applied sciences

MDPI

Article

DCSPose: A Dual-Channel Siamese Framework for Unseen Textureless Object Pose Estimation

Zhen Yue [1,2], Zhenqi Han [1], Xiulong Yang [1] and Lizhuang Liu [1,*]

[1] Shanghai Advanced Research Institute, Chinese Academy of Sciences, Shanghai 201210, China; yuez@sari.ac.cn (Z.Y.)
[2] University of Chinese Academy of Sciences, Beijing 100049, China
* Correspondence: liulz@sari.ac.cn

Featured Application: This work has the capability to estimate the pose of unseen textureless objects, which can find applications in automated assembly processes. For example, it can assist a robotic arm in assembly or docking tasks by determining the relative spatial poses of multiple objects. Additionally, this work has potential applications in augmented reality tasks, such as estimating the pose of a mold to enable coloring or overlaying visual effects on the mold.

Abstract: The demand for object pose estimation is steadily increasing, and deep learning has propelled the advancement of this field. However, the majority of research endeavors face challenges in their applicability to industrial production. This is primarily due to the high cost of annotating 3D data, which places higher demands on the generalization capabilities of neural network models. Additionally, existing methods struggle to handle the abundance of textureless objects commonly found in industrial settings. Finally, there is a strong demand for real-time processing capabilities in industrial production processes. Therefore, in this study, we introduced a dual-channel Siamese framework to address these challenges in industrial applications. The architecture employs a Siamese structure for template matching, enabling it to learn the matching capability between the templates constructed from high-fidelity simulated data and real-world scenes. This capacity satisfies the requirements for generalization to unseen objects. Building upon this, we utilized two feature extraction channels to separately process RGB and depth information, addressing the limited feature issue associated with textureless objects. Through our experiments, we demonstrated that this architecture effectively estimates the three-dimensional pose of objects, achieving a 6.0% to 10.9% improvement compared to the state-of-the-art methods, while exhibiting robust generalization and real-time processing capabilities.

Keywords: object pose estimation; textureless; template matching; siamese network

check for updates

Citation: Yue, Z.; Han, Z.; Yang, X.; Liu, L. DCSPose: A Dual-Channel Siamese Framework for Unseen Textureless Object Pose Estimation. *Appl. Sci.* **2024**, *14*, 730. https://doi.org/10.3390/app14020730

Academic Editor: Jesús B. Alonso-Hernández

Received: 27 October 2023
Revised: 4 January 2024
Accepted: 13 January 2024
Published: 15 January 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In recent years, there has been a growing demand for object pose estimation, and the utilization of deep learning technologies has significantly enhanced estimation accuracy. The emergence of large annotated datasets, such as those involved in BOP Benchmark [1,2], has fostered research in this field. However, despite the considerable progress these researchers have made in areas such as robotics [3], autonomous driving [4], and augmented reality [5], significant challenges persist within the field of industrial applications.

The most prominent challenge in industrial settings is the need to deal with a wide variety of objects. As a result, there is a heightened requirement for the algorithm's generalization capability, enabling rapid adaptation to new target objects. Moreover, current mainstream 3D pose estimation methods often necessitate extensive annotated data for training to achieve satisfactory estimating performance [6,7]. However, annotating 3D pose data imposes higher demands in terms of accuracy, difficulty, and cost compared to

annotating 2D data. These impediments limit the practical application of many research algorithms in industrial production.

Furthermore, in industrial manufacturing, the objects to be processed frequently exhibit a lack of texture or possess minimal textural features, such as ceramics, resin, and metallic components. Pose estimation tasks for these objects find diverse applications in industry, such as object grasping, assembly, and stacking. The scarcity of texture features makes it challenging to extract sufficient information for accurate pose estimation through traditional feature detection methods. Low accuracy in target pose estimation can adversely affect various industrial applications. For instance, robots may fail to precisely determine the position and orientation of objects during grasping tasks, leading to unsuccessful grabs. Similarly, misalignment during assembly processes can result in component damage. Consequently, addressing the problem of pose estimation for textureless objects in industrial production scenarios remains a critical concern in this field [8].

Finally, systems applied in industrial production typically require sufficient real-time processing capabilities. Overly complex network designs or rendering and registration processes that demand extensive computational resources can jeopardize real-time performance. Therefore, proposing an efficient and accurate estimation framework that can adapt to industrial production is both necessary and challenging.

The primary objective of this research was to address the three major challenges encountered in object pose estimation tasks within industrial settings, as described above.

To reduce dependence on annotated data and meet the demands for generalization, an intuitive approach is template matching. Capturing multi-angle information of the objects to be detected in advance allows for the construction of a rich and scalable template library, where each template corresponds to a specific pose parameter. When real-time estimation is required, the on-site scene is compared with the template library to find the best-matching template. The pose parameters of the matching template are then used as the prediction result. If the process of building the template library can be automated, the system will be able to quickly adapt to new objects. Furthermore, by dividing the entire matching process into offline template set generation and online matching, computational resources and time costs during runtime processing can be reduced.

Recent work has ingeniously tackled the task of pose estimation by utilizing template matching methods, leading to noteworthy results [9–11]. Template-based approaches primarily rely on computing the similarity between query images from real-world scenes and template images rendered from CAD models [12,13]. By constructing template sets using CAD models, these methods inject new object features into the neural network model, ensuring it retains sufficient matching capabilities when confronted with novel objects.

However, previous work has not demonstrated satisfactory performance when dealing with textureless objects. Textureless objects tend to provide less texture and geometric feature information [14,15]. It has been observed that relying solely on RGB information is insufficient to achieve optimal matching results. RGB information is also susceptible to environmental lighting conditions, leading to challenges in maintaining consistent performance when there are significant differences between training and testing scenes.

The reduced cost and increased availability of industrial depth cameras have made it more convenient to obtain depth information for objects in industrial settings. Distinguishing itself from RGB, depth data can reflect the surface geometry of the target object. This capability aids in distinguishing between different poses of textureless objects, thus generating valuable data. Therefore, utilizing depth information for feature extraction and fusion is a viable approach to explore.

An analysis of real-world industrial scenarios and previous research results prompted us to design a novel template matching approach that leverages multiple types of data. We constructed a template set comprising predefined pose parameters, encompassing both RGB images and depth maps, generated with high-fidelity rendering using CAD models. Subsequently, we employed a dual-channel feature extractor to obtain fused features from RGB and depth data, followed by training using a Siamese network structure. Finally, by

computing the similarity between features, we identified the template that best matched the real-world scene, enabling us to estimate the object's pose, as illustrated in Figure 1. To facilitate visualization, the depth map is displayed in pseudo-color format.

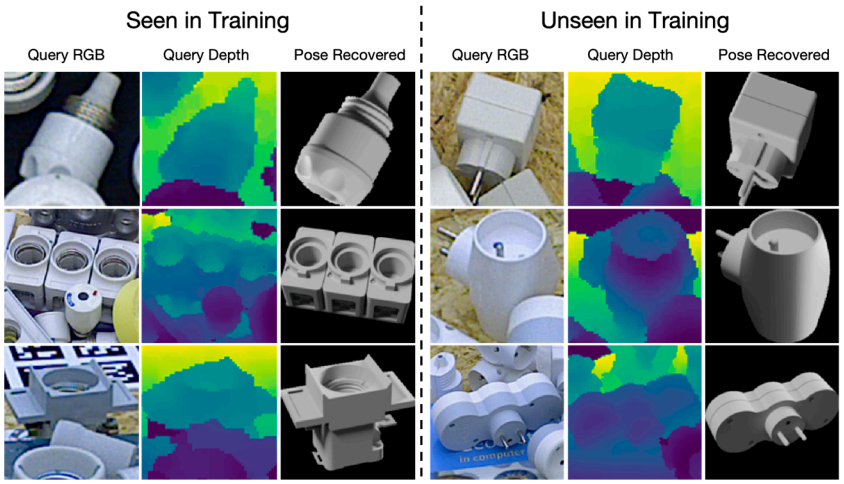


Figure 1. Our framework can estimate the pose of textureless objects, even for new categories of objects that have not been seen during training.

This approach facilitates the alignment of real-world images with rendered templates, harnessing multidimensional features from both RGB and depth data to enhance the comprehensiveness and robustness of pose estimation. Moreover, our method demonstrates high versatility when handling object categories unseen during the training phase. By simply providing the CAD model of a new object, the network's matching capabilities can be transferred through the automated construction of templates for the new object.

We conducted extensive experiments on the designed network framework. The experimental results demonstrate that our pose estimation framework, DCSPose, effectively enhances the pose estimation of novel textureless objects. The dual-channel structure enables feature extraction from multiple sources while preventing premature feature fusion from diverse origins. Through automated template construction and employing contrastive learning, the dependence on annotated data from real scenes can be reduced. Due to the adoption of a template matching method with separate offline and online stages, our framework exhibits strong generalization and real-time processing capabilities.

We summarize the contributions of this work as follows:

1. To address the need for pose estimation of unseen textureless objects in industrial settings, we introduced a novel network architecture. This network comprises two distinct feature extraction branches, one focused on RGB features and the other on depth features. Utilizing a contrastive learning approach, we learned the correlation between real and simulated data from these two feature types. Our joint feature learning framework effectively enhances the performance of pose estimation for unseen textureless objects.

2. To facilitate the matching of depth features, we generated a new template set based on T-LESS using CAD models. This offline-constructed template set comprises RGB projections and depth maps of objects from T-LESS at various viewpoints. By utilizing this rendering template set, our approach allows for the rapid application of the model to different types of target objects in industrial production. We have made this template set publicly available, facilitating further research by subsequent investigators.

3. We conducted comprehensive experiments and analyses to demonstrate that the inclusion of depth map templates and additional feature processing channels effectively enhances the pose estimation for textureless objects. By introducing low-cost depth sensors in industrial production, our approach can significantly improve estimation accuracy.

2. Related Work

2.1. Pose Estimation

Object pose estimation is a critical and valuable research area in the field of computer vision, with its primary objective being the identification of an object's position and orientation in three-dimensional space. Building upon the foundation of object detection tasks, pose estimation introduces more advanced technical requirements and holds the potential for a growing range of practical applications. This technology currently plays a pivotal role in a variety of application scenarios, and its demand is steadily increasing with the development of industrial automation and spatial computing technologies. Currently, in specific application domains, there has been notable research progress, such as in industrial automation [16–18], autonomous driving [19–21], and virtual and augmented reality [22,23], among others. In the context of industrial automation, object pose estimation serves as a critical tool for assisting robots in determining the position and orientation of objects, enabling more precise grasping, assembly, or docking. In the field of autonomous driving, this technology contributes to enhanced environmental perception in vehicles, aiding in obstacle avoidance and route planning. In augmented reality scenarios, this technology can perceive the orientation of objects, facilitating object re-rendering and the application of special effects.

Thanks to the advancements in deep learning, there has been a surge of exciting research in recent years in the field of object pose estimation. We can categorize deep learning methods for object pose estimation into the following types. **Regression Methods**: These methods use deep neural networks to directly regress the 6D pose of the camera or the objects [19,24,25]. The advantage of this approach is its speed in prediction. However, it requires a substantial amount of annotated data to train the network and is sensitive to noise and occlusions. Moreover, such methods are often useless when it comes to objects that have not been seen during training. **Retrieval or Classification Methods**: Drawing inspiration from template matching, these methods transform the pose estimation problem into a retrieval or classification task [9,11,26,27]. This approach is robust against noise and occlusions but necessitates a large template library for matching. **Alignment or Registration Methods**: Such methods typically employ an iterative or mathematical approach for image alignment and point cloud registration, utilizing 2D projections or 3D point cloud representations for matching. The iterative process involves rendering, matching, or registration, ultimately yielding the pose parameters that fulfill the error criteria [28–30].

Deep learning methods have demonstrated excellent performance in many object pose estimation tasks. However, they also encounter several challenges and limitations, with the most prominent being the requirement for large volumes of annotated data. Unlike conventional object detection and classification tasks, collecting and annotating 6D pose data for objects is a costly and time-consuming endeavor. This significantly impedes the practical application of deep learning-based pose estimation algorithms. Some researchers have conducted research and practice on this problem. For instance, ref. [31] generated training data for object poses by constructing a photorealistic textured mesh model and rendering it from various perspectives and distances.

Another significant challenge lies in the need for model generalization in practical applications. Deep learning models might encounter difficulties in maintaining their performance when faced with situations different from the training data, such as variations in lighting conditions, backgrounds, or viewpoints. Enhancing a model's performance for unseen objects is also a crucial concern in practical applications. Additionally, improving a model's generalization capability can also help reduce training costs.

The need for real-time processing is yet another challenge that must be addressed in practical applications [32]. Deep learning models typically demand substantial computational resources and time for inference. This might not be suitable for real-time applications and resource-constrained environments. Designing straightforward and efficient pose estimation algorithms, along with optimizing the detection workflow, becomes crucial in

addressing this issue. Reasonably choosing estimation strategies, such as favoring regression or retrieval methods over computationally resource-intensive real-time rendering and alignment methods, can contribute to enhancing efficiency. Addressing these challenges and limitations in deep learning-based object pose estimation is essential for ensuring its practical applicability in a wide range of industries and scenarios.

2.2. Template-Based Method

Many deep-learning-based methods rely on large, annotated 3D model datasets to improve the accuracy and speed of object pose estimation. However, datasets with pose annotations are relatively scarce and have limited sample sizes. Annotating the pose parameters of objects is more challenging than annotating object categories, particularly in non-standard environmental conditions. This scarcity of annotated pose data makes it difficult to train high-quality pose estimators and significantly hampers the application of data-driven deep learning methods to new objects and scenarios.

Indeed, when dealing with unknown shapes and types of objects and wishing to avoid retraining neural network models, template matching is a highly intuitive solution. Traditional template matching often requires manually designed matching features and may exhibit limited robustness against variations in background or lighting conditions. Therefore, combining deep learning feature extraction with the template matching concept is a logical and empirically proven approach.

Deep learning methods based on the template matching concept have substantial potential in addressing the problem of pose estimation for unknown objects. These methods typically involve three key processes: template set creation, feature extraction and mapping, and feature matching.

Templates Creation. To handle objects from different viewpoints and poses, a collection of templates needs to be created. These templates can be images representing the object from various poses. Each template in the set represents a possible pose of the object.

Feature Extraction and Mapping. Deep Neural Networks are used to extract features from input images. These features capture essential information from the images, such as object edges and textures. To further compress and encode the features effectively, they can be mapped to a feature space where they maintain good discriminability. Feature mapping projects the extracted image features into the feature space of the template set.

Feature Matching. Based on the results of feature mapping, the most similar template to the input image can be selected. This selection process often involves comparing feature similarity scores and identifying the template with the highest score. Once the most similar template is found, its pose information can be used to estimate the object's pose in the input image.

This approach effectively combines the advantages of deep learning feature extraction and template matching, making it a powerful method for object pose estimation, especially in scenarios involving novel objects and diverse viewpoints. The similarity-based template matching process provides a reliable way to estimate object poses even when dealing with challenging conditions like variations in lighting, occlusions, and textureless objects.

2.3. Contrastive Learning and Siamese Network

Contrastive learning is a self-supervised learning method designed to map similar samples to nearby points while pushing dissimilar samples apart. In contrastive learning, a key concept involves training a model by maximizing the similarity score of similar sample pairs and minimizing the similarity score of dissimilar sample pairs. Typically, contrastive learning could employ Siamese network structures to learn the similarity between samples [33–37]. Siamese networks are a type of deep neural network structure, usually composed of two or more identical sub-networks, which handle different input samples but share the same weights. By training Siamese networks to maximize the similarity scores of similar sample pairs and minimize those of dissimilar sample pairs, the goal of feature matching can be achieved. Furthermore, the Siamese structure offers a

method for learning from unlabeled data, learning knowledge by comparing instead of directly using labels [38]. Training with a Siamese structure helps to address the challenge of insufficient labeled data.

In the task of object pose estimation, contrastive learning and Siamese networks can be employed to measure the similarity between input images and templates. This aids in identifying the template that is most similar to the input image, thereby facilitating the estimation of the object's pose.

3. Materials and Methods

3.1. Framework

We devised a dual-channel Siamese framework to estimate object pose parameters in real-time detection scenarios. Given the RGB and depth images of a target region during real-time detection, this framework can locate the most similar template from the template set constructed through automated template creation and use the template's corresponding pose as the estimation result.

The inference process of the entire framework can be divided into two parts: offline and online, as illustrated in Figure 2. The offline part encompasses template rendering and feature extraction, with the goal of constructing a template library and a feature repository for the objects to be detected. The template set was generated offline using a CAD model, which is elaborated in Section 3.2. The online part involves feature extraction from data collected in real time, as well as performing similarity retrieval against the template feature set to ascertain the best-matching pose.

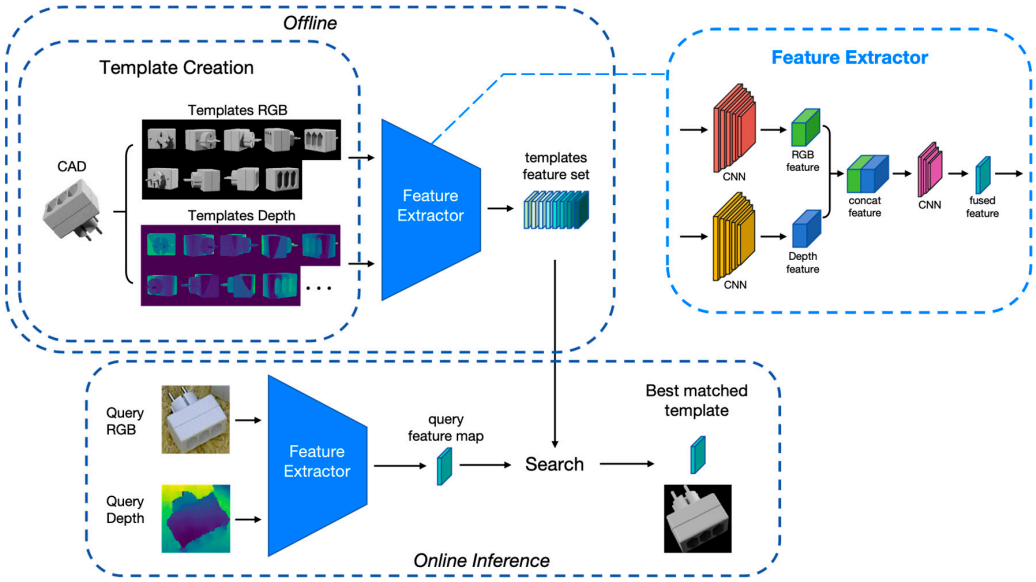


Figure 2. The dual-channel Siamese framework. This framework achieves object pose estimation of objects by matching online scenes and offline templates.

The feature extraction process is facilitated through a dual-channel feature extractor capable of extracting and integrating both RGB textures and depth features, elucidated in Section 3.3. The feature extractor was trained using contrastive learning, employing a Siamese structure, without separating the offline and online phases. Hence, the entire framework simultaneously employs two feature extractors, dedicated to processing synthetic templates and real-world scene data, respectively. Each feature extractor comprises two channels responsible for extracting RGB and depth features, resulting in a total of four data pathways. Details regarding model training are elaborated in Section 3.4.

In this framework, the term "dual-channel" refers to the two processing pathways for RGB and depth features in the feature extractor, distinct from the two sub-networks used in the Siamese network structure for handling real and simulated scenes.

3.2. Template Creation

To support accurate estimation of object poses from multiple viewpoints, it is necessary to project and render the object CAD model from various perspectives. Initially, a unit sphere constructed, and multiple points are sampled on the sphere to obtain different viewpoints. These viewpoints can be used to render the object from different angles using a high-fidelity rendering engine, generating multi-view projection results. These images are then used to build the RGB template library.

While obtaining projections from different angles, it is also crucial to calculate the distances from each pixel on the projection plane to the camera for each viewpoint. This information yields depth maps for each viewpoint, which are used to construct the depth template library. The RGB and depth templates are organized based on viewpoints and plane rotations, with corresponding pose information labeled, resulting in a comprehensive template library. The process of creating templates with various poses using a polyhedral sphere is illustrated in Figure 3.

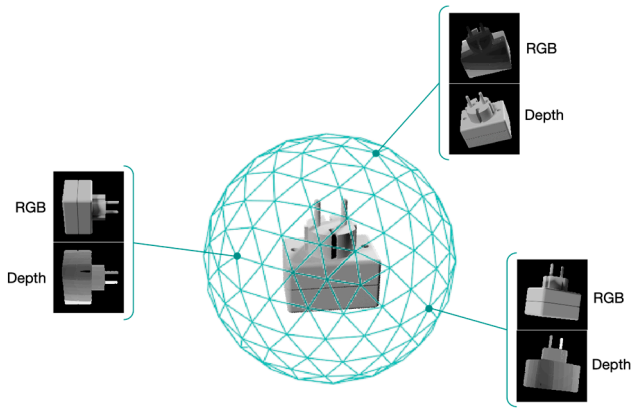


Figure 3. Template creation with a polyhedral sphere.

We followed a similar approach as used in refs. [9,11], where a polyhedron was created with 602 viewpoints and 36 plane rotations at 10-degree intervals. By combining different viewpoints and plane rotations, we generated a total of 21,672 viewpoints for each object. We utilized BlenderProc [39] for high-fidelity rendering to obtain RGB projections, depth maps, and masks from various viewpoints. In the end, we rendered 1806 images as a template library of individual objects. Figure 4 shows how a single object would be rendered from several perspectives. This strategy allows for a comprehensive set of templates to cover a wide range of object poses and rotations, which is beneficial for accurate pose estimation.

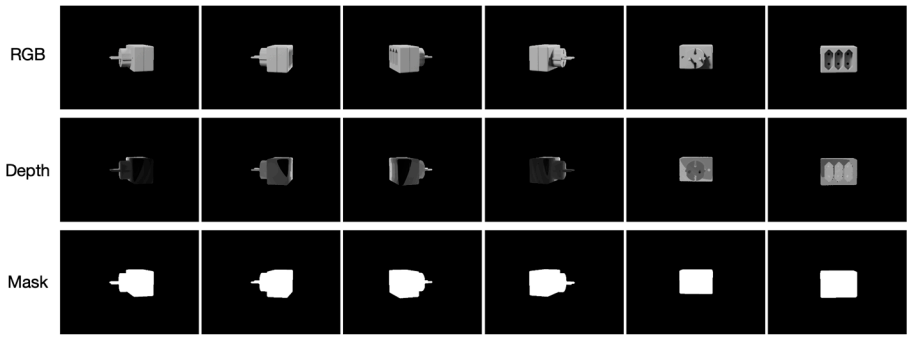


Figure 4. The rendering results from certain viewpoints include RGB images, depth maps, and mask images.

3.3. Feature Extractor

We employed ResNet50 [40] as the backbone of the feature extraction network and set up two parallel channels for extracting RGB and depth information. This means that there were two separate ResNet50 networks dedicated to extracting features. These two channels did not share weights during the training process, facilitating their individual focus on the color and geometric features respectively. After obtaining the RGB feature map and the depth feature map, we concatenated these two feature maps while preserving pixel correspondences. Subsequently, these concatenated feature maps were passed through a three-layer 1×1 convolution network to integrate them into a composite feature map. In the end, the feature extractor converted two query images with sizes of $3 \times 224 \times 224$ into a $16 \times 28 \times 28$ feature map, which represents the pose characteristics of the input images.

3.4. Training Phase

Positive and negative sample pairs between the template and real data were constructed based on pose discrepancies. Subsequently, two feature extractors with shared weights extracted features from these sample pairs. The similarity between features of sample pairs is then computed, facilitating network training through contrastive learning. This entire process constitutes the Siamese structure, as depicted in Figure 5.

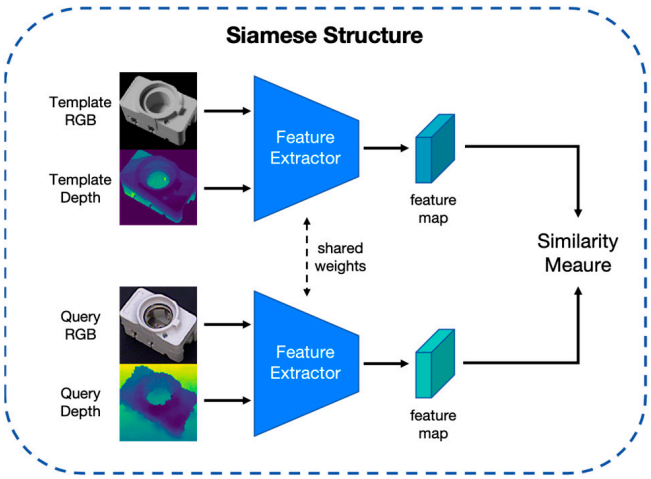


Figure 5. Siamese structure for training.

Construction of learning samples. To train the network to excel in matching, it is important to construct positive and negative sample pairs. Positive sample pairs consist of matching viewpoints between the query and simulated template images, while negative sample pairs consist of differing viewpoints between the query and template images. Given that viewpoint sampling is discrete, and the discriminability of different viewpoint poses depends on the sampling density, a balance is needed. Taking all factors into account, we consider views with an angular difference of less than 5 degrees to be classified as positive sample pairs. Views with an angular difference greater than 5 degrees are regarded as negative sample pairs. Consequently, for each object, we can generate N positive sample pairs and $(N-1) \times N$ negative sample pairs.

Loss and Similarity. We referenced the analysis and experimental results of several common contrastive learning loss functions in [9]. From these findings, We opted for the InfoNCE loss function [41] for our model training:

$$L_{InfoNCE} = -\sum_{i}^{N} log \frac{exp(S(q,t))}{\sum_{k}^{N} 1_{k \neq i} exp(S(q,t))},$$

where $S(q,t)$ represents the similarity between query and template feature maps, utilizing cosine similarity. This approach aims to maximize the consistency between positive sample

pairs while minimizing the consistency between negative sample pairs. It helps the network learn to match the query and template images effectively.

4. Experiments

4.1. Experimental Setup

Dataset. We employed the T-LESS [42] dataset for our evaluation, as it contains 30 industry-relevant objects that lack significant texture or discriminative color. To assess the robustness of our model on unseen objects, we followed the evaluation criteria outlined in [9,11,26]. Specifically, we used objects 1–18 for training and reserved objects 19–30 as unseen objects to validate the model's generalization capability. The diversity in object shapes is illustrated in Figure 6, and there is a substantial visual contrast between seen and unseen objects.

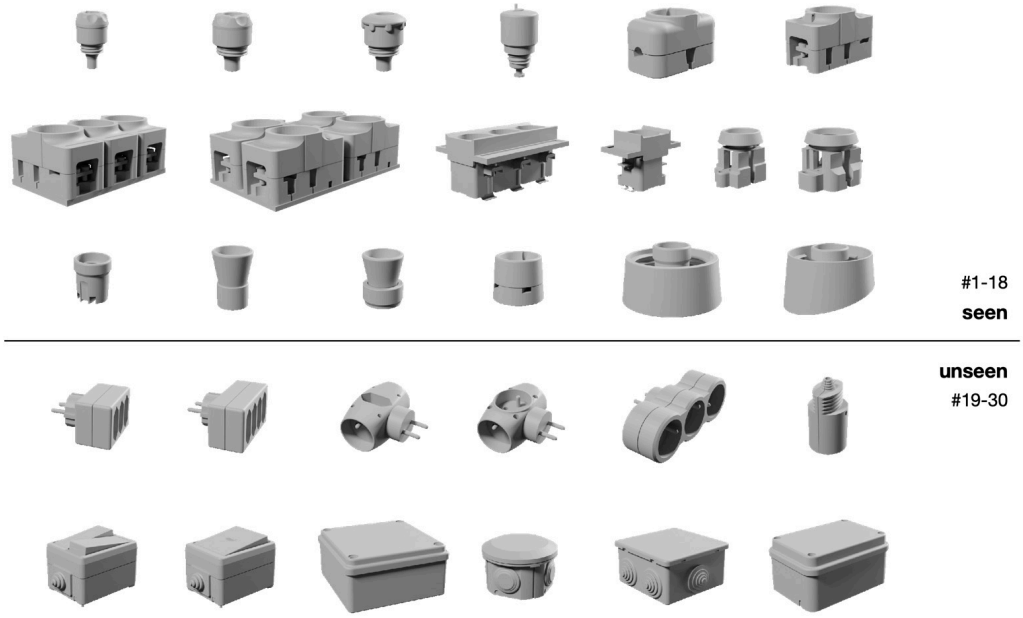


Figure 6. Dataset division. Objects numbered 1 to 18 are allocated for training, while all objects are used for testing. Additionally, objects numbered 19 to 30 are utilized to assess generalization capabilities.

Data preprocessing. In this work, the RGB and depth data used were captured by a structured-light RGB-D camera, Primesense CARMINE 1.09, which has an operation range of 0.35–3 m. To adapt to the differences between template and real-world depth images, preprocessing of depth maps is necessary. For template depth maps, the first step involves removing the background generated during rendering, followed by normalizing the object's depth values to a range of 0 to 255. As for the depth map of the query, the preprocessing does not consider the background. After removing invalid values such as holes, the depth was directly normalized to the range of 0 to 255.

For facilitating performance comparisons across different models, object bounding boxes from the T-LESS dataset were employed to crop RGB and depth images, subsequently adjusted in width and height to fit a 224×224 size. Similarly, for rendered template images, bounding boxes were obtained using binary masks of the templates and subsequently utilized to crop RGB and depth templates. This method facilitated the scaling of objects in both template and query images to approximate sizes, thereby mitigating discrepancies in object imaging sizes at different distances.

Data Augmentation. During the training process, to ensure the network can cope with interference from different backgrounds, random background and image augmentation techniques were applied to the RGB and depth data. This approach helps the network generalize effectively to various background conditions that may be encountered during real-world detection scenarios. Specifically, we performed image segmentation on the original dataset to remove the background, followed by the addition of random backgrounds with the aim of making the network adaptable to unknown and complex backgrounds. Additionally, we also conducted traditional data augmentation, which involved random adjustments to contrast, brightness, as well as Gaussian blur, sharpening, and other filtering operations. These aforementioned procedures were applied to both RGB and depth images. However, when processing the depth images, we omitted brightness-related adjustments, and the random backgrounds for depth images were grayscale. The input images used during training can be referenced in Figure 7.

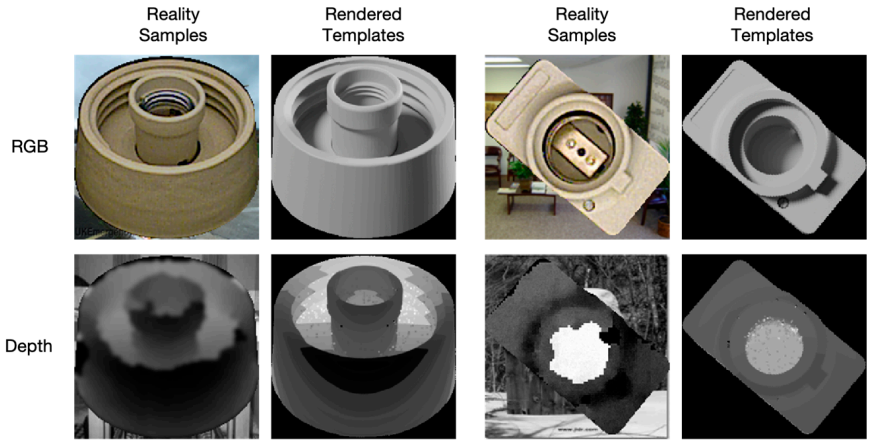


Figure 7. Examples of training data. The reality samples were further processed from the T-LESS dataset, while the rendered images were generated through template rendering operations. Our model is designed to learn the correspondence between real-world scenes and simulated images.

Evaluation metrics. It is a common practice to use recall under the Visible Surface Difference (VSD, err_{vsd}) as evaluation metrics, particularly for objects with many symmetrical structures such as the T-LESS dataset. The VSD is an ambiguity-invariant pose error function that is determined by the distance between the estimated and ground truth visible object depth surfaces [26]. It follows that

$$err_{VSD} = avg_{p \in \hat{V} \cup \overline{V}} \begin{cases} 0 & if\ p \in \hat{V} \cap \overline{V} \wedge \left| \hat{D}(p) - \overline{D}(p) \right| < \tau \\ 1 & otherwise \end{cases}$$

where $\hat{D}$ and $\overline{D}$ are distance maps obtained by rendering the object model in the estimated pose and the ground-truth pose respectively, $\hat{V}$ and $\overline{V}$ are the visibility masks obtained by comparingthe depth maps. As in the BOP benchmark [1,9,11,27], we report the recall of 6D object poses at $err_{vsd} < 0.3$ with tolerance τ = 20 mm and >10% object visibility.

The primary focus of this paper is on the identification of matching pose templates. However, for the purpose of facilitating result comparison, we adopted the "Projection Distance Estimation" method from [43] as employed in [11,26,28]. This method leverages the input bounding boxes from query images to compute the translation of objects. The work in this paper does not involve object classification, presuming that object categories are known during the testing phase.

Implementation Details. We conducted the experiments using an NVIDIA GeForce RTX 4090 GPU. The hyperparameters for the training process were kept similar to those used in related works [9,11,26]. Practical testing was conducted to achieve faster convergence speeds. Specifically, the model's training was performed using the Adam optimizer, and the training parameters can be referenced in Table 1. We utilized MOCOv2 [44] as the pre-training parameter to initialize the ResNet50 used in the feature extraction, aiming to reduce the training time. The whole model typically required training for 30 to 40 epochs. We set the learning rate to decay by a certain proportion at specific milestones.

Table 1. Experimental parameters. The processors used for the experiments and the parameters employed during training.

Hardware	CPU	Intel(R) Xeon(R) Platinum 8336C CPU @ 2.30 GHz
	GPU	NVIDIA GeForce RTX 4090
Training	Batch size	12
	Optimizer	Adam
	Learning rate	2.0×10^{-4}
	Weight decay	5×10^{-4}
	Milestones	20, 30, 35
	Decay rate	0.2

4.2. Comparison

In Table 2, we present the performance of our method compared to other state-of-the-art contrastive learning algorithms on the T-LESS dataset. Thanks to contrastive learning, all models exhibited robust generalization capabilities, but our approach showed significant improvements in the task of textureless object recognition compared to other methods. Visual results comparisons between our method and state-of-the-art method are presented in Figures A1 and A2, demonstrating that our method can provide more accurate pose estimations in some cluttered scenes.

Table 2. Quantitative comparison with [9,11,26] on seen (#1–18) and unseen objects (#19–31) of T-LESS using the same protocol from [26].

Method	**Number Templates**	**Mean VSD Recall**		
		Seen Obj. 1–18	**Unseen Obj. 19–30**	**Total**
MPL [11]	92 K	35.25	33.17	34.42
Implicit [26]	92 K	35.60	42.45	38.34
TPO [9]	21 K	59.14	56.91	58.25
DCSPose (ours)	21 K	65.86	62.53	64.20

4.3. Experiment of Depth Mask

Depth maps represent the distance between objects and the camera in the scene, so the depth values can vary significantly in different scenarios. However, the relative depth values within the object region tend to remain relatively consistent and are related to the object's size. Therefore, intuitively, separating the background from the depth map and normalizing the depth within the object's region should enhance the utility of the depth map.

To further explore the value of depth features, an experimental group was established by introducing masks into depth images to segment and isolate the depth data containing only object regions. These masks used were derived from the T-LESS dataset. The segmented depth images underwent the data preprocessing operations outlined in Section 4.1, ultimately scaling the depth values to fit within the range of 0 to 255.

These segmented depth images were utilized throughout both the training and testing phases of the network. The evaluation outcomes of this experiment are presented in Table 3, where the suffix 'masked' denotes the introduction of masks. The comparative experiment's results align with our hypothesis: undisturbed object depth information contributes to enhancing pose estimation accuracy, as it contains purer geometric features of the object surfaces without background interference. However, this enhancement comes at a cost, as it requires additional instance segmentation masks from the upstream task.

Table 3. Experimental results. The "basic" is the foundational version of our framework. The "masked" corresponds to the experiment of Section 4.3. The "1CH" corresponds to the experiment of Section 4.4.

Experiments	Mean VSD Recall		
	Seen Obj. 1–18	**Unseen Obj. 19–30**	**Total**
DCSPose-basic-1CH	63.05	61.62	62.34
DCSPose-basic	65.86	62.53	64.20
DCSPose-masked-1CH	65.54	63.89	67.22
DCSPose-masked	69.74	68.48	69.11

4.4. Experiment of Feature Extractor

As described in Section 3.4, the feature extractor adopts a dual-channel design, comprising two distinct ResNets dedicated to processing RGB and depth data separately. To validate the efficacy of employing two separate channels instead of a shared channel, a comparative experiment was devised. Specifically, an experimental group was established where the two channels of the feature extractor shared one ResNet during the training process. This is equivalent to having a single channel learn and extract both types of data features simultaneously. The experimental outcomes are presented in Table 3, where the suffix '1CH' represents the experimental group utilizing a shared channel. The results indicate that employing two independent convolutional extraction networks contributes to enhancing model performance. The distinct channels facilitate the network in learning different types of features due to the disparities between RGB and depth features.

4.5. Performance Analysis

Failure Cases. To explore potential shortcomings, we further conducted individual assessment tests for each object, and the results are presented in Table 4. While we observed that our framework achieved high scores on the majority of objects, the lower scores on a few objects did impact the overall performance. Upon analysis of the corresponding results and sample data, the primary reason for these lower scores can be attributed to the excessively cluttered scenes encountered in the test samples. Taking object 23 as an example, as depicted in Figure 8, the object was heavily occluded by other objects in the scene, which interfered with the assessment of color and depth. In some test cases, the structure of the object was almost entirely occluded, making the estimation of the object's pose very challenging. Additionally, we observed that when key structures of the object were obscured, as shown in Figure 9, the model struggled to accurately determine its precise pose.

Table 4. Separate evaluation for each individual object in the T-LESS dataset. While our method achieved promising results for the majority of objects, there were a few specific objects where the scores were comparatively lower.

Obj.	**#1**	**#2**	**#3**	**#4**	**#5**	**#6**	**#7**	**#8**	**#9**	**#10**	**#11**	**#12**	**#13**	**#14**	**#15**	**#16**	**#17**	**#18**
VSD	0.65	0.70	0.85	0.65	0.68	0.77	0.43	0.27	0.52	0.64	0.72	0.68	0.61	0.71	0.74	0.87	0.83	0.76
Obj.	**#19**	**#20**	**#21**	**#22**	**#23**	**#24**	**#25**	**#26**	**#27**	**#28**	**#29**	**#30**						
VSD	0.70	0.58	0.50	0.70	0.48	0.60	0.62	0.73	0.44	0.69	0.65	0.84						

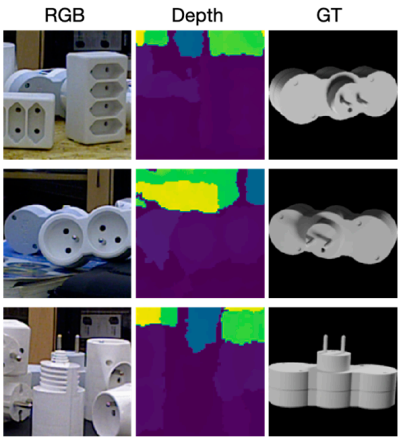


Figure 8. Visualization of failure cases of object #23. In some test cases, severe occlusion of objects not only disrupts color information but also renders depth information ineffective in providing meaningful support.

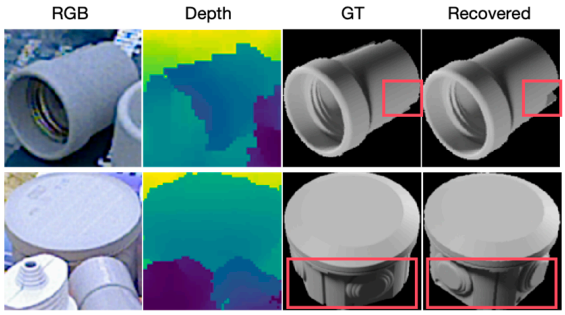


Figure 9. Visualization of failure cases. The occlusion of key structures results in the inability to accurately estimate the pose.

Time consumption. The time consumption of our method in template rendering, feature extraction, and online inference is presented in Table 5. The query and template images used here, including both RGB and depth maps, have the same resolution of 224×224. The data presented here were obtained through actual measurements on the hardware platform mentioned in Section 4.1. Thanks to the separation of offline and online processes, our framework achieves real-time processing capabilities suitable for practical applications.

Table 5. Time consumption for both offline and online processes.

Number of Templates	Offline		Online
	Templates Rendering	Features Extraction	Run-Time Inference
21,672	5.6 min	1.77 min	28 fps

5. Discussion

Conclusions. Our novel network architecture DCSPose, employing dual-channel Siamese networks, effectively leverages multi-source data to address the challenge of pose estimation for unseen texture-less objects in industrial settings. It exhibits significant improvements compared to existing state-of-the-art methods. By employing a Siamese network structure, we achieved the capability to match both real-world and simulated scenes, consequently enabling effective object pose estimation using the template matching approach. Through the dual-channel design, we made full use of RGB and depth features from real scenes, effectively addressing the issue of limited features on textureless objects. The offline template library construction allowed us to rapidly extend estimation capabilities to new objects, simultaneously alleviating the time pressure associated with run-time processing. Our framework effectively addresses three major challenges posed by object pose estimation tasks in industrial applications, including handling textureless targets, ensuring model generalization, and meeting real-time processing requirements.

Future research directions. The template matching approach is intuitive and practical for addressing real-world problems. However, it does come with certain limitations. In the process of creating the template set, the angular sampling interval has an impact on the minimal error in the final matching results. If the sampling intervals are too large, even when matching the most similar template, it can result in substantial errors. On one hand, one can attempt to add post-processing through interpolation, aiming to make discrete parameter values as continuous as possible. On the other hand, increasing template density to reduce pose intervals is an option, though it might require a trade-off between time and computational costs.

Through this work, we aim to drive the research in 3D pose detection towards practical applications, advancing the field of industrial automation. Simultaneously, we recognize the ongoing shortage of high-quality, large-scale 3D annotated datasets for industrial scenarios, which remains a pressing issue compared to 2D image datasets. Additionally, we observe that a considerable portion of current pose estimation research trends toward designing end-to-end network models. However, we believe that addressing real-world tasks can benefit from a multi-stage, modular design. Such an approach not only has the potential to reduce the demand for extensive 3D data annotation but also facilitates the system's rapid adaptation to various object types during practical applications.

Author Contributions: Investigation, Z.Y.; resources, L.L.; writing—original draft preparation, Z.Y.; writing—review and editing, Z.H., X.Y. and L.L.; visualization, Z.Y.; project administration, Z.H.; funding acquisition, L.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Shanghai Science and Technology Innovation Project (Grant No. XTCX-KJ-2023-75) and Science and Technology Service Network Initiative, Chinese Academy of Sciences (STS, Grant No. 20211600200102).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: We have created a brand-new set of simulation rendering templates for the T-LESS dataset, comprising RGB projections, depth maps, and mask images of 30 objects from 602 different viewpoints, totaling 54,180 images. For the convenience of future researchers, we have made this template data publicly available on Google Drive at https://drive.google.com/drive/folders/1cVBPewNpUIa6qF64wL6QenitSVDlfd0H?usp=sharing (accessed on 27 October 2023).

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

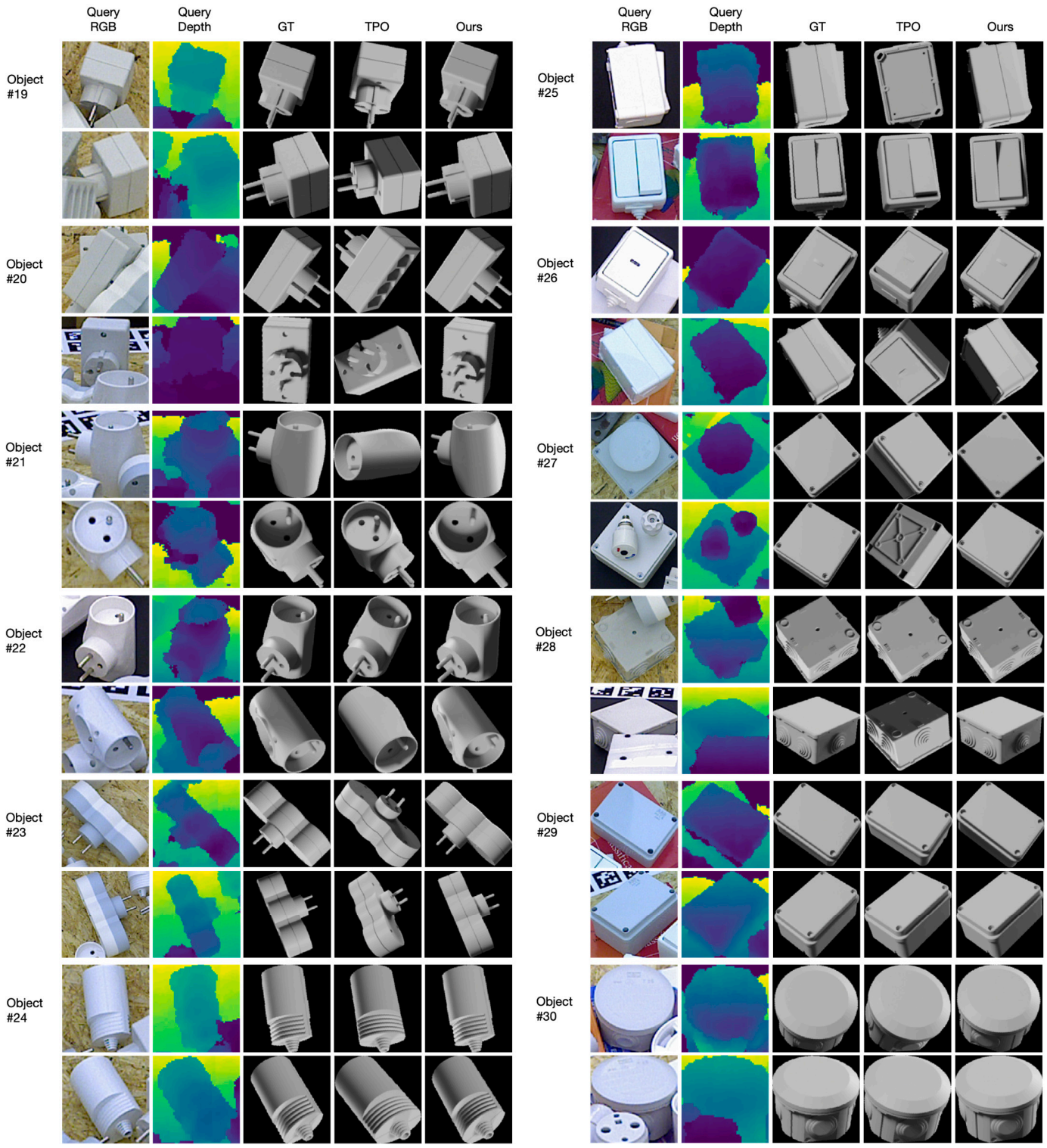


Figure A1. Qualitative comparison results on unseen objects. This figure illustrates the visual results of pose estimation for unseen objects. TPO [9] is the SOTA model for current template matching methods. Our method can achieve better results in many scenarios where TPO cannot be correctly estimated.

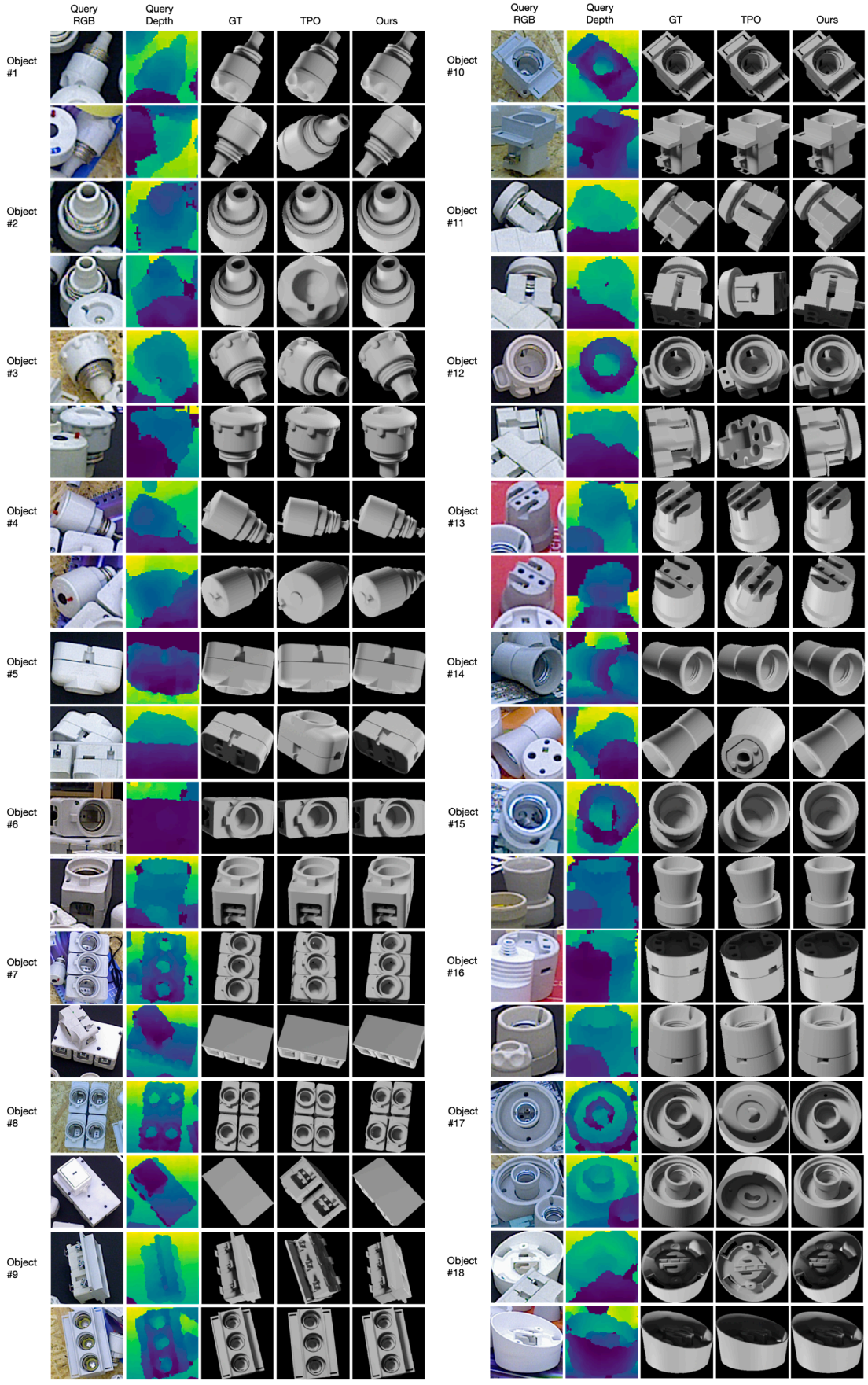


Figure A2. Qualitative comparison results on seen objects.

References

1. Hodan, T.; Michel, F.; Brachmann, E.; Kehl, W.; Buch, A.G.; Kraft, D.; Drost, B.; Vidal, J.; Ihrke, S.; Zabulis, X.; et al. BOP: Benchmark for 6D Object Pose Estimation. In Proceedings of the 15th European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 19–35.
2. Sundermeyer, M.; Hodaň, T.; Labbe, Y.; Wang, G.; Brachmann, E.; Drost, B.; Rother, C.; Matas, J. Bop challenge 2022 on detection, segmentation and pose estimation of specific rigid objects. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 2784–2793.
3. Du, G.; Wang, K.; Lian, S.; Zhao, K. Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: A review. *Artif. Intell. Rev.* **2021**, *54*, 1677–1734. [CrossRef]
4. Huang, Y.; Chen, Y. Autonomous driving with deep learning: A survey of state-of-art technologies. *arXiv* **2020**, arXiv:2006.06091.
5. Marchand, E.; Uchiyama, H.; Spindler, F. Pose estimation for augmented reality: A hands-on survey. *IEEE Trans. Vis. Comput. Graph.* **2015**, *22*, 2633–2651. [CrossRef] [PubMed]
6. He, Z.; Feng, W.; Zhao, X.; Lv, Y. 6D Pose Estimation of Objects: Recent Technologies and Challenges. *Appl. Sci.* **2021**, *11*, 228. [CrossRef]
7. Hoque, S.; Arafat, M.Y.; Xu, S.; Maiti, A.; Wei, Y. A Comprehensive Review on 3D Object Detection and 6D Pose Estimation With Deep Learning. *IEEE Access* **2021**, *9*, 143746–143770. [CrossRef]
8. Lugo, G.; Hajari, N.; Cheng, I. Semi-supervised learning approach for localization and pose estimation of texture-less objects in cluttered scenes. *Array* **2022**, *16*, 100247. [CrossRef]
9. Van Nguyen, N.; Hu, Y.; Xiao, Y.; Salzmann, M.; Lepetit, V. Templates for 3D Object Pose Estimation Revisited: Generalization to New Objects and Robustness to Occlusions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 6761–6770.
10. Zou, D.W.; Cao, Q.; Zhuang, Z.L.; Huang, H.Z.; Gao, R.Z.; Qin, W. An Improved Method for Model-Based Training, Detection and Pose Estimation of Texture-Less 3D Objects in Occlusion Scenes. In Proceedings of the 11th CIRP Conference on Industrial Product-Service Systems, Zhuhai, China, 29–31 May 2019; pp. 541–546.
11. Sundermeyer, M.; Durner, M.; Puang, E.Y.; Marton, Z.-C.; Vaskevicius, N.; Arras, K.O.; Triebel, R. Multi-path learning for object pose estimation across domains. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, Washington, USA, 14–19 June 2020; pp. 13916–13925.
12. Zhu, Y.; Li, M.; Yao, W.; Chen, C. A review of 6d object pose estimation. In Proceedings of the 2022 IEEE 10th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Chongqing, China, 17–19 June 2022; pp. 1647–1655.
13. Marullo, G.; Tanzi, L.; Piazzolla, P.; Vezzetti, E. 6D object position estimation from 2D images: A literature review. *Multimed. Tools Appl.* **2023**, *82*, 24605–24643. [CrossRef]
14. Wu, C.; Chen, L.; Wu, S. A Novel Metric-Learning-Based Method for Multi-Instance Textureless Objects' 6D Pose Estimation. *Appl. Sci.* **2021**, *11*, 10531. [CrossRef]
15. Chen, C.; Jiang, X.; Miao, S.; Zhou, W.; Liu, Y. Texture-Less Shiny Objects Grasping in a Single RGB Image Using Synthetic Training Data. *Appl. Sci.* **2022**, *12*, 6188. [CrossRef]
16. Liang, G.; Chen, F.; Liang, Y.; Feng, Y.; Wang, C.; Wu, X. A Manufacturing-Oriented Intelligent Vision System Based on Deep Neural Network for Object Recognition and 6D Pose Estimation. *Front. Neurorobot.* **2021**, *14*, 616775. [CrossRef]
17. Zhuang, C.; Zhao, H.; Li, S.; Ding, H. Pose prediction of textureless objects for robot bin picking with deep learning approach. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2023**, *237*, 449–464. [CrossRef]
18. Wang, J.W.; Li, C.L.; Chen, J.L.; Lee, J.J. Robot grasping in dense clutter via view-based experience transfer. *Int. J. Intell. Robot. Appl.* **2022**, *6*, 23–37. [CrossRef]
19. Xu, M.; Zhang, Z.; Gong, Y.; Poslad, S. Regression-Based Camera Pose Estimation through Multi-Level Local Features and Global Features. *Sensors* **2023**, *23*, 4063. [CrossRef]
20. Sun, J.; Ji, Y.-M.; Liu, S.-D. Dynamic Vehicle Pose Estimation with Heuristic L-Shape Fitting and Grid-Based Particle Filter. *Electronics* **2023**, *12*, 1903. [CrossRef]
21. Ju, J.; Zheng, H.; Li, C.; Li, X.; Liu, H.; Liu, T. AGCNNs: Attention-guided convolutional neural networks for infrared head pose estimation in assisted driving system. *Infrared Phys. Technol.* **2022**, *123*, 104146. [CrossRef]
22. Lee, T.; Jung, C.; Lee, K.; Seo, S. A study on recognizing multi-real world object and estimating 3D position in augmented reality. *J. Supercomput.* **2022**, *78*, 7509–7528. [CrossRef]
23. Zhang, S.; Zhao, W.; Peng, J.; Zhang, X.; Hu, Q.; Wang, J. Augmented reality museum display system based on object 6D pose estimation. *J. Northwest Univ. Nat. Sci. Ed.* **2021**, *51*, 816–823.
24. Wang, G.; Manhardt, F.; Tombari, F.; Ji, X.Y.; Ieee Comp, S.O.C. GDR-Net: Geometry-Guided Direct Regression Network for Monocular 6D Object Pose Estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 16606–16616.
25. Li, F.; Vutukur, S.R.; Yu, H.; Shugurov, I.; Busam, B.; Yang, S.; Ilic, S. Nerf-pose: A first-reconstruct-then-regress approach for weakly-supervised 6d object pose estimation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Vancouver, BC, Canada, 17–24 June 2023; pp. 2123–2133.

26. Sundermeyer, M.; Marton, Z.-C.; Durner, M.; Brucker, M.; Triebel, R. Implicit 3d orientation learning for 6d object detection from rgb images. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 699–715.
27. Konishi, Y.; Hattori, K.; Hashimoto, M. Real-time 6D object pose estimation on CPU. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 3451–3458.
28. Sundermeyer, M.; Marton, Z.-C.; Durner, M.; Triebel, R. Augmented Autoencoders: Implicit 3D Orientation Learning for 6D Object Detection. *Int. J. Comput. Vis.* **2020**, *128*, 714–729. [CrossRef]
29. Labbé, Y.; Manuelli, L.; Mousavian, A.; Tyree, S.; Birchfield, S.; Tremblay, J.; Carpentier, J.; Aubry, M.; Fox, D.; Sivic, J. MegaPose: 6D Pose Estimation of Novel Objects via Render & Compare. *arXiv* **2022**, arXiv:2212.06870.
30. Wu, J.; Wang, Y.; Xiong, R. Unseen Object Pose Estimation via Registration. In Proceedings of the 2021 IEEE International Conference on Real-Time Computing and Robotics (RCAR), Xining, China, 15–19 June 2021; pp. 974–979.
31. Pateraki, M.; Sapoutzoglou, P.; Lourakis, M. Crane Spreader Pose Estimation from a Single View. Available online: https://www.researchgate.net/profile/Manolis-Lourakis/publication/367051971_Crane_Spreader_Pose_Estimation_from_a_Single_View/links/63f3218151d7af05403c16ad/Crane-Spreader-Pose-Estimation-from-a-Single-View.pdf (accessed on 3 January 2024).
32. Yoon, Y.; DeSouza, G.N.; Kak, A.C. Real-time tracking and pose estimation for industrial objects using geometric features. In Proceedings of the 2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422), Taipei, Taiwan, 14–19 September 2003; pp. 3473–3478.
33. Chicco, D. Siamese neural networks: An overview. In *Artificial Neural Networks*; Humana: New York, NY, USA, 2021; pp. 73–94.
34. Guo, Q.; Feng, W.; Zhou, C.; Huang, R.; Wan, L.; Wang, S. Learning dynamic siamese network for visual object tracking. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1763–1771.
35. Melekhov, I.; Kannala, J.; Rahtu, E. Siamese network features for image matching. In Proceedings of the 2016 23rd international conference on pattern recognition (ICPR), Cancun, Mexico, 4–8 December 2016; pp. 378–383.
36. Peng, X.; Wang, K.; Zhu, Z.; Wang, M.; You, Y. Crafting better contrastive views for siamese representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 16031–16040.
37. Chen, X.; He, K. Exploring simple siamese representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 15750–15758.
38. Li, Y.; Chen, C.P.; Zhang, T. A survey on siamese network: Methodologies, applications, and opportunities. *IEEE Trans. Artif. Intell.* **2022**, *3*, 994–1014. [CrossRef]
39. Denninger, M.; Sundermeyer, M.; Winkelbauer, D.; Zidan, Y.; Olefir, D.; Elbadrawy, M.; Lodhi, A.; Katam, H. BlenderProc. *arXiv* **2019**, arXiv:1911.01911.
40. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
41. van den Oord, A.; Li, Y.; Vinyals, O. Representation Learning with Contrastive Predictive Coding. *arXiv* **2018**, arXiv:1807.03748.
42. Hodan, T.; Haluza, P.; Obdržálek, Š.; Matas, J.; Lourakis, M.; Zabulis, X. T-LESS: An RGB-D dataset for 6D pose estimation of texture-less objects. In Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, USA, 24–31 March 2017; pp. 880–888.
43. Kehl, W.; Manhardt, F.; Tombari, F.; Ilic, S.; Navab, N. SSD-6D: Making RGB-Based 3D Detection and 6D Pose Estimation Great Again. In Proceedings of the 16th IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 1530–1538.
44. Chen, X.; Fan, H.; Girshick, R.; He, K. Improved Baselines with Momentum Contrastive Learning. *arXiv* **2020**, arXiv:2003.04297. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.