mathcal{T}$ to the query image $\mathcal{Q}$. Specifically, we have

$$\mathbf{M}_{t\to q} = \begin{bmatrix} s\cos(\alpha) & -s\sin(\alpha) & \mathbf{t}_x \\ s\sin(\alpha) & s\cos(\alpha) & \mathbf{t}_y \\ 0 & 0 & 1 \end{bmatrix} , \tag{5.4}$$

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

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

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

$$\mathcal{L}_{\text{inp}} = \sum_{k=1}^{B} \sum_{i=1}^{n_k} \left[\left(\ln(s_k^i) - \ln(s_k^*)\right)^2 + \text{geo}(\alpha_k^i, \alpha_k^*) \right] , \tag{5.5}$$

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

$$\text{geo}(\alpha_1, \alpha_2) = \text{acos}\big(\cos(\alpha_1)\cos(\alpha_2) + \sin(\alpha_1)\sin(\alpha_2)\big) . \tag{5.6}$$

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

We initialize $\mathbf{F}_{\text{ist}}$ with a modified version of ResNet18 [45] instead of the DINOv2 [106] as DINOv2 is trained with random augmentations that includes in-plane rotations and cropping, making its features invariant to scale and in-plane rotation. Similarly to the features from $\mathbf{F}_{\text{ae}}$, we offload the feature computation of $\mathbf{F}_{\text{ist}}$ to the onboarding stage for all templates to avoid the computational burden at runtime.

5.2 Experiments

In this section, we first describe our experimental setup (Section 5.2.1). Next, we compare our method with previous works [72, 3, 123] on the seven core datasets of

the BOP challenge [51] (Section 5.2.2). We conduct this comparison to evaluate our method's accuracy, runtime performance, and robustness to segmentation errors, highlighting our contributions. Finally, we present an ablation study that explores different settings of our method (Section 5.2.3).

5.2.1 Experimental setup

Evaluation Datasets. We evaluate our method on the seven core datasets of the BOP challenge [51]: Occluded-Linemod (LM-O) [8], T-LESS [54], TUD-L [50], IC-BIN [27], ITODD [30], HomebrewedDB (HB) [64] and YCB-Video (YCB-V) [150]. These datasets consist of a total of 132 different objects and 19 048 testing instances, presented in cluttered scenes under partial occlusions. It is worth noting that, in contrast to the *seen* object setting, the *novel* object pose estimation setting is far from being saturated in terms of both accuracy and run-time.

Evaluation metrics. For all experiments, we use the standard BOP evaluation protocol [56], which relies on three metrics: Visible Surface Discrepancy (VSD), Maximum Symmetry-Aware Surface Distance (MSSD), and Maximum Symmetry-Aware Projection Distance (MSPD). The final score, referred to as the average recall (AR), is calculated by averaging the individual average recall scores of these three metrics across a range of error thresholds.

Baselines. We compare our method with MegaPose [72], ZS6D [3], and OSOP [123]. As of the time of writing, the source codes for ZS6D and OSOP are not available. Therefore, we can only report their performance as provided in their papers, but not their run-time.

Refinement. To demonstrate the potential of GigaPose, we have applied the refinement methods from MegaPose [72] and GenFlow [97] to our results. We extract the top-1 and the top-5 pose candidates and subsequently refine them using 5 iterations of MegaPose's refinement network [72] or GenFlow's refinement network [97]. For the top-5 hypotheses case, these refined hypotheses are scored by the coarse network of MegaPose [72], and the best one is selected.

Pose estimation with a 3D model predicted from a single image. We use Wonder3D [84] to predict a 3D model from a single image for objects from LM-O. We then evaluate the performance of MegaPose and our method using reconstructed models instead of the accurate CAD models provided by the dataset. Due to the sensitivity of Wonder3D to the quality of input images, we carefully select reference images.

Implementation details. We use the input image of size 224 $\times$224, resulting in features of size 16$\times$16$\times$1024 and 16$\times$16$\times$256 via the networks $\mathbf{F}_{\text{ae}}$ and $\mathbf{F}_{\text{ist}}$ respectively. We train our networks using the Adam optimizer with an initial learning rate of 1e-5 for $\mathbf{F}_{\text{ae}}$ and 1e-3 for $\mathbf{F}_{\text{ist}}$. The training process takes less than 10 hours when using four V100 GPUs. All the inference experiments are run on a single V100 GPU.

	Method	Detections	Refinement	N. instances:	LM-O 1445	T-LESS 6423	TUD-L 600	IC-BIN 1786	ITODD 3041	HB 1630	YCB-V 4123	MEAN	RUN-TIME
1	OSOP [123]	OSOP [123]	–		27.4	40.3	–	–	–	–	29.6	–	–
2	MegaPose [72]	Mask R-CNN [44]	–		18.7	19.7	20.5	15.3	8.00	18.6	13.9	16.2	–
3	ZS6D [3]	CNOS [102]	–		29.8	21.0	–	–	–	–	32.4	–	–
4	MegaPose [72]	CNOS [102]	–		22.9	17.7	25.8	15.2	10.8	25.1	28.1	20.8	15.5 s
5	GigaPose	CNOS [102]	–		29.6	26.4	30.0	22.3	17.5	34.1	27.8	26.8	0.4 s
6	MegaPose [72]	CNOS [102]	MegaPose [72]		49.9	47.7	65.3	36.7	31.5	65.4	60.1	50.9	17.0 s
7	GigaPose	CNOS [102]	MegaPose [72]		55.7	54.1	58.0	45.0	37.6	69.3	63.2	54.7	2.3 s
8	MegaPose [72]	CNOS [102]	MegaPose + 5 Hypo. [72]		56.0	50.7	68.4	41.4	33.8	70.4	62.1	54.7	21.9 s
9	GigaPose	CNOS [102]	MegaPose + 5 Hypo. [72]		59.8	56.5	63.1	47.3	39.7	72.2	66.1	57.8	7.7 s
10	MegaPose [72]	CNOS [102]	GenFlow + 5 Hypo. [97]		56.3	52.3	68.4	45.3	39.5	73.9	63.3	57.0	20.8 s
11	GigaPose	CNOS [102]	GenFlow + 5 Hypo. [97]		63.1	58.2	66.4	49.8	45.3	75.6	65.2	60.5	10.6 s

Table 5.1: **Results on the BOP datasets.** We report the AR score on each of the seven core datasets of the BOP challenge and the mean score across datasets. The best results with CNOS's detections [102] without refinement are highlighted in blue, with MegaPose's refinement using 1 hypothesis in yellow, and using 5 hypotheses in orange, and with GenFlow's refinement using 5 hypotheses in red.

5.2.2 Comparison with the state of the art

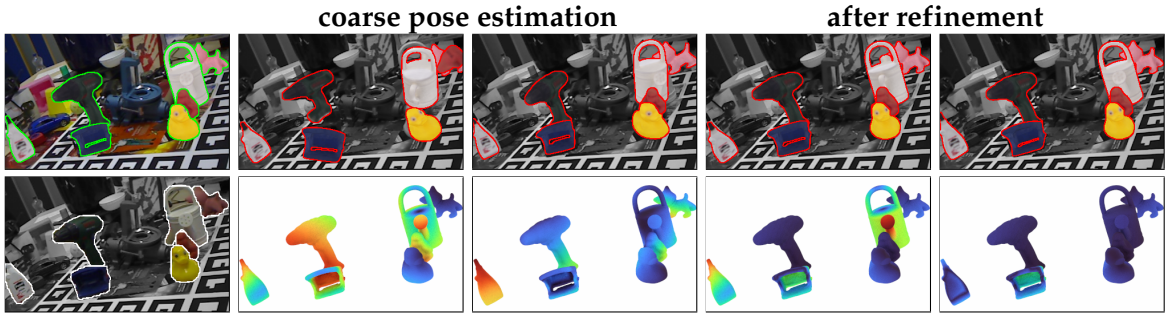


Figure 5.4: **Qualitative results on LM-O [8].** The first column shows the ground-truth and CNOS [102] segmentation. The second and third columns show the results without refinement for both MegaPose [72] and our method, including depth error heatmaps at the bottom. The last two columns compare the results using the same refinement [72] for MegaPose [72] and our method. In the error heatmap, darker red indicates higher error with respect to the ground truth pose (legend: 0 cm 10 cm). As demonstrated in this figure, our method estimates a more accurate coarse pose and avoids local minima during refinement, such as with the white "watering_can" object from LM-O.

Accuracy. Table 5.1 compares the results of our method with those of previous work [72, 3, 123]. Across all settings, whether with or without refinement, our method consistently outperforms MegaPose while maintaining significantly faster processing times. Notably, our method significantly improves accuracy on the challenging T-LESS, IC-BIN, and ITODD, with more than a 5% increase in AR score for coarse pose estimation and more than a 4% increase in AR score after refinement compared to MegaPose.

It is important to note that although the coarse and refinement networks in MegaPose [72] are not trained together, they were trained to *work* together: As mentioned in Section 3.2 of [72], the positive samples of the coarse network *"are*

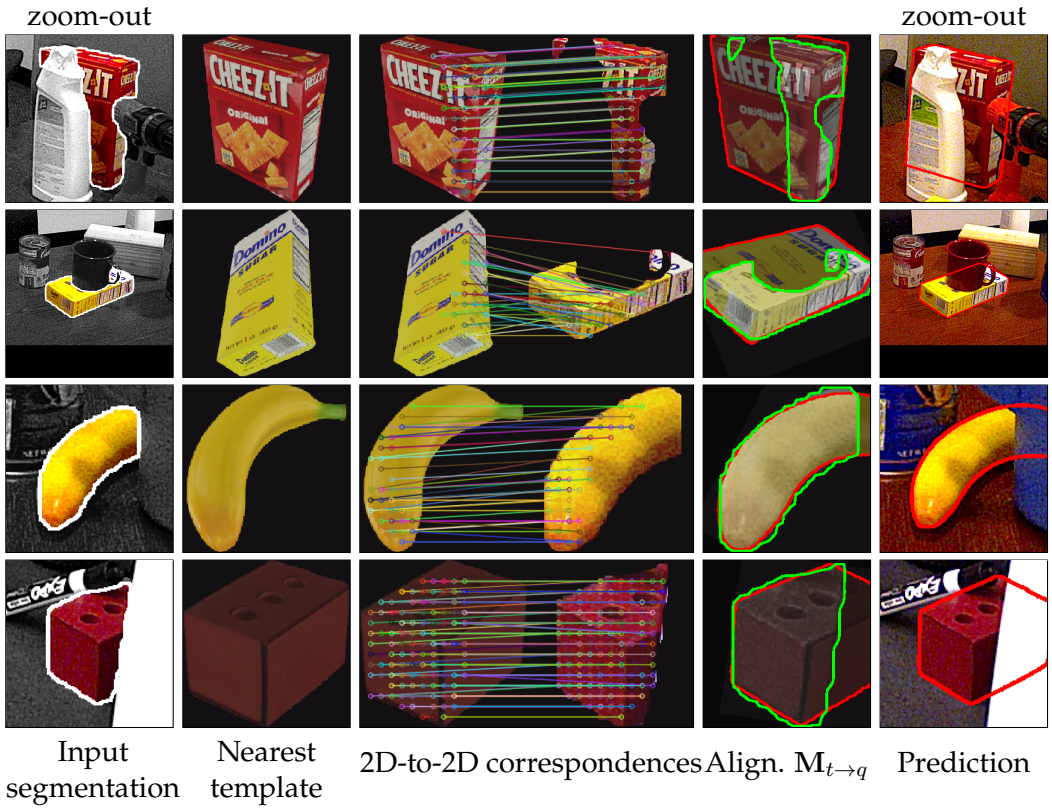


Figure 5.5: **Qualitative results on YCB-V [150].** The first column shows CNOS [102]'s segmentation. The second and third columns illustrate the outputs of the nearest neighbor search step, which includes the nearest template ($\mathbf{R}_{\text{ae}}$) and the 2D-to-2D correspondences. The fourth column demonstrates the alignment achieved by applying the predicted affine transform $\mathbf{M}_{t\to q}$ to the template, then overlaying it on the query input: The green contour indicates the noisy segmentation by CNOS [102], while the red contour highlights the boundary of the aligned template. The last column show the final prediction after refinement [72].

sampled from the same distribution used to generate the perturbed poses the refiner network is trained to correct". This pose sampling biases MegaPose's refinement process towards MegaPose's coarse estimation errors. This explains why the refinement process brings larger improvements to MegaPose than GigaPose, in particular on TUD-L where the refinement improves MegaPose by 39.5% and our method only by 28.0%. However, TUD-L represents only about 3% of the total test data, our method still outperforms MegaPose over the 7 datasets in all settings.

Figure 5.4 shows qualitative comparisons with MegaPose [72] before and after refinement on LM-O dataset [46] showcasing our more accurate pose estimates. Figure 5.5 shows qualitative results of each step of GigaPose on YCB-V dataset [150].

Accuracy when using predicted 3D models. As shown in Table 5.2, our method outperforms MegaPose when using predicted 3D models. Results in Table 5.2 implies that when no CAD model is available for an object, we can use Wonder3D to predict a 3D model from a single image, then apply GigaPose and MegaPose refinement. These results are close to GigaPose's performance and surpass MegaPose's coarse performance when using an accurate CAD model.

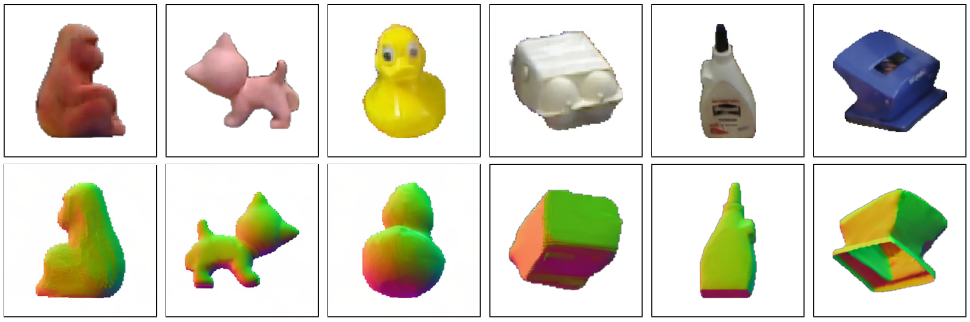


Figure 5.6: **3D reconstruction by Wonder3D [84].** The first row displays the input reference image, the second shows the predicted normal maps from the view opposite to the reference image.

	Method	Detection [102]	Single image		GT 3D model
			Coarse	Refined	w/o refinement
1	MegaPose [72]	GT 3D model	16.3	25.6	22.9
2	GigaPose (ours)	GT 3D model	**18.3**	**27.9**	**29.6**
3	MegaPose [72]	Single image	15.4	25.2	22.7
4	GigaPose (ours)	Single image	**17.6**	**27.2**	**29.4**

Table 5.2: **Results with predicted 3D models on LM-O [8].** We report AR score using 3D models predicted from a single reference image by Wonder3D [84]. The 3D reconstruction is shown in Figure 5.6. Rows 3 and 4 display additional results for MegaPose and our method, where CNOS [102] is also given 3D predicted models.

Run-time. We report the speed of GigaPose in Table 5.1 (rightmost column) following the BOP evaluation protocol. It measures the total processing time *per image* averaged over the datasets including the time taken by CNOS [102] to segment each object, the time to estimate the object pose for all detections, and the refinement time if applicable.

Table 5.3 gives a breakdown of the run-time *per detection* for each stage of MegaPose and of our method. Our method takes only 48 ms for coarse pose estimation, more than 38x faster than the 1.68 seconds taken by MegaPose. This improvement can be attributed to our sublinear nearest neighbor search, significantly faster than feed-forwarding each of the 576 input-template pairs as done in MegaPose.

Robustness to segmentation errors. To demonstrate the robustness of our method, we analyze its performance under various levels of segmentation errors on three standard datasets: LM-O [8], T-LESS [54], and YCB-V [150]. We use the ground-truth masks to classify the segmentation errors produced by CNOS's segmentation [102] using the Intersection over Union (IoU) metric. For each IoU threshold, we retain only the input masks from CNOS that matched the ground-truth masks with an IoU smaller than this threshold and evaluate the AR score for coarse pose estimation.

As shown in Figure 5.7, our method has a stable AR score across all IoU thresholds for both T-LESS and YCB-V, in contrast with MegaPose, which yields high scores primarily for high IoU thresholds only.

Method	Run-time		
	Onboarding	Coarse pose	Refinement [72]
MegaPose [72]	0.82 s	1.68 s	33 ms
GigaPose (ours)	11.5 s	48 ms	33 ms

Table 5.3: **Run-time.** Breakdown of the average run-time for each stage of MegaPose [72] and our method on a single V100 GPU to estimate the pose per object (i.e., per detection). Our method is more than 35× faster than MegaPose for coarse pose estimation.

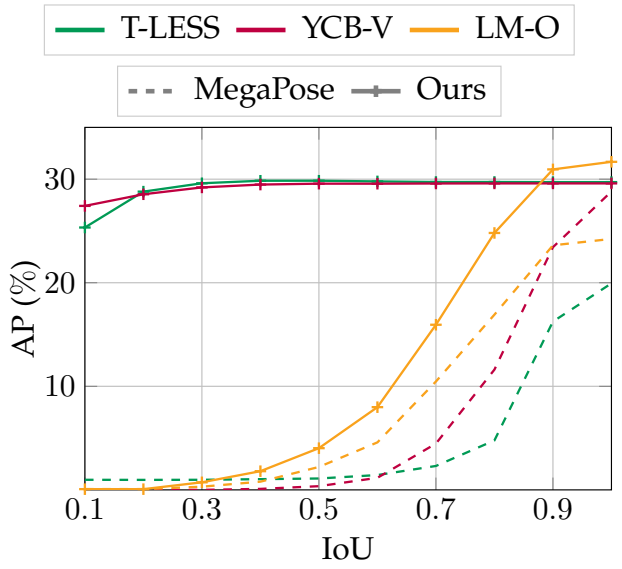


Figure 5.7: **Robustness to segmentation errors.** We analyze the performance of MegaPose and our method under various levels of segmentation errors, defined by the IoU between the predicted masks from CNOS [102] and the ground-truth masks. Our method demonstrates much higher stability in AP across all IoU thresholds than MegaPose, showing its robustness against segmentation errors. The improvement is more limited on LM-O because of the small appearance size of the objects especially after occlusions.

5.2.3 Ablation study

In Table 5.4, we present several ablation evaluations on the three standard datasets LM-O [8], T-LESS [54], and YCB-V [150]. Our results are in Row 5.

Fine-tuning $\mathbf{F}_{ae}$. Row 1 of Table 5.4 presents the results of using the DINOv2 features [106] without fine-tuning $\mathbf{F}_{ae}$. As shown in Row 5, fine-tuning significantly improves template-correspondences, leading to a 8.9% increase in AR score.

Estimating in-plane rotation with templates. Row 2 of Table 5.4 shows in-plane rotation estimation results using templates by dividing in-plane angle into 36 bins of 10 degrees, yielding 5832 templates per object. This approach decreases the AR score by 5.8% compared to direct predictions with $\mathbf{F}_{ist}$ and $\mathbf{H}$ in Row 5, underscoring the effectiveness of our hybrid template-patch correspondence approach.

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

Table 5.4: **Ablation study.** We report the AR score of different settings of our method including: without fine-tuning $\mathbf{F}_{ae}$ in Row 1, estimating in-plane rotation with dense 3DoF templates in Row 2, different "PnP" variants in Rows 3 and 4. The results of the complete method are on Row 5. We show in Row 6 our results using the same 576 templates as in MegaPose [72]. See Section 5.2.3.

2D-to-2D vs 3D-to-2D correspondences. In Row 3, we introduce a "3D-to-2D correspondence" variant by replacing the 2D locations of the matched patches in the template with their 3D counterparts obtained from the template depth map. We then estimate the complete 6D object pose using the ePnP algorithm [75] implemented in OpenCV [9]. Furthermore, in Row 4, we present a two-"2D-to-2D correspondences" variant, where the scale and in-plane rotation are computed using a 2D variant of the Kabsch algorithm. Our single-correspondence approach in Row 5 is more effective at exploiting patch correspondences for estimating scale and in-plane rotation directly.

Number of templates. In Row 6, we present our results using the same 576 templates as MegaPose [72]. This improves by only 0.6% the AR score compared to using 162 templates (Row 5). This confirms that the correspondences also allows to decrease the memory footprint of the templates without hurting the accuracy.

5.3 Conclusion

As discussed in Section 5.2.2, GigaPose fails on challenging conditions where heavy occlusions, low-resolution segmentation, and low-fidelity CAD models are present, as observed in the LM-O dataset. To address this issue, incorporating additional modalities, such as depth images, can be beneficial. Depth images, which capture information about object geometries, can significantly enhance model performance under these conditions. Moreover, although our method shows promising results in a single-reference setting using Wonder3D [84], it requires manual selection to achieve high-quality 3D reconstruction. Therefore, the development of more advanced 3D reconstruction techniques capable of generating high-quality outputs from a single image would be particularly valuable in this context.

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

Chapter 6

NOPE: Novel Object Pose Estimation from a Single Image

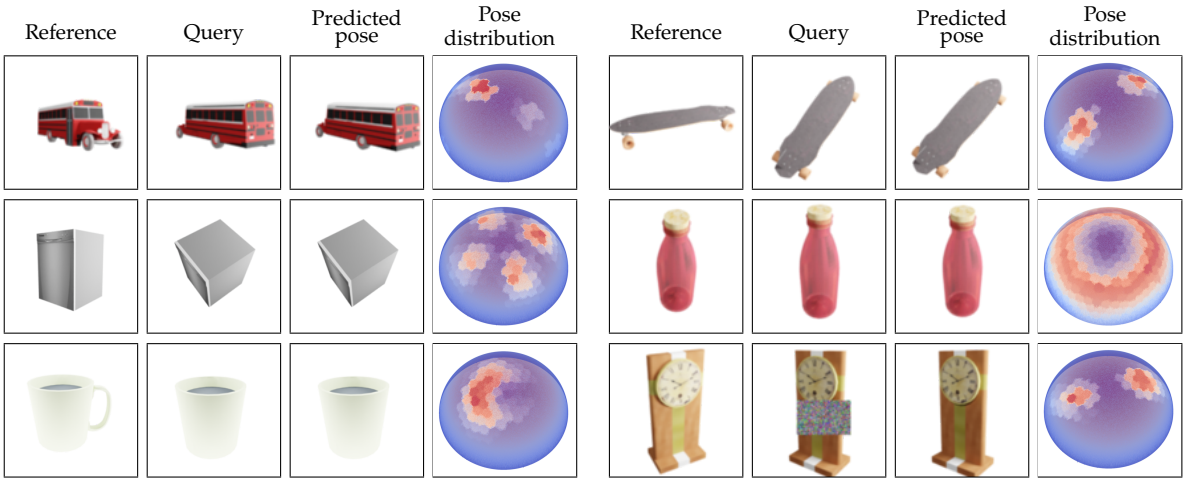


Figure 6.1: Given as input a single reference view of a novel object, our method predicts the relative 3D pose (rotation) of a query view and its ambiguities. **We visualize the predicted pose by rendering the object from this pose, but the 3D model is only used for visualization purposes, not as input to our method.** Our method works by estimating a probability distribution over the space of 3D poses, visualized here on a sphere centered on the object. **We use the canonical pose of the 3D model to visualize this distribution, but not as input to our method.** From this distribution, we can also identify the pose ambiguities: For example, in the case of the bottle, any pose with the same pitch and roll is possible; in the case of the mug, a range of poses are possible as the handle is not visible in the query image. Our method is also robust to partial occlusions, as shown on the clock hidden in part by a rectangle in the query image.

The work presented in this chapter was initially presented in:

[101] **Van Nguyen Nguyen**, Thibault Groueix, Georgy Ponimatkin, Yinlin Hu, Renaud Marlet, Mathieu Salzmann, Vincent Lepetit. *NOPE: Novel Object Pose Estimation from a Single Image*. Computer Vision and Pattern Recognition (CVPR), 2024.

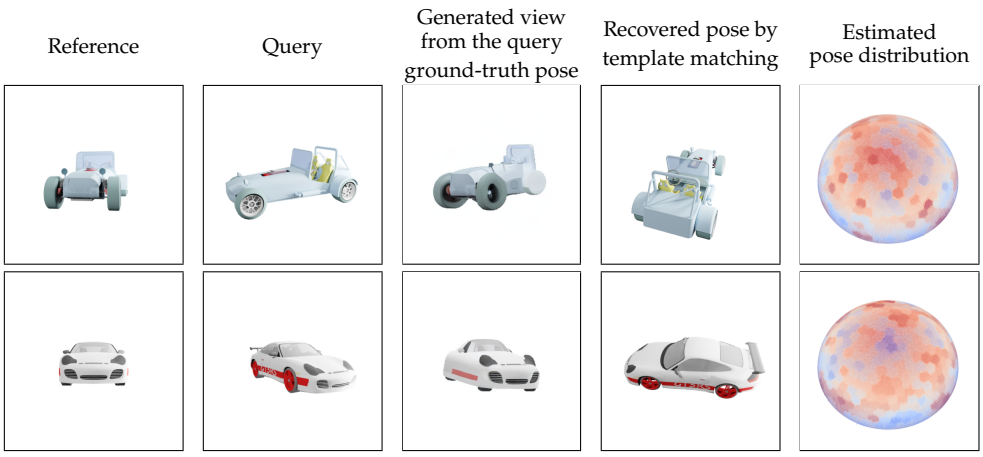


Figure 6.2: **The limit of novel view synthesis for pose prediction.** While the images generated by Wonder3D [84] look very realistic, they have to invent unseen parts, impairing the similarity computation between the query image and the generated view, and hence the pose estimation: The probability distributions computed by template matching do not peak on the right pose but show many wrong local maxima. This is not a limitation of Wonder3D but of view synthesis from a single view in general.

In this chapter, we introduce an approach, which we call **NOPE** for **N**ovel **O**bject **P**ose **E**stimation, that only requires a single image of the new object to predict the relative pose of this object in any new images, without the need for the object's 3D model and without training on the new object. This is a very challenging task, as, by contrast with the multiple views used in [126, 158] for example, a single view only provides limited information about the object's geometry.

To achieve this, we train NOPE to predict the appearance of the object under novel views. We use these predictions as 'templates' annotated with the corresponding poses. Matching these templates with new input views lets us estimate the object relative pose with respect to the initial view. This approach is motivated by the good performance of recent related work [104, 123]. In particular, [104] showed that template matching can be extremely fast and robust to partial occlusions. This contrasts with methods that rely on a deep network to predict the probability of a pose [158].

Since our method relies on predicting the appearance of the target object, it relates to recent developments in novel view synthesis. However, it has two critical differences: The first difference is that instead of predicting color images, we directly predict discriminative embeddings of the views. These embeddings are extracted by passing the input image through a U-Net architecture with attention and conditioned on the desired pose for the new view.

The second main difference of our approach with novel view synthesis is more fundamental. We first note that generating novel views given a single view of an object is ambiguous. Novel view synthesis usually focuses on generating a single possible image for a given point of view. This is however not suitable for our purpose: The view synthesis method will "invent" the parts that were not visible in the input view. As illustrated in Figure 6.2, these invented parts create a plausible novel view but there is no guarantee this view actually corresponds to the actual view. For our goal of pose estimation, the invented parts will not match in general the query

view and this will result in incorrect pose estimation. The limitations of using novel view synthesis for pose estimation will further be quantitatively demonstrated in our experiments (see Table 6.1).

Our approach to handling the ambiguities in novel view synthesis for template matching is to consider the *distribution* of all the possible appearances of the object for the target viewpoint. More exactly, we train NOPE to predict the average of all the possible appearances of the object. We then treat the predicted average as a template: Under some simple assumptions, the distance between this template and the query view is directly related to the probability of the query view to be a sample from the distribution of the possible appearances of the object. This approach allows us to deal with the ambiguities of novel view prediction in a robust and efficient way: Predicting the average views is just a direct inference of NOPE and is thus very fast, and robust to partial occlusions thank to template-matching.

Furthermore, our approach can identify the pose ambiguities due, for example, to symmetries [88], even if we do not have access to the object 3D model but only to a single view. To this end, we estimate the distribution over all poses for the query, which becomes increasingly less peaked as the pose suffers from increasingly many ambiguities. Figure 6.1 depicts a variety of ambiguous and unambiguous cases with their pose distributions.

In summary, our main contribution in this chapter is to show we can efficiently and reliably recover the relative pose of an unseen object in novel views given only a single view of that object as reference. To the best of our knowledge, our approach is the first to predict ambiguities due to symmetries and partial occlusions of unseen objects from only a single view.

6.1 Novel view synthesis

One very simple way to relax the requirement of the 3D CAD model is using novel view synthesis methods to generate different viewpoints from reference images. This relates to the pioneering work of Nerfs [95] since it performs novel-view synthesis. Recent works [167, 93] have had successes generating novel views via Nerfs using a sparse set of views as input by leveraging 2D diffusion models. For images, the breakthrough in diffusion models [49, 124] have unlocked several workflows [117, 121, 120]. For 3D applications, DreamFusion [112] pioneered a score-distillation sampling that allows for the use of a 2D diffusion model as an image-based loss, leveraged by 3D applications via differentiable rendering. This has resulted in significant improvements for tasks previously trained with a CLIP-based image loss [116, 61, 67, 79, 138, 62] or diffusion-based methods [84, 82, 80, 144].

In this chapter, while our main goal is to perform object pose estimation for novel objects, we will demonstrate how recent advancements in novel view synthesis methods can be used to relax the requirement of 3D CAD models, enabling pose estimation from a single reference image.

6.2 Method

In this section, we first introduce our formalism, then describe our architecture and how we train it, and finally how we use it for pose prediction and for identifying

pose ambiguities.

6.2.1 Formalization

Given a reference image or a template $\mathcal{T}$ of a target object and a query image $\mathcal{Q}$ of the same object, we would like to estimate the probability $p(\Delta R \mid \mathcal{T}, \mathcal{Q})$ that the relative motion between $\mathcal{T}$ and $\mathcal{Q}$ is a certain discretized relative pose ΔR. We assume that this probability follows a normal distribution in the embedding space of the images:

$$p(\Delta R \mid \mathcal{T}, \mathcal{Q}) = \mathcal{N}(\mathbf{q} \mid \mu(\mathbf{t}, \Delta R), \Sigma(\mathbf{t}, \Delta R)) \,, \tag{6.1}$$

where $\mathbf{q}$ and $\mathbf{t}$ are the embeddings for query image $\mathcal{Q}$ and reference image $\mathcal{T}$ respectively, $\mu(\mathbf{t}, \Delta R)$ is the mean of the normal distribution, and $\Sigma(\mathbf{t}, \Delta R)$ its covariance. This approach allows us to handle the fact that the object can have various appearances from viewpoint ΔR given the reference image, as discussed in the introduction.

We take the mean $\mu(\mathbf{t}, \Delta R)$ as the average embedding for the appearance of the object from pose ΔR over the possible 3D shapes for the object:

$$\mu(\mathbf{t}, \Delta R) = \int_{\mathcal{M}} \mu(\Delta R, \mathcal{M}) p(\mathcal{M}|\mathbf{t}) d\mathcal{M} \,, \tag{6.2}$$

with $\mathcal{M}$ a 3D model of testing object and $\mathbf{e}(\Delta R, \mathcal{M})$ the image embedding of same object under pose ΔR. $\mu(\mathbf{t}, \Delta R)$ may look complicated to compute, but it is in fact easy to train a deep network to predict it using the L2 loss:

$$\sum_{(\mathbf{q}, \mathbf{t}, \Delta R)} \|F(\mathbf{t}, \Delta R) - \mathbf{q}\|_2^2 \,. \tag{6.3}$$

F denotes the network, $(\mathbf{q}, \mathbf{t}, \Delta R)$ is a training sample where we know the ground-truth. During training, given enough samples, $F(\mathbf{t}, \Delta R)$ will converge naturally towards $\mu(\mathbf{t}, \Delta R)$.

6.2.2 Framework

Figure 6.3 gives an overview of our approach. We train a deep architecture to predict the average embeddings of novel views of an object using pairs of images of objects and the corresponding pose changes from a first set of object categories. In practice, we consider embeddings computed from the pretrained VAE of [119], as it was shown to be robust for template matching. To generate these embeddings, we use a U-Net-like network with a pose conditioning mechanism that is very close to the one of 3DiM [144].

More precisely, we first use an MLP to convert the desired relative viewpoint ΔR with respect to the object pose in the reference view to a pose embedding. We then integrate this pose embedding into the feature map at every stage of our U-Net using cross-attention, as in [119].

Training. At each iteration, we build a batch composed of N pairs of images, a reference image and another image of the same object with a known relative pose. The U-Net model takes as input the embedding of the reference image and as conditioning

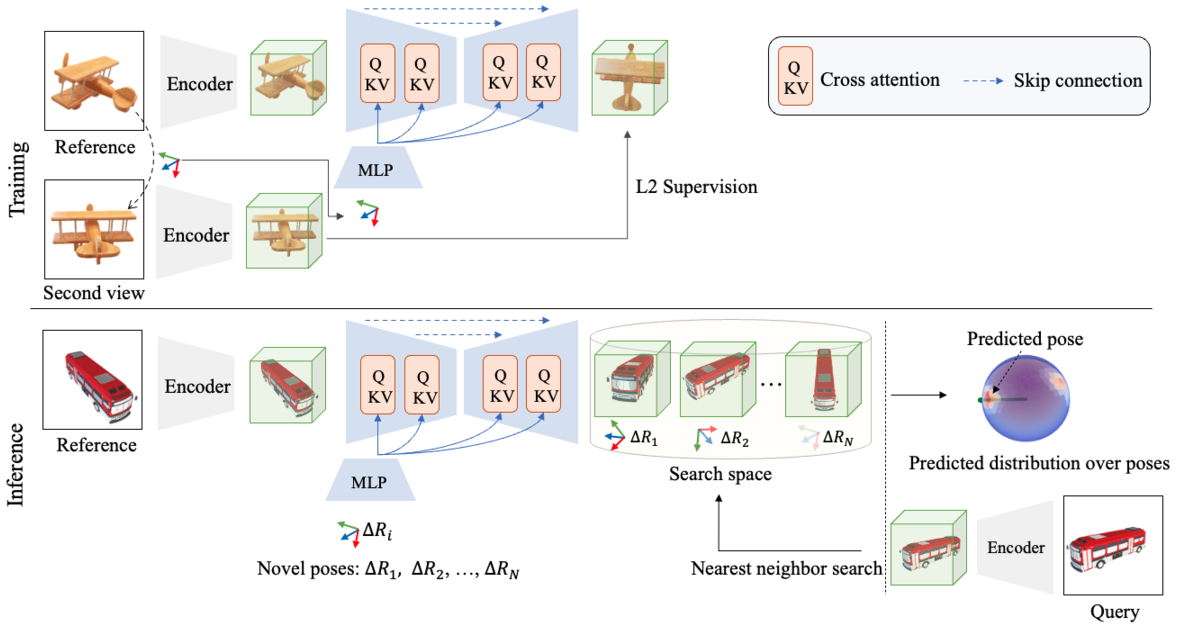


Figure 6.3: **Overview.** During training, we train a U-Net to predict the embedding of a novel view of an object, given a reference image of the object and a relative pose. The U-Net is conditioned on an embedding of the relative pose computed using an MLP, which we train jointly with the U-Net. At inference, our method first takes as input a reference image of a new object and predicts the embeddings of views of the object under many relative poses. This inference takes around 1 second on a single GPU V100. Then, given a query image of the object, we first compute its embedding and match it against the set of predicted embeddings. This gives us a distribution over the possible relative poses between the reference and query images, where the maximum corresponds to the predicted pose.

the embedding of the relative pose to predict an embedding for the second image. We jointly optimize the U-Net and the MLP by minimizing the Euclidean distance between this predicted embedding and the embedding of the query image. Note that we freeze the pretrained VAE network of [119] during the training.

By training it on a dataset of diverse objects, this architecture generalizes well to novel unseen object categories. Interestingly, our method does not explicitly learn any symmetries during training, but it is able to detect pose ambiguities during testing as discussed below.

6.2.3 Pose prediction

Template matching. Once our architecture is trained, we can use it to generate the embeddings for novel views: Given a reference image and a set of N relative viewpoints $\mathcal{P} = (\Delta R_1, \Delta R_2, \ldots, \Delta R_N)$, we can obtain a corresponding set of predicted embeddings $(\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_N)$. To define these viewpoints, we follow the approach used in [104]: We start with a regular isosphere and subdivide each triangle recursively into four smaller triangles twice to get 342 final viewpoints. Finally, we simply perform a nearest neighbor search to determine the reference point that has the embedding closest to the embedding of the query image.

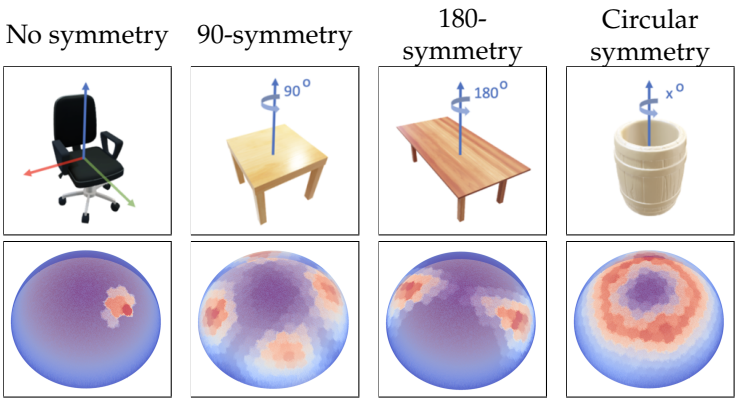
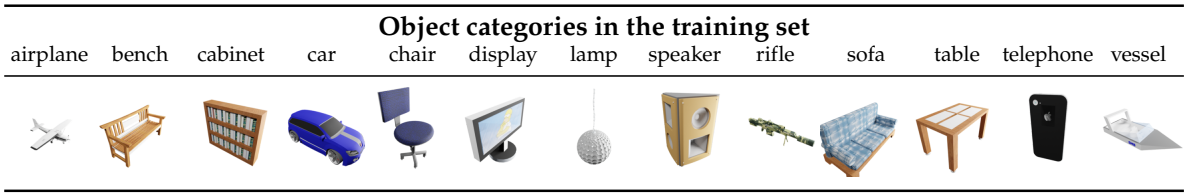


Figure 6.4: **Object symmetries and the pose ambiguities they may generate**, as estimated by our method given a pair of reference and query images.



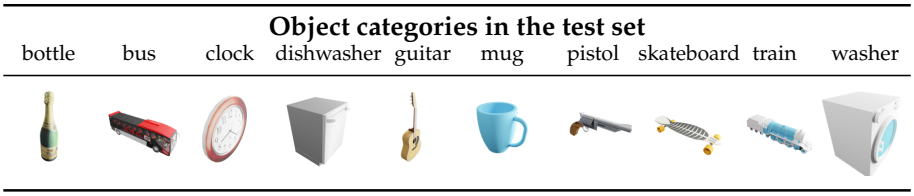


Figure 6.5: **Visualization of training and test sets from the ShapeNet dataset [16].** The shapes and appearances in the training and test sets are very different and thus constitute a good test bed for generalization to unseen categories.

Detecting pose ambiguities. Pose ambiguities arise when the object has symmetries or when an object part that could remove the ambiguity is not visible, as for the mug in Figure 6.1. By considering the distance between the embedding of the query image and the generated embeddings, we not only can predict a single pose but also identify all the other poses that are possible given the reference and query views. This can be done simply by relying on the normal distribution introduced in Equation (6.1):

$$\log p(\Delta R \mid \mathcal{T}, \mathcal{Q}) \propto \|F(\mathbf{t}, \Delta R) - \mathbf{q}\|^2 . \tag{6.4}$$

To illustrate this, we show in Figure 6.4 three distinct types of symmetry and visualize the pose distribution for corresponding pairs of reference and query images (not shown). The number of regions with high similarity scores is consistent with the number of symmetries and pose ambiguities: If an object has no symmetry, the probability distribution has a clear mode. The probability distribution for objects with symmetries have typically several modes or even a continuous high-probability region in case of rotational symmetry. We provide additional qualitative results in Section 6.3.

6.3 Experiments

In this section, we first describe our experimental setup in Section 6.3.1. We then compare our method to others [100, 91, 90, 144, 127, 104] on both synthetic and real-world datasets in Section 6.3.2. Section 6.3.3 reports an evaluation of the robustness to partial occlusions. We provide the run-time in Section 6.3.4. Finally, we discuss failure cases in Section 6.3.5.

6.3.1 Experimental setup

To the best of our knowledge, we are the first method addressing the problem of object pose estimation from a single image when the object belongs to a category not seen during training: PIZZA [100] evaluated on the DeepIM refinement benchmark, which is made of pairs of images with a small relative pose; SSVE [91] and ViewNet [90] evaluated only on objects from categories seen during training. We therefore had to create a new benchmark to evaluate our method.

Synthetic dataset. We created a dataset as in FORGE [63] using the same ShapeNet [16] object categories. For the training set, we randomly select 1000 object instances from each of the 13 categories as done in FORGE (*airplane, bench, cabinet, car, chair, display, lamp, loudspeaker, rifle, sofa, table, telephone, and vessel*), resulting in a total of 13,000 instances. We build two separate test sets for evaluation. The first test set is the "novel instances" set, which contains 50 new instances for each training category. The second test set is the "novel category" set, which includes 100 models per category for the 10 unseen categories selected by FORGE (*bus, guitar, clock, bottle, train, mug, washer, skateboard, dishwasher, and pistol*). For each 3D model, we randomly select camera poses to produce five reference images and five query images. We use BlenderProc [26] as rendering engine.

Figure 6.5 illustrates the categories used for training our architecture and the categories used for testing it. The shapes and appearances of the categories in the test set are very different from the shapes and appearances of the categories in the training set, and thus constitute a good test set for generalization to unseen categories.

Real-world dataset. We evaluate on the T-LESS dataset [54] following the evaluation protocol of [127]: we train only on objects 1-18 and test on the full PrimeSense test set using the ground-truth masks. At inference, we randomly sample a non-occluded reference image either from *all views* or only from *front views* (-45°$\leq$ azimuth $\leq$ 45°), which often offers more information on the object and illustrates the influence of the reference view.

Metrics. For the ShapeNet dataset, we report two different metrics based on relative camera pose error as done in [91]. Specifically, we provide the median pose error across instances for each category in the test set, and the accuracy Acc 30 for which a prediction is treated as correct when the pose error is $\leq 30°$. Additionally, we present the results of our method for the top 3 and 5 nearest neighbors retrieved by template matching.

For the T-LESS dataset, as most objects are symmetric, we report the recall VSD metric as done in [127]. Please note that for the evaluation on the T-LESS dataset,

	Method	novel inst.	bottle*	bus	clock	dishwasher	guitar	mug	pistol	skateboard	train	washer	mean
Acc 30 ↑	ViewNet [90]	**77.5**	48.4	36.2	23.5	16.4	37.8	31.3	17.9	33.9	44.8	25.1	35.7
	SSVE [91]	75.3	61.5	38.2	41.8	21.3	46.8	38.4	36.8	**62.3**	41.5	50.8	46.8
	PIZZA [100]	72.3	76.0	38.6	38.5	32.6	30.8	35.6	40.4	58.3	52.9	**61.0**	48.8
	3DiM [144]	77.3	95.1	43.5	23.6	24.5	36.0	32.0	31.9	50.3	37.0	56.1	46.1
	Ours (top 1)	75.5	**96.0**	**53.6**	**48.0**	**48.0**	**49.0**	**44.6**	**69.0**	57.8	**55.2**	60.6	**59.8**
	Ours (top 3)	92.0	97.4	83.8	73.4	78.5	66.8	56.0	83.8	86.2	86.0	84.4	80.8
	Ours (top 5)	95.5	97.8	89.8	80.4	88.2	74.6	62.8	88.4	92.8	95.4	93.4	87.1
Median ↓	ViewNet [90]	6.6	26.7	35.8	40.3	96.3	50.6	51.6	42.8	37.4	26.8	44.3	41.7
	SSVE [91]	6.1	23.8	45.2	41.9	90.4	47.6	49.6	24.0	13.5	24.9	48.1	37.7
	PIZZA [100]	5.8	25.5	26.4	43.2	80.6	40.2	45.5	23.4	17.3	20.3	38.5	33.3
	3DiM [144]	**5.7**	**1.8**	19.8	47.3	98.8	35.2	35.7	21.2	**12.5**	**17.6**	19.2	28.6
	Ours (top 1)	8.1	**1.8**	**18.4**	**39.9**	**77.6**	**31.6**	**35.5**	**13.4**	15.5	18.3	**8.5**	**24.4**
	Ours (top 3)	5.0	1.3	5.8	9.1	4.8	16.0	22.6	8.1	6.5	6.7	5.7	8.3
	Ours (top 5)	4.5	1.2	4.5	7.1	4.4	11.6	18.4	6.1	5.6	4.9	5.0	6.6

Table 6.1: **Quantitative results on ShapeNet.** *We treat “bottle" as a symmetric category, i.e., the error is only the difference of elevation angle. Since the quality of prediction may depend on the reference image, we report the score as the average over 5 runs with 5 different reference images.

we also predict the translation by using the same formula “projective distance estimation” as SSD-6D [65], as done in [128, 127]. This translation is deduced from the retrieved template and the relative scale factor between the two input images as done in [104].

Baselines. We compare our work with all previous methods that aim to predict a pose from a single view: PIZZA [100], a regression-based approach that directly predicts the relative pose, as well as SSVE [91] and ViewNet [90], which employ semi-supervised and self-supervised techniques to treat viewpoint estimation as an image reconstruction problem using conditional generation. We also compare our method with the recent diffusion-based method 3DiM [144], which generates pixel-level view synthesis. Since 3DiM originally only targets view-synthesis and is not designed for 3D object pose, we use it to generate templates and perform nearest neighbor search to estimate a 3D object pose. To make 3DiM work in the same setting as us, we retrain it using relative pose conditioning instead of canonical pose conditioning.

Implementation. Only the code of PIZZA is available. The other methods did not release their code at the time of writing, however we re-implemented them. We use a ResNet18 backbone as in [100] for PIZZA, SSVE, and ViewNet. We train all models on input images with a resolution of 256×256 except for 3DiM for which we use a resolution of 128×128 since 3DiM performs view synthesis in pixel space, which takes much more memory. Our re-implementations achieve similar performance as the original papers when evaluated on the same data for seen categories, as shown in Table 6.1, which validates our comparisons. Our method also uses the frozen encoder from [119] to encode the input images into embeddings of size 32×32×8. In all settings, we train the baselines and our method using the same training set and AdamW [85] with an initial learning rate of 5×10^{-5}. Training takes about 20 hours on 4 V100 GPUs for each method.

	Method	Ref. image sampling	Recall VSD		
			Seen object	Novel object	Avg
GT CAD	Nguyen et al. [104]	-	**60.15**	**58.70**	**59.57**
	MultiPath [127]	-	43.17	43.33	43.24
1 ref. image (avg 5 runs)	PIZZA [100]	all views	20.05	15.90	18.39
	Ours	all views	**47.03**	**45.69**	**46.49**
	PIZZA [100]	front views	21.63	15.55	19.19
	Ours	front views	**49.30**	**48.46**	**48.96**

Table 6.2: **Comparison to PIZZA [100] and CAD-based methods [104, 127]** on seen (obj. 1-18) and novel (obj. 19-30) objects of T-LESS. We report numbers averaged over 5 different samplings and runs.

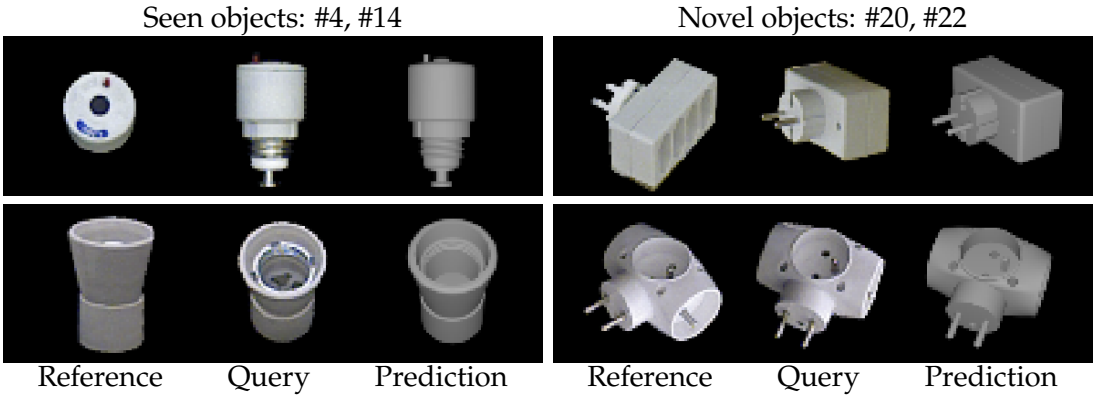


Figure 6.6: **Qualitative results on real images of T-LESS.** For each sample, we show in the last column the predicted poses.

6.3.2 Comparison with the state of the art

Results on ShapeNet. Table 6.1 summarizes the results of our method compared with the baselines discussed above. Under both the Acc30 and Median metrics, our method consistently achieves the best overall performance, outperforming the baselines by more than 10% in Acc30 and 10^o in Median. In particular, while other works produce reasonable results on unseen instances of seen training categories, they often struggle to estimate the 3D pose of objects from unseen categories. By contrast, our method works well in this case, demonstrating a better generalization ability on unseen categories.

Figure 6.7 shows some visualization results of our method on unseen categories, with and without symmetries. Our method produces more accurate 3D poses than the baselines when there is a symmetry axis.

Results on T-LESS. Table 6.2 shows our comparison with [100, 127, 104] on real images of T-LESS. While our method focuses on the more challenging case of using *a single reference image*, [104, 127] rely on ground-truth CAD models. Our method consistently outperforms the baseline PIZZA by a large margin. Interestingly, although there is still a gap compared to the SOTA [104], our method outperforms MultiPath [127]. Figure 6.6 shows results on seen and unseen objects of T-LESS.

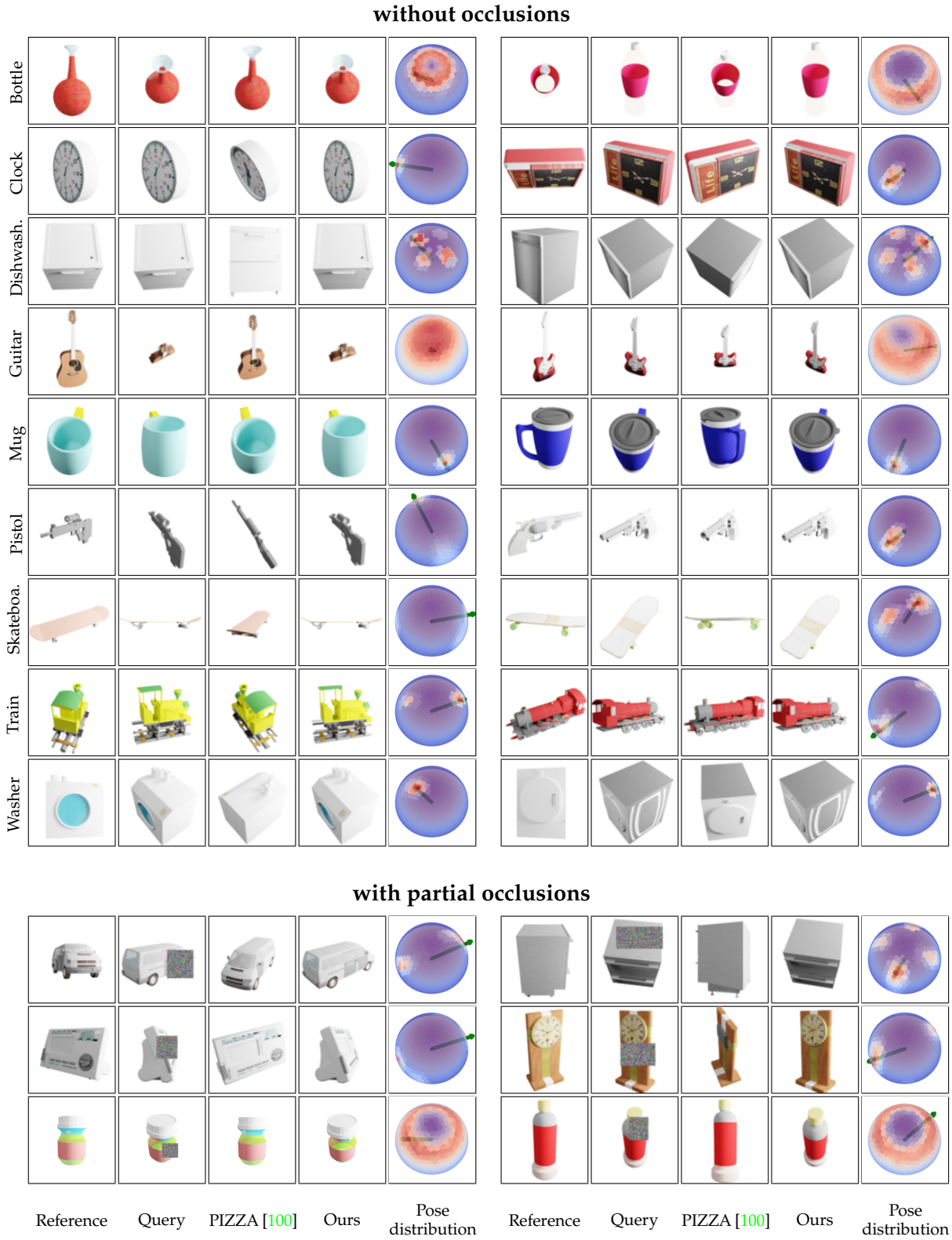


Figure 6.7: **Visual results on unseen categories** from ShapeNet. An arrow indicates the pose with the highest probability as recovered by our method. We visually compare with PIZZA, which is the method with the second best performance. **We visualize the predicted poses by rendering the object from these poses, but the 3D model is only used for visualization purposes, not as input to our method. Similarly, we use the canonical pose of the 3D model to visualize this distribution, but not as input to our method.**

Acc 30 ↑ Method	0%	5%	10%	15%	20%	25%
PIZZA [100]	48.9	44.6	33.3	24.5	18.2	14.6
NOPE (ours)	**59.8**	**54.3**	**48.4**	**45.1**	**43.7**	**40.5**

Table 6.3: **Robustness to partial occlusions.** We add rectangles of Gaussian noise to the query image, and vary the ratio between the area of the rectangle and the area of the object's 2D bounding box. Our method remains robust under large occlusions, while PIZZA's performance decreases significantly.

Method	Memory	Run-time	
		Processing	Neighbors search
3DiM [144]	358.6 MB	13 min	0.31 s
NOPE (ours)	22.4 MB	1.01 s	0.18 s

Table 6.4: **Average run-time** of our method and 3DiM [144] on a single GPU V100. We report the memory used for storing novel views, the time taken to generate novel views, and the time taken for nearest neighbor search to obtain the final prediction.

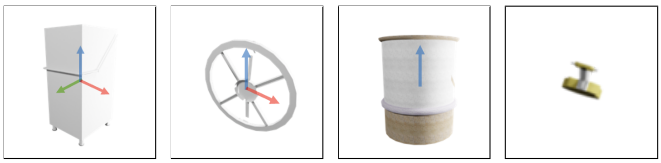


Figure 6.8: **Failure cases.** "Dishwasher", "clock", and "dishwasher" are "nearly symmetrical" while "guitar" are barely visible from some viewpoints. This makes the pose estimation very challenging, and all the methods perform poorly on these categories.

6.3.3 Robustness to occlusions

To evaluate the robustness of our method against occlusions, we added random rectangle filled with Gaussian noise to the query images over the objects, in a similar way to Random Erasing [163]. We vary the size of the rectangles to cover a range betwen 0% to 25% of the bounding box of the object. Figures 6.1 and 6.7 show several examples.

6.3.4 Runtime analysis

We report the running time of NOPE and 3DiM in Table 6.4. Our method is significantly faster than 3DiM, thanks to our strategy of predicting the embedding of novel viewpoints with a single step instead of multiple diffusion steps.

6.3.5 Failure cases

All the methods fail to yield accurate results when evaluated on "clock", "dishwasher", "guitar", and "mug" categories, as indicated by the high median errors. As shown in Figure 6.8, these categories except "guitar" are "almost symmetric", in the sense that only small details make the pose non-ambiguous. Our predictions

using the top-3 and top-5 nearest neighbors significantly improves median errors for 90-symmetrical, 180-symmetrical objects, but not circular-symmetrical as mug objects. Additionally, guitar objects can appear very thin under certain viewpoints.

6.4 Conclusion

Our experiments in this chapter have shown that direct inference of average view embeddings from a single view, as in NOPE, leads to accurate object pose estimation. This is true even for objects from unseen categories, while requiring neither retraining nor a 3D model. NOPE also lets us estimate the pose ambiguities that arise for many objects.

Chapter 7

Conclusion

7.1 Contributions

We presented methods for detecting, segmenting, and estimating the 6D pose of novel objects without requiring retraining. We focused on practical scenarios where the test objects are unknown, and only the 3D model or a single reference image is available for inference. Our proposed methods demonstrated comparable or superior performance to existing approaches, despite requiring less input data, and even achieved state-of-the-art results on standard benchmarks.

More precisely, we introduced CNOS in Chapter 3, which detects and segments novel objects from only their 3D models. CNOS outperformed supervised methods such as Mask R-CNN [44] and was recognized as the top-performing method for detecting and segmenting unseen objects in the BOP Challenge 2023 [51]. In Chapter 4, we revisited the limitations of traditional template matching for 6D pose estimation and proposed novel methods using local features, which demonstrated strong generalization. Chapter 5 extended this work with GigaPose, a faster and more robust approach to overcome limitations in similar settings. In Chapter 6, we presented NOPE, which leverages novel-view synthesis to estimate the pose of novel objects from a single image.

We also demonstrated in all of our proposed methods that "templates", which are 2D views of target 3D objects, is a very useful concept for pose estimation of novel objects. Templates not only allow natural generalization to novel objects by extending the set of templates but also facilitate robustness to domain gaps, lack of textures, and clutter and partial occlusions using discriminative features learnt with contrastive learning.

7.2 Future work

In this section, we discuss several paths for future research building upon our contributions.

7.2.1 Model-free novel object pose estimation

While we presented promising results for pose estimation from a single image settings in Chapter 5 and in Chapter 6, we found that there is still a significant gap in the

results compared to those obtained using ground-truth CAD models.

To address the gap, we can use additional templates created from reference video(s) as in model-free settings introduced in BOP Challenge 2024[1]. Specifically, in "static" reference video settings where object poses of templates are provided, we can seamlessly use our method. On the other hand, in "dynamic" reference videos settings where object poses of templates are not provided, we can first use methods such as BundleSDF [146] or Hampali *et al*. [41] to obtain the poses, and then further apply our method to estimate object poses. Relaxing the requirement of CAD models will enable more practical applications in real-world scenarios.

7.2.2 Articulated object pose estimation

This thesis focuses on rigid objects, but it is important to consider non-rigid objects commonly found in everyday settings, such as tools, furniture, and heavy equipment. Addressing the poses of articulated objects is therefore crucial in many applications, particularly in robotics. However, articulated objects possess additional degrees of freedom that complicate pose estimation compared to rigid objects due to their complex motion and articulation constraints. One strong baseline for this problem is to first estimate the pose of each object part independently using rigid object pose estimation methods such as the ones proposed in this thesis, and then refine them with articulated constraints.

[1] https://bop.felk.cvut.cz/challenges/bop-challenge-2024/

Bibliography

[1] Arslan Artykov, Antoine Guedon, Clementin Boittiaux, Van Nguyen Nguyen, and Vincent Lepetit. "MAGIC-GS: Monocular Articulated Generic Object Reconstruction with Gaussian Splatting". In: *In submission*. 2024.

[2] Guillaume Astruc, Nicolas Dufour, Ioannis Siglidis, Constantin Aronssohn, Nacim Bouia, Stephanie Fu, Romain Loiseau, Van Nguyen Nguyen, Charles Raude, Elliot Vincent, Lintao Xu, Hongyu Zhou, and Loic Landrieu. "Open Street View - 5M: The Many Roads to Global Visual Geolocation". In: *CVPR*. 2024.

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

[4] Nicholas Ayache and Olivier D. Faugeras. "HYPER: A New Approach for the Recognition and Positioning of Two-Dimensional Objects". In: *PAMI*. 1986.

[5] Vassileios Balntas, Andreas Doumanoglou, Caner Sahin, Juil Sock, Rigas Kouskouridas, and Tae-Kyun Kim. "Pose Guided RGBD Feature Learning for 3D Object Pose Estimation". In: *ICCV*. 2017.

[6] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. "SURF: Speeded up robust features". In: *ECCV*. 2006.

[7] Tolga Birdal and Slobodan Ilic. "Point pair features based object detection and pose estimation revisited". In: *3DV*. 2015.

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

[9] G. Bradski. "The OpenCV Library". In: *Dr. Dobb's Journal of Software Tools*. 2000.

[10] Garrick Brazil, Abhinav Kumar, Julian Straub, Nikhila Ravi, Justin Johnson, and Georgia Gkioxari. "Omni3d: A large benchmark and model for 3d object detection in the wild". In: *CVPR*. 2023.

[11] Rodney A. Brooks. "Symbolic Reasoning Among 3-D Models and 2-D Images". In: *Artif. Intell.* 1981.

[12] Benjamin Busam, Hyun Jun Jung, and Nassir Navab. "I Like to Move It: 6D Pose Estimation as an Action Decision Process". In: *arXiv*. 2020.

[13] Hongping Cai, Tomáš Werner, and Jiří Matas. "Fast detection of multiple textureless 3-D objects". In: *ICVS*. 2013.

[14] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. “Unsupervised Learning of Visual Features by Contrasting Cluster Assignments”. In: *NeurIPS*. 2020.

[15] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. “Emerging properties in self-supervised vision transformers”. In: *ICCV*. 2021.

[16] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. “Shapenet: An information-rich 3d model repository”. In: *arXiv*. 2015.

[17] Jianqiu Chen, Mingshan Sun, Tianpeng Bao, Rui Zhao, Liwei Wu, and Zhenyu He. “3D Model-Based Zero-Shot Pose Estimation Pipeline”. In: *arXiv*. 2023.

[18] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. “A Simple Framework for Contrastive Learning of Visual Representations”. In: *ICML*. 2020.

[19] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. “Big Self-Supervised Models are Strong Semi-Supervised Learners”. In: *NeurIPS*. 2020.

[20] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. “Improved Baselines with Momentum Contrastive Learning”. In: *ArXiv*. 2020.

[21] Xu Chen, Zijian Dong, Jie Song, Andreas Geiger, and Otmar Hilliges. “Category level object pose estimation via neural analysis-by-synthesis”. In: *ECCV*. 2020.

[22] C. Choi and H.I. Christensen. “3D Pose Estimation of Daily Objects Using an RGB-D Camera”. In: *IROS*. 2012.

[23] Changhyun Choi, Yuichi Taguchi, Oncel Tuzel, Ming-Yu Liu, and Srikumar Ramalingam. “Voting-based pose estimation for robotic assembly using a 3D sensor”. In: *ICRA*. 2012.

[24] Alex Clark et al. “The Pillow Imaging Library. https://github.com/python-pillow/Pillow”. In: *IEEE Robot. Autom. Mag.* 2015.

[25] Xinke Deng, Arsalan Mousavian, Yu Xiang, Fei Xia, Timothy Bretl, and Dieter Fox. “PoseRBPF: A Rao-Blackwellized Particle Filter for 6D Object Pose Tracking”. In: *RSS*. 2019.

[26] Maximilian Denninger, Martin Sundermeyer, Dominik Winkelbauer, Youssef Zidan, Dmitry Olefir, Mohamad Elbadrawy, Ahsan Lodhi, and Harinandan Katam. “Blenderproc”. In: *arXiv* (2019).

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

[28] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. “Google scanned objects: A high-quality dataset of 3d scanned household items”. In: *ICRA*. 2022.

[29] Bertram Drost and Slobodan Ilic. “3D object detection and localization using multimodal point pair features”. In: *3DIMPVT*. 2012.

[30] Bertram Drost, Markus Ulrich, Paul Bergmann, Philipp Hartinger, and Carsten Steger. “Introducing MVTec ITODD – A dataset for 3D object recognition in industry”. In: *ICCVW*. 2017.

[31] Bertram Drost, Markus Ulrich, Nassir Navab, and Slobodan Ilic. “Model globally, match locally: Efficient and robust 3D object recognition”. In: *CVPR*. 2010.

[32] Mihailo DS. *15 Types of Warehouse Robotics For Optimal Efficiency — modula.us*. https://modula.us/blog/warehouse-robotics/. [Accessed 20-05-2024].

[33] Yuming Du, Wen Guo, Yang Xiao, and Vincent Lepetit. “1st Place Solution for the UVO Challenge on Image-Based Open-World Segmentation 2021”. In: *ICCVW*. 2021.

[34] Yuming Du, Yang Xiao, and Vincent Lepetit. “Learning to Better Segment Objects from Unseen Classes with Unlabeled Videos”. In: *ICCV*. 2021.

[35] Maximilian Durner, Wout Boerdijk, Martin Sundermeyer, Werner Friedl, Zoltan-Csaba Marton, and Rudolph Triebel. “Unknown Object Segmentation from Stereo Images”. In: *IROS*. 2021.

[36] M. A. Fischler and R. C. Bolles. “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography”. In: *ACM*. 1981.

[37] Mathieu Garon and Jean-François Lalonde. “Deep 6-DOF Tracking”. In: *TVCG* (2017).

[38] Mathieu Garon, Denis Laurendeau, and Jean-François Lalonde. “A Framework for Evaluating 6DOF Object Trackers”. In: *ECCV*. 2018.

[39] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. “Yolox: Exceeding yolo series in 2021”. In: *arXiv* (2021).

[40] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. “Imagebind: One Embedding Space to Bind Them All”. In: *CVPR*. 2023.

[41] Shreyas Hampali, Tomas Hodan, Luan Tran, Lingni Ma, Cem Keskin, and Vincent Lepetit. “In-hand 3d object scanning from an rgb sequence”. In: *CVPR*. 2023.

[42] Zhao Han, Jenna Parrillo, Alexander Wilkinson, Holly A Yanco, and Tom Williams. “Projecting robot navigation paths: Hardware and software for projected AR”. In: *International Conference on Human-Robot Interaction (HRI)*. 2022.

[43] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. “Momentum Contrast for Unsupervised Visual Representation Learning”. In: *CVPR*. 2020.

[44] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. “Mask r-cnn”. In: *ICCV*. 2017.

[45] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep Residual Learning for Image Recognition”. In: *CVPR*. 2016.

[46] 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.

[47] Stefan Hinterstoisser, Vincent Lepetit, Naresh Rajkumar, and Kurt Konolige. "Going further with point pair features". In: *ECCV*. 2016.

[48] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. "Learning Deep Representations by Mutual Information Estimation and Maximization". In: *ICLR*. 2019.

[49] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising Diffusion Probabilistic Models". In: *NeurIPS*. 2020.

[50] Tomas Hodan, Frank Michel, Eric Brachmann, Wadim Kehl, Anders Glentbuch, Dirk Kraft, Bertram Drost, Joel Vidal, Stephan Ihrke, Xenophon Zabulis, et al. "Bop: Benchmark for 6D Object Pose Estimation". In: *ECCV*. 2018.

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

[52] Tomáš Hodaň. "Pose estimation of specific rigid objects". PhD thesis. Czech Technical University, 2021.

[53] Tomáš Hodaň, Dániel Baráth, and Jiří Matas. "EPOS: Estimating 6D Pose of Objects with Symmetries". In: *CVPR*. 2020.

[54] Tomáš Hodaň, Pavel Haluza, Štěpán Obdržálek, Jiří Matas, Manolis Lourakis, and Xenophon Zabulis. "T-LESS: An RGB-D Dataset for 6D Pose Estimation of Texture-less Objects". In: *WACV*. 2017.

[55] 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: *ECCV*. 2018.

[56] 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: *ECCV*. 2020.

[57] Yinlin Hu, Pascal Fua, and Mathieu Salzmann. "Perspective flow aggregation for data-limited 6d object pose estimation". In: *ECCV*. 2022.

[58] Yinlin Hu, Pascal Fua, Wei Wang, and Mathieu Salzmann. "Single-stage 6d object pose estimation". In: *CVPR*. 2020.

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

[60] Daniel P Huttenlocher and Shimon Ullman. "Recognizing rigid objects by aligning them with an image". In: *IJCV*. 1987.

[61] Ajay Jain, Ben Mildenhall, Jonathan T. Barron, Pieter Abbeel, and Ben Poole. "Zero-Shot Text-Guided Object Generation with Dream Fields". In: *CVPR*. 2022.

[62] Ajay Jain, Amber Xie, and Pieter Abbeel. “VectorFusion: Text-to-SVG By Abstracting Pixel-Based Diffusion Models”. In: *SIGGRAPH*. 2022.

[63] Hanwen Jiang, Zhenyu Jiang, Kristen Grauman, and Yuke Zhu. “Few-View Object Reconstruction with Unknown Categories and Camera Poses”. In: *3DV*. 2024.

[64] Roman Kaskman, Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. “Homebreweddb: RGB-D Dataset for 6D Pose Estimation of 3D Objects”. In: *ICCVW*. 2019.

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

[66] Wadim Kehl, Federico Tombari, Nassir Navab, Slobodan Ilic, and Vincent Lepetit. “Hashmod: A Hashing Method for Scalable 3D Object Detection”. In: *BMVC*. 2015.

[67] Nasir Mohammad Khalid, Tianhao Xie, Eugene Belilovsky, and Popa Tiberiu. “CLIP-Mesh: Generating Textured Meshes from Text Using Pretrained Image-Text Models”. In: *SIGGRAPH*. 2022.

[68] Eunyoung Kim and Gerard Medioni. “3D object recognition in range images using visibility context”. In: *IROS*. 2011.

[69] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, et al. “Segment Anything”. In: *arXiv*. 2023.

[70] Yann Labbé. “Pose estimation of rigid objects and robots”. PhD thesis. Ecole Normale Supérieure, 2023.

[71] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. “CosyPose: Consistent multi-view multi-object 6D pose estimation”. In: *ECCV*. 2020.

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

[73] Ales Leonardis and Horst Bischof. “Dealing with occlusions in the eigenspace approach”. In: *CVPR*. 1996.

[74] Vincent Lepetit. “Recent advances in 3d object and hand pose estimation”. In: *arXiv*. 2020.

[75] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. “EPnP: An Accurate O(n) Solution to the PnP Problem”. In: *IJCV* (2009).

[76] Fu Li, Ivan Shugurov, Benjamin Busam, Minglong Li, Shaowu Yang, and Slobodan Ilic. “Polarmesh: A star-convex 3d shape approximation for object pose estimation”. In: *RAL*. 2022.

[77] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. “DeepIM: Deep iterative matching for 6D pose estimation”. In: *ECCV*. 2018.

[78] Zhigang Li, Gu Wang, and Xiangyang Ji. “Cdpn: Coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation”. In: *ICCV*. 2019.

[79] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. "Magic3D: High-Resolution Text-to-3D Content Creation". In: *CVPR*. 2022.

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

[81] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. "Ssd: Single shot multibox detector". In: *ECCV*. 2016.

[82] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. "SyncDreamer: Learning to Generate Multiview-consistent Images from a Single-view Image". In: *ICLR*. 2024.

[83] Yuan Liu, Yilin Wen, Sida Peng, Cheng Lin, Xiaoxiao Long, Taku Komura, and Wenping Wang. "Gen6D: Generalizable Model-Free 6DoF Object Pose Estimation from RGB Images". In: *ECCV*. 2022.

[84] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, and Wenping Wang. "Wonder3D: Single Image to 3D using Cross-Domain Diffusion". In: *ICLR*. 2024.

[85] Ilya Loshchilov and Frank Hutter. "Decoupled Weight Decay Regularization". In: *ICLR*. 2019.

[86] David G Lowe. "Three-dimensional object recognition from single two-dimensional images". In: 1987.

[87] David G Lowe et al. "Object recognition from local scale-invariant features." In: *ICCV*. 1999.

[88] Fabian Manhardt, Diego Martin Arroyo, Christian Rupprecht, Benjamin Busam, Tolga Birdal, Nassir Navab, and Federico Tombari. "Explaining the Ambiguity of Object Detection and 6D Pose From Visual Data". In: *ICCV*. 2019.

[89] Lucas Manuelli, Wei Gao, Peter Florence, and Russ Tedrake. "kpam: Keypoint affordances for category-level robotic manipulation". In: *ISRR*. 2019.

[90] Octave Mariotti, Oisin Mac Aodha, and Hakan Bilen. "ViewNet: Unsupervised Viewpoint Estimation from Conditional Generation". In: *ICCV*. 2021.

[91] Octave Mariotti and Hakan Bilen. "Semi-Supervised Viewpoint Estimation with Geometry-aware Conditional Generation". In: *ECCVW*. 2020.

[92] Jiří Matas, Ondřej Chum, Martin Urban, and Tomáš Pajdla. "Robust wide-baseline stereo from maximally stable extremal regions". In: *IVC*. 2004.

[93] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. "RealFusion: 360^o Reconstruction of Any Object from a Single Image". In: *CVPR*. 2023.

[94] Ajmal Mian, Mohammed Bennamoun, and Robyn Owens. "On the repeatability and quality of keypoints for local feature-based 3D object retrieval from cluttered scenes". In: *IJCV*. 2010.

[95] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. "Nerf: Representing Scenes as Neural Radiance Fields for View Synthesis". In: *ECCV*. 2020.

[96] Ishan Misra and Laurens Van Der Maaten. "Self-Supervised Learning of Pretext-Invariant Representations". In: *CVPR*. 2020.

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

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

[99] Hiroshi Murase and Shree K Nayar. "Visual learning and recognition of 3-D objects from appearance". In: *IJCV*. 1995.

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

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

[102] Van Nguyen Nguyen, Thibault Groueix, Georgy Ponimatkin, Vincent Lepetit, and Tomas Hodan. "CNOS: A Strong Baseline for CAD-based Novel Object Segmentation". In: *CVPR*. 2023.

[103] Van Nguyen Nguyen, Thibault Groueix, Mathieu Salzmann, and Vincent Lepetit. "GigaPose: Fast and Robust Novel Object Pose Estimation via One Correspondence". In: *CVPR*. 2024.

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

[105] Aaron Oord, Yazhe Li, and Oriol Vinyals. "Representation Learning with Contrastive Predictive Coding". In: *ArXiv*. 2018.

[106] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. "Dinov2: Learning Robust Visual Features Without Supervision". In: *arXiv*. 2023.

[107] Keunhong Park, Arsalan Mousavian, Yu Xiang, and Dieter Fox. "LatentFusion: End-to-End Differentiable Reconstruction and Rendering for Unseen Object Pose Estimation". In: *CVPR*. 2020.

[108] Kiru Park, Timothy Patten, and Markus Vincze. "Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation". In: *ICCV*. 2019.

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

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

[111] Jean Ponce, Svetlana Lazebnik, Fredrick Rothganger, and Cordelia Schmid. "Toward true 3D object recognition". In: *RFIA*. 2004.

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

[113] Pyrender. *https://github.com/mmatl/pyrender*. URL: https://github.com/mmatl/pyrender (visited on 11/29/2024).

[114] Guocheng Qian, Jinjie Mai, Abdullah Hamdi, Jian Ren, Aliaksandr Siarohin, Bing Li, Hsin-Ying Lee, Ivan Skorokhodov, Peter Wonka, Sergey Tulyakov, and Bernard Ghanem. "Magic123: One Image to High-Quality 3D Object Generation Using Both 2D and 3D Diffusion Priors". In: *ICLR*. 2024.

[115] Mahdi Rad and Vincent Lepetit. "BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth". In: *ICCV*. 2017.

[116] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. "Learning Transferable Visual Models From Natural Language Supervision". In: *ICML*. 2021.

[117] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. "Hierarchical Text-Conditional Image Generation with CLIP Latents". In: *arXiv*. 2022.

[118] Lawrence G Roberts. "Machine perception of three-dimensional solids". PhD thesis. Massachusetts Institute of Technology, 1963.

[119] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. "High-Resolution Image Synthesis with Latent Diffusion Models". In: *CVPR*. 2022.

[120] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. "DreamBooth: Fine Tuning Text-To-Image Diffusion Models for Subject-Driven Generation". In: *arXiv*. 2022.

[121] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, et al. "Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding". In: *arXiv*. 2022.

[122] Samuele Salti, Federico Tombari, and Luigi Di Stefano. "SHOT: Unique signatures of histograms for surface and texture description". In: *CVIU*. 2014.

[123] Ivan Shugurov, Fu Li, Benjamin Busam, and Slobodan Ilic. "Osop: A multi-stage one shot object pose estimation framework". In: *CVPR*. 2022.

[124] Jiaming Song, Chenlin Meng, and Stefano Ermon. "Denoising Diffusion Implicit Models". In: *ICLR*. 2021.

[125] Yongzhi Su, Mahdi Saleh, Torben Fetzer, Jason Rambach, Nassir Navab, Benjamin Busam, Didier Stricker, and Federico Tombari. "ZebraPose: Coarse to Fine Surface Encoding for 6 DoF Object Pose Estimation". In: *CVPR*. 2022.

[126] Jiaming Sun, Zihao Wang, Siyu Zhang, Xingyi He, Hongcheng Zhao, Guofeng Zhang, and Xiaowei Zhou. "OnePose: One-Shot Object Pose Estimation Without CAD Models". In: *CVPR*. 2022.

[127] Martin Sundermeyer, Maximilian Durner, En Yen Puang, Zoltan-Csaba Marton, Narunas Vaskevicius, Kai O Arras, and Rudolph Triebel. "Multi-path learning for object pose estimation across domains". In: *CVPR*. 2020.

[128] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. "Implicit 3D Orientation Learning for 6D Object Detection from RGB Images". In: *ECCV*. 2018.

[129] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, and Rudolph Triebel. "Augmented Autoencoders: Implicit 3D Orientation Learning for 6D Object Detection". In: *IJCV*. 2020.

[130] Bugra Tekin, Sudipta N. Sinha, and Pascal Fua. "Real-Time Seamless Single Shot 6D Object Pose Prediction". In: *CVPR*. 2018.

[131] Yonglong Tian, Dilip Krishnan, and Phillip Isola. "Contrastive Multiview Coding". In: *ECCV*. 2020.

[132] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. "FCOS: A simple and strong anchor-free object detector". In: *PAMI*. 2020.

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

[134] Stephen Tyree, Jonathan Tremblay, Thang To, Jia Cheng, Terry Mosier, Jeffrey Smith, and Stan Birchfield. "6-DoF Pose Estimation of Household Objects for Robotic Manipulation: An Accessible Dataset and Benchmark". In: *IROS*. 2022.

[135] Ranjith Unnikrishnan and Martial Hebert. "Multi-scale interest regions from unorganized point clouds". In: *CVPRW*. 2008.

[136] Laurens Van der Maaten and Geoffrey Hinton. "Visualizing Data using t-SNE". In: *JMLR*. 2008.

[137] Joel Vidal, Chyi-Yeu Lin, Xavier Lladó, and Robert Martí. "A Method for 6D Pose Estimation of Free-Form Rigid Objects Using Point Pair Features on Range Data". In: *Sensors*. 2018.

[138] Yael Vinker, Ehsan Pajouheshgar, Jessica Y. Bo, Roman Christian Bachmann, Amit Haim Bermano, Daniel Cohen-Or, Amir Zamir, and Ariel Shamir. "CLIPasso: Semantically-Aware Object Sketching". In: *SIGGRAPH*. 2022.

[139] Eric Wahl, Ulrich Hillenbrand, and Gerd Hirzinger. "Surflet-pair-relation histograms: a statistical 3D-shape representation for rapid classification". In: *3DIM*. 2003.

[140] Chen Wang, Roberto Martín-Martín, Danfei Xu, Jun Lv, Cewu Lu, Li Fei-Fei, Silvio Savarese, and Yuke Zhu. "6-PACK: Category-Level 6D Pose Tracker with Anchor-Based Keypoints". In: *ICRA*. 2020.

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

[142] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A. Yeh, and Greg Shakhnarovich. "Score Jacobian Chaining: Lifting Pretrained 2D Diffusion Models for 3D Generation". In: *CVPR*. 2023.

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

[144] Daniel Watson, William Chan, Ricardo Martin-Brualla, Jonathan Ho, Andrea Tagliasacchi, and Mohammad Norouzi. "Novel View Synthesis with Diffusion Models". In: *ICLR*. 2023.

[145] Bowen Wen and Kostas Bekris. "Bundletrack: 6D Pose Tracking for Novel Objects Without Instance or Category-Level 3D Models". In: *ICIRS*. 2021.

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

[147] Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. "Foundationpose: Unified 6d pose estimation and tracking of novel objects". In: *CVPR*. 2024.

[148] Paul Wohlhart and Vincent Lepetit. "Learning Descriptors for Object Recognition and 3D Pose Estimation". In: *CVPR*. 2015.

[149] Zhirong Wu, Yuanjun Xiong, S. Yu, and D. Lin. "Unsupervised Feature Learning via Non-parametric Instance Discrimination". In: *CVPR*. 2018.

[150] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. "PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes". In: *RSS*. 2018.

[151] Yu Xiang, Christopher Xie, Arsalan Mousavian, and Dieter Fox. "Learning RGB-D Feature EMbeddings for Unseen Object Instance Segmentation". In: *CoRL*. 2021.

[152] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. "Sun database: Large-scale scene recognition from abbey to zoo". In: *CVPR*. 2010.

[153] Yang Xiao, Yuming Du, and Renaud Marlet. "PoseContrast: Class-Agnostic Object Viewpoint Estimation in the Wild with Pose-Aware Contrastive Learning". In: *3DV*. 2021.

[154] Yang Xiao, Xuchong Qiu, Pierre-Alain Langlois, Mathieu Aubry, and Renaud Marlet. "Pose from shape: Deep pose estimation for arbitrary 3d objects". In: *BMVC*. 2019.

[155] Christopher Xie, Yu Xiang, Arsalan Mousavian, and Dieter Fox. "Unseen Object Instance Segmentation for Robotic Environments". In: *Transactions on Robotics*. 2021.

[156] Yann LeCun. *A Path Towards Autonomous Machine Intelligence*. Tech. rep. Meta, 2022.

[157] Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. "Dpod: 6d pose object detector and refiner". In: *ICCV*. 2019.

[158] Jason Y. Zhang, Deva Ramanan, and Shubham Tulsiani. "RelPose: Predicting Probabilistic Relative Rotation for Single Objects in the Wild". In: *ECCV*. 2022.

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

[160] Xu Zhao, Wenchao Ding, Yongqi An, Yinglong Du, Tao Yu, Min Li, Ming Tang, and Jinqiao Wang. "Fast Segment Anything". In: *arXiv*. 2023.

[161] Yuyang Zhao, Zhun Zhong, Nicu Sebe, and Gim Hee Lee. "Novel Class Discovery in Semantic Segmentation". In: *CVPR*. 2022.

[162] Yu Zhong. "Intrinsic shape signatures: A shape descriptor for 3D object recognition". In: *ICCVW*. 2009.

[163] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. "Random erasing data augmentation". In: *AAAI*. 2020.

[164] Huizhong Zhou, Benjamin Ummenhofer, and Thomas Brox. "DeepTAM: Deep Tracking and Mapping". In: *ECCV*. 2018.

[165] Xingyi Zhou, Arjun Karpur, Linjie Luo, and Qixing Huang. "StarMap for Category-Agnostic Keypoint and Viewpoint Estimation". In: *ECCV*. 2018.

[166] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. "On the continuity of rotation representations in neural networks". In: *CVPR*. 2019.

[167] Zhizhuo Zhou and Shubham Tulsiani. "SparseFusion: Distilling View-conditioned Diffusion for 3D Reconstruction". In: *CVPR*. 2023.