# Learning Symmetry-Aware Geometry Correspondences for 6D Object Pose Estimation

Heng Zhao[1,4] Shenxing Wei[2] Dahu Shi[1,3*] Wenming Tan[1*]
Zheyang Li[1] Ye Ren[1] Xing Wei[2] Yi Yang[3] Shiliang Pu[1]
[1]Hikvision Research Institute [2]Xi'an Jiaotong University [3]Zhejiang University
[4]Key Laboratory of Peace-building Big Data of Zhejiang Province
{zhaoheng13,shidahu,tanwenming,lizheyang,renye,pushiliang}@hikvision.com
wsx1064432511@stu.xjtu.edu.cn, weixing@mail.xjtu.edu.cn, yangyics@zju.edu.cn

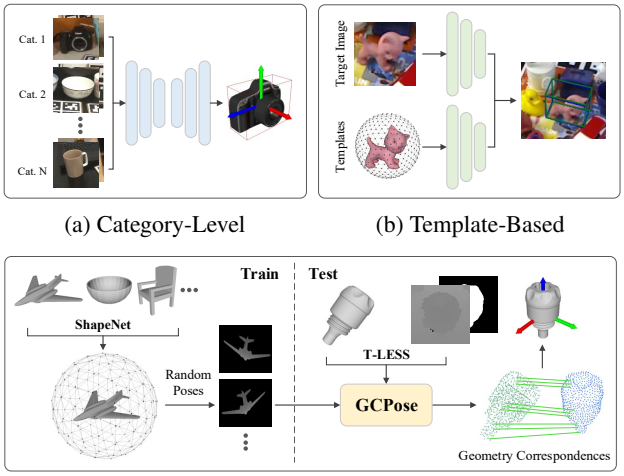
## Abstract

*Current 6D pose estimation methods focus on handling objects that are previously trained, which limits their applications in real dynamic world. To this end, we propose a geometry correspondence-based framework, termed GCPose, to estimate 6D pose of arbitrary unseen objects without any re-training. Specifically, the proposed method draws the idea from point cloud registration and resorts to object-agnostic geometry features to establish the 3D-3D correspondences between the object-scene point cloud and object-model point cloud. Then the 6D pose parameters are solved by a least-squares fitting algorithm. Taking the symmetry properties of objects into consideration, we design a symmetry-aware matching loss to facilitate the learning of dense point-wise geometry features and improve the performance considerably. Moreover, we introduce an online training data generation with special data augmentation and normalization to empower the network to learn diverse geometry prior. With training on synthetic objects from ShapeNet, our method outperforms previous approaches for unseen object pose estimation by a large margin on T-LESS, LINEMOD, Occluded-LINEMOD, and TUD-L datasets. Code is available at https://github.com/hikvision-research/GCPose.*

## 1. Introduction

The 6D pose of an object represents a geometry transformation between the object coordinate system and camera coordinate system, which consists of 3D rotation and 3D translation. Estimating object pose plays an important role in many real-world applications, such as robotic grasping [8] and augmented reality [38].

*Corresponding author.




(a) Category-Level (b) Template-Based

(c) Correspondence-Based

Figure 1: **Comparison of different open-set pose estimation frameworks.** Unlike previous methods that rely on category-specific network training (a) or struggle to estimate precise poses (b), our GCPose can perform well on arbitrary unseen objects (c).

Driven by the recent developments in deep learning, various methods have been proposed to explore the instance-level 6D pose estimation problem. Some existing works [43, 15] employ CNNs to detect a set of keypoints predefined on the 3D object model. Then the 6D pose can be solved by Perspective-n-Points (PnP) [31] or least-squares fitting algorithm. Since the detected keypoints serve as sparse correspondences, these methods often struggle when objects have view-point changes, occlusions, or lack of texture. Another alternative is to predict pixel-wise 3D coordinates for building dense 2D-3D correspondence maps [42, 59, 51, 36, 22]. They allow for significantly better treatment of occlusions and lead to more precise poses. Al-

though instance-level methods can obtain impressive results on existing benchmarks, they are still under the close-set assumption that the object space is identical in both training and testing phases. As a result, laborious data collection and re-training are required when unseen objects appear, which does not adhere to application in the real dynamic world.

To loosen the restriction with generalizability to unseen objects, category-level pose estimation paradigm [60, 33, 7, 35, 62] is proposed, as shown in Figure 1a. These approaches predict the object pose for previously seen or unseen objects from a known set of categories, but can not generalize to new instances having significantly different appearances or shapes [17, 53]. Another way to address the open-set problem is to resort to the template-based mechanism [55, 40, 5, 49, 30], which matches the input image to a series of templates generated from their 3D object models, as shown in Figure 1b. Obviously, these methods struggle to estimate precise poses due to occlusions and the limited number of viewpoints. To remedy this disadvantage, OVE6D [5] presents an in-plane orientation regression network and OSOP [49] introduces an extra network estimating dense 2D-2D correspondences between the input image and the matched template for pose refinement. In general, these methods have cascade pipelines and result in less accurate pose estimation for unseen objects.

As is known, point cloud registration methods have demonstrated excellent generalization to previously unseen point clouds as the geometry features of the point cloud are generic and object-agnostic. Inspired by this idea, we present an unseen object 6D pose estimation framework based on dense geometry correspondences, termed GCPose. Specifically, given the object-scene point cloud and object-model point cloud as input, GCPose can establish dense 3D-3D correspondences between them through the geometry features. Finding correspondences across two scene-level point clouds is well-studied in point cloud registration [1, 23, 65, 64, 45] field. However, it is challenging to learn 3D-3D matching for object-level point clouds due to ambiguous correspondences caused by symmetric properties in many object models. For instance, a 3D location in the object-scene point cloud may correspond to multiple 3D locations on the surface of the symmetric object, and vice versa. Therefore, exploiting the off-the-shelf point cloud registration method is sub-optimal to building correspondences for object pose estimation. To this end, we design a symmetry-aware matching loss to let the network learn the object symmetries explicitly.

Following the practice in OVE6D [5], we train the network using a large number of synthetic 3D object models from the ShapeNet [6] dataset. Specifically, we introduce a simple yet effective online training data generation with special data augmentation and normalization to empower the network to learn diverse geometry prior. After training on synthetic objects with varied shapes, our method is capable of generalizing to an arbitrary unseen object without any re-training. At inference time, the proposed method requires a depth image with a target object mask and the associated object CAD model, which are utilized to generate object-scene and object-model point clouds.

Our contribution can be summarized as follows:

1) We employ the point cloud registration framework and adapt it to work well for unseen object pose estimation.

2) A symmetry-aware matching loss is proposed for building unambiguous and robust 3D-3D correspondences, which improves the performance significantly.

3) GCPose achieves state-of-the-art performance on T-LESS, LineMOD, Occluded-LineMOD, and TUD-L datasets under an open-set pose estimation setting. Besides, the performance of GCPose can be further improved by scaling up the training data.

## 2. Related Work

### 2.1. 6D Object Pose Estimation

**Direct Pose Regression Approaches.** Some methods directly regress the 6D pose with deep neural networks, such as [63, 25]. Despite these approaches seeming simple, the non-linearity of the rotation space limits their generalization. Some post-refinement methods [58, 29, 24] are generally utilized to refine the pose iteratively.

**Correspondence-Based Approaches.** Another popular line to estimate the 6D pose relies on pre-defined keypoints on the object model. For instance, PVNet [43] selects $K$ 3D keypoints from the object surface. Then it employs a CNN to predict the offset to the projection of 3D keypoints in 2D images and the 6D pose is calculated by PnP [31]. PVN3D [16] and FFB6D [15] extract the per-point feature from an RGBD image to predict more precise offset to 3D keypoints and improve the performance. Since the keypoints serve as sparse correspondences, above methods often struggle in occluded scenes. An alternative is to generate dense correspondences. Some of the representative works include CDPN [34], EPOS [19], GDR-Net [59], and SO-Pose [11]. Recently, OnePose [53] and FS6D [17] focus on 6D pose estimation under a few-shot setting, which is more challenging.

**Template-Based Approaches.** To estimate the 6D pose of previously-unseen objects, the template-based mechanism [2, 41, 40, 5, 49] is proposed. These methods compare the input image to a series of templates, which are rendered images of objects associated with the corresponding 6D poses. And the pose of the best-matched template is selected as the final estimation result. Nevertheless, due to the discretization error of the limited number of viewpoints,

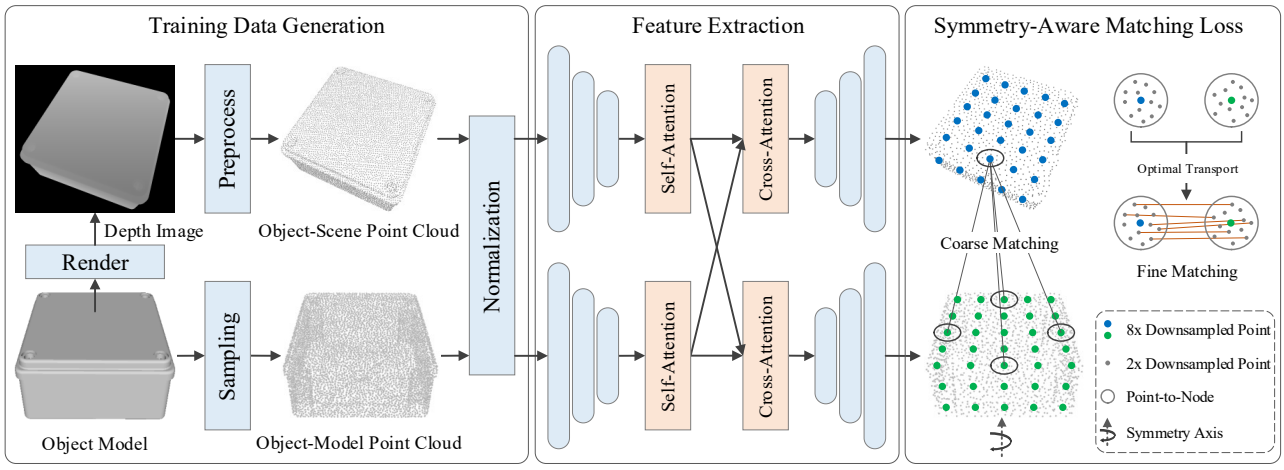



Figure 2: **The training pipeline of GCPose.** Firstly, the object-scene point cloud is obtained from the rendered depth image from ShapeNet [6] and the object-model point cloud is generated by uniform sampling on the surface of the object model. Then, a shared backbone network (*i.e.*, KPConv) is utilized to extract multi-level features of the input point clouds. The interleaved self- and cross-attention layers are adopted to enhance the geometry features of both coarse-level superpoints and fine-level points. Finally, a symmetry-aware matching loss is employed to supervise the learning of geometry features for establishing unambiguous and robust 3D-3D correspondences.

template-based approaches result in inaccurate pose estimation. OVE6D [5] presents an in-plane orientation regression network for rotation refinement. And OSOP [49] proposes an extra network estimating 2D-2D correspondences between the input image and the matched template for pose refinement.

### 2.2. Point Cloud Registration

Point cloud registration aims to align the two input point clouds by estimating a 3D rigid transformation. The typical pipeline contains detecting keypoints [32, 1], extracting feature descriptors [44, 10, 23], and estimating the transformation. Recently, detection-free methods[65, 64, 45] achieve state-of-the-art performance on several point cloud registration benchmarks. These methods generally assume a one-to-one relationship for two input point clouds, which goes against the nature of symmetry existing in object pose estimation. Specially, we propose a symmetry-aware matching loss to supervise the many-to-many relationship for object pose estimation and eliminate the ambiguity of correspondences.

### 2.3. Symmetries in Pose Estimation

Multiple poses can be inferred under a specific appearance for symmetric objects, which leads to the problem of pose ambiguity. To relieve this problem, PoseCNN [63], DenseFusion [58], and GDR-Net [59] adopt the ADD-S [63] metric as the loss during training. ES6D [39] designs a symmetry-invariant pose distance metric to make the network converge to the correct state. [37] predicts multiple poses for the object to estimate the specific pose distribution generated by symmetries. EPOS [19] and SurfEmb [13] learn the 2D-3D correspondence distributions over surface fragments or the entire surface and let the network learn the object symmetries implicitly. These methods focus on memorizing the symmetry information of specific objects and the trained network only performs well on a fixed set of objects. In this paper, we leverage the symmetries of objects and supervise the network with many-to-many correspondences in two input point clouds during training. Therefore, our network learns the object symmetries explicitly, which is proven effective for unseen object pose estimation.

## 3. Methodology

In this section, we first present the overall architecture of our GCPose. Next, the training data generation and feature extraction module are introduced. Then, we elaborate on the proposed symmetry-aware matching loss. At last, the inference pipeline of our framework is summarized.

### 3.1. Overview

Recent GeoTransformer [45] achieves outstanding performance in point cloud registration field and shows strong capabilities for finding correspondences. In this paper, we renovate the GeoTransformer [45] pipeline and adapt it for object pose estimation. Specifically, we design a symmetry-aware matching loss to eliminate the ambiguity of correspondences (Section 3.4). To make features more discriminative for establishing correspondence, we enrich fine-level point features with global structural cues (Section 3.3).

| Augmentation | Hyper-parameter |
|---|---|
| Downscale | scale range $s \in [0.1, 0.3]$ |
| Laplace Noise | scale param. $b \in [0, 0.004m]$ |
| Median Blur | kernel size $k = 3, 5$ |
| Upscale | scale range $s \in [3.3, 10]$ |
| Dropout | area ratio $r \in [0.01, 0.1]$ |

Table 1: **Data augmentation strategies.** We adopt these strategies to bridge the domain gap between the synthetic and real-world depth images.

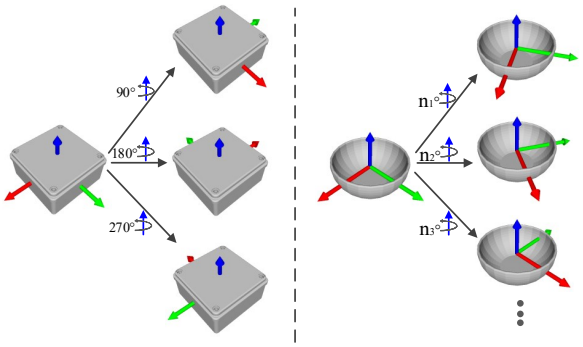
Moreover, an online training data generation with special data augmentation and normalization is devised to empower the network to learn diverse geometry priors (Section 3.2).

The training pipeline of GCPose is illustrated in Figure 2. Following OVE6D [5], our network is trained using rendered depth images based on 3D object models from ShapeNet [6]. The object-scene point cloud is obtained from the depth image within the object mask and the object-model point cloud is generated by uniform sampling on the surface of the object model. Given the object-scene and object-model point clouds as input, our goal is to estimate the rigid transformation $\mathbf{T}$ from the object coordinate system to the camera coordinate system, consisting of a 3D rotation $\mathbf{R} \in SO(3)$ and a translation vector $\mathbf{t} \in \mathbb{R}^3$.

Concretely, the KPConv-FPN [56] is employed as the backbone network to extract the multi-level features of the input point clouds. We refer to the coarsest level downsampled points ($\frac{1}{8}$ resolution of original point cloud) from the backbone as superpoints, denoted as $\hat{S}, \hat{M}$ for object-scene and object-model point clouds. The first level downsampled points ($\frac{1}{2}$ resolution of original point cloud), denoted as $S, M$ for object-scene and object-model point clouds, are treated as fine-level points. Besides, we also adopt interleaved self- and cross-attention layers [47, 66, 61, 48] to enhance the geometry features of both coarse-level superpoints and fine-level points. In the end, a symmetry-aware matching loss is proposed to supervise the learning of geometry features for establishing robust 3D-3D correspondences.

### 3.2. Training Data Generation

Our network is optimized using synthetic depth images rendered from 3D object models. Specifically, for each 3D object model, the ModernGL library is employed to synthesize the depth images under random poses. After that, we apply a set of data augmentation strategies to bridge the domain gap between the synthetic and real-world depth images. The depth image is first cropped based on the object mask and then resized to a fixed size (224 $\times$ 224 in this paper) for later preprocessing pipeline. Specifically, 1) the depth image is downsampled at a random scale, 2) Median blur is conducted on it to weaken high-frequency information of synthetic depth image, 3) random Laplace noise is added to simulate salt and pepper noise, 4) the depth image is zoomed into their original size, 5) random square areas are dropped from the image. The detailed hyper-parameters of these data augmentations can be found in Table 1. The resulting depth image with the object mask is converted into the object-scene point cloud using the camera intrinsic matrix. And the object-model point cloud is obtained by uniform surface sampling from the 3D object model.




(a) Discrete Symmetry (b) Continuous Symmetry

Figure 3: **Examples of symmetric objects.** (a) The object has finite ambiguity poses (4 in the case). (b) The object has infinite ambiguity poses.

In contrast with the training dataset where the set of objects' size is finite, the size of different objects in the real world varies widely. To improve generalization ability of our GCPose, the object-scene and object-model point clouds are normalized before being fed into the network. The normalization scale is calculated as the largest distance between any pair of the object-model point cloud, *i.e.*, the diameter of the smallest circumscribed sphere of the object-model point cloud.

### 3.3. Feature Extraction

The KPConv-FPN [56] only encodes local geometry features which are less distinctive for correspondence search. To capture the global geometry structures and strengthen superpoint features, a self-attention layer is applied to the coarse-level superpoint features extracted from the last downsample stage of KPConv-FPN. Then a cross-attention layer is adopted to enable the bidirectional communication between object-scene and object-model point clouds, which aims to model the inter-point-cloud geometry consistency. In contrast to GeoTransformer [45], to enrich fine-level point features, we inject coarse-level superpoint features with global structural cues into the upsampling stage of KPConv-FPN.

### 3.4. Symmetry-Aware Matching Loss

#### 3.4.1 Symmetries

Discrete and continuous symmetries are two types of symmetries. As shown in Figure 3, discrete symmetries can be described by a finite transformation set while continuous symmetries are described by an infinite transformation set. The symmetry property of an object can lead to multiple poses with the same visual appearance. In other words, there exist repeated geometry structures on the surface of a symmetry object. Therefore, the extracted geometry feature of a point in the object-scene point cloud should be similar to multiple points features in the symmetric object-model point cloud, and vice versa. To this end, we design a symmetry-aware matching loss to supervise the many-to-many relationship and let the network learn the object symmetries explicitly.

Specifically, we first use the method described in [39] to calculate the symmetry axes and angles for each object from ShapeNet. After that, we can obtain a set of transformation $\widehat{\mathbf{T}} = \{\mathbf{T}_0, \mathbf{T}_1, ..., \mathbf{T}_n\}, \mathbf{T}_i \in SE(3)$ for each object. The visual appearance of an object remains unchanged under each transformation of $\widehat{\mathbf{T}}$. Here we use $\mathbf{T}_0$ to represent the identity transformation and the set only containing $\mathbf{T}_0$ indicates a non-symmetrical object. For continuous symmetries objects, we discretize the rotation angle for obtaining a finite transformation set. Finally, the ground truth pose $\mathbf{T}$ of an object-scene point cloud can be extended to a pose set, which can be formulated as

$$\overline{\mathbf{T}} = \{\mathbf{T}\mathbf{T}_i \mid \mathbf{T}_i \in \widehat{\mathbf{T}}\}. \tag{1}$$

$\overline{\mathbf{T}}$ is then used to calculate the following matching loss.

#### 3.4.2 Matching Loss

Given the ground truth pose set $\overline{\mathbf{T}}$ and superpoints $\hat{S}, \hat{M}$, we calculate the overlap ratio for constructing positive and negative superpoint pairs between object-scene and object-model point clouds. For each superpoint pair $(\hat{S}_i, \hat{M}_j)$, the overlap ratio $o_{ij}$ is computed:

$$o_{ij} = \frac{o_{i\to j} + o_{j\to i}}{2} \tag{2}$$

$$o_{i\to j} = \frac{|\{p \in P_i^S | \exists q \in P_j^M, \overline{\mathbf{T}}_i \in \overline{\mathbf{T}} \text{ s.t.} \|\overline{\mathbf{T}}_i^{-1}(p) - q\| < \tau\}|}{|P_i^S|} \tag{3}$$

$$o_{j\to i} = \frac{|\{q \in P_j^M | \exists p \in P_i^S, \overline{\mathbf{T}}_i \in \overline{\mathbf{T}} \text{ s.t.} \|\overline{\mathbf{T}}_i(q) - p\| < \tau\}|}{|P_j^M|} \tag{4}$$

where $P_i^S, P_j^M$ are the set of fine points allocated to $\hat{S}_i, \hat{M}_j$ by point-to-node strategy [65], respectively. $\tau$ is the distance threshold. $\|\cdot\|$ is the Euclidean norm. $|\cdot|$ is the set cardinality. $\overline{\mathbf{T}}_i^{-1}$ is inverse transformation of $\overline{\mathbf{T}}_i$. The positive set $\varepsilon_p$ consists of all superpoint pairs with overlap ratio greater 0.1. Similarly, all superpoint pairs without overlap form the set of negatives $\varepsilon_n$. Taking $\varepsilon_p, \varepsilon_n$, and the associated overlap ratio as input, an overlap-aware circle loss [54, 45] is adopted as coarse supervision, denoted as $\mathcal{L}_c$.

Given a positive superpoint pair $\langle \hat{S}_k, \hat{M}_k \rangle$, an optimal transport layer is utilized to extract fine point correspondences. The set of their associated fine points $\langle P_k^S \in \mathbb{R}^{m\times 3}, P_k^M \in \mathbb{R}^{n\times 3} \rangle$ are allocated by point-to-node strategy.. Taking their features $\langle F_k^S \in \mathbb{R}^{m\times d}, F_k^M \in \mathbb{R}^{n\times d} \rangle$ as input, we firstly compute cost matrix $C^k \in \mathbb{R}^{m\times n}$ by inner product. Then Sinkhorn algorithm [50] is employed to obtain a refined cost matrix $\mathcal{Z}^k \in \mathbb{R}^{m\times n}$. Each entry $(i, j)$ in the matrix represents the matching confidence between the point $i$ and point $j$ from $P_k^S$ and $P_k^M$, respectively. If $(i, j)$ is a positive correspondence, the distance of the point $i$ and point $j$ under one pose of $\overline{\mathbf{T}}$ less than a matching threshold, and $(i, j)$ is a negative correspondence otherwise. Finally, the fine matching loss $\mathcal{L}_f^k$ is obtained by minimizing the negative log-likelihood of the cost matrix $\mathcal{Z}^k$ as in [46, 45].

We randomly sample $N_c$ superpoint pairs from $\varepsilon_p$ for fine matching loss. Formally, the overall loss function can be formulated as:

$$\mathcal{L} = \mathcal{L}_c + \frac{1}{N_c}\sum_{k=1}^{N_c} \mathcal{L}_f^k. \tag{5}$$

### 3.5. Inference

Following the setting of OVE6D [5], we obtain the class labels and segmentation masks of objects by the MaskR-CNN [14]. Given a depth image with mask and object model as input, the object-scene and object-model point clouds are first generated. After feature extraction, the coarse similarity matrix $\hat{C}$ is computed using the coarse features $\hat{F}^S, \hat{F}^M$.

$$\hat{c}_{i,j} = e^{-\|\hat{f}_i^S - \hat{f}_j^M\|}. \tag{6}$$

where $\hat{c}_{i,j} \in \hat{C}, \hat{f}_i^S \in \hat{F}^S, \hat{f}_j^M \in \hat{F}^M$. The largest K entries in $\hat{C}$ are selected as potential coarse correspondence candidates.

For each selected superpoint pair $\langle \hat{S}_k, \hat{M}_k \rangle$, the cost matrix $\mathcal{Z}^k \in \mathbb{R}^{m\times n}$ is computed as described in 3.4.2. To filter outlier matches, point correspondence candidates are selected by mutual nearest neighbor criteria as described in [52]. These selected dense point pairs are denoted as $\mathcal{M}^k = \{\langle p_u^S, p_v^M \rangle \mid p_u^S \in P_k^S, p_v^M \in P_k^M\}$. A hypothesis pose set $\{\mathbf{R}_k, \mathbf{t}_k \mid k \in [1, K]\}$ is computed:

$$\mathbf{R}_k, \mathbf{t}_k = \arg\min_{\mathbf{R},\mathbf{t}} \sum_{p_u^S, p_v^M \in \mathcal{M}^k} z_{u,v}^k \|\mathbf{R}p_v^M + \mathbf{t} - p_u^S\|. \tag{7}$$

where $z_{u,v}^k$ is the score from cost matrix $\mathcal{Z}^k$. The Eq. 7 can be solved by a weighted least-squares fitting algorithm [3]. Then, each hypothesis pose is verified by voting among

the entire point correspondences set $\mathcal{M} = \bigcup_{k=1}^{K} \mathcal{M}^k$. The hypothesis with the largest number of inlier points will be selected as the final estimation result:

$$\mathbf{R}, \mathbf{t} = \underset{\mathbf{R}_k, \mathbf{t}_k}{\arg\max} \sum_{p_u^S, p_v^M \in \mathcal{M}} \mathbb{I}( \|\mathbf{R}_k p_v^M + \mathbf{t}_k - p_u^S\| < \delta ). \tag{8}$$

where $\mathbb{I}$ represents the indicator function, $\delta$ is the voting threshold.

# 4. Experiment

## 4.1. Datasets and Metrics

**Datasets.** We evaluate our method on four public benchmark datasets: T-LESS [20], LINEMOD [18], Occluded-LINEMOD [4], and TUD-L [21].

**Metrics.** We follow the evaluation metric used in the BOP Challenge [21]. The performance of pose estimation is evaluated using three errors, *i.e.*, Visible Surface Discrepancy (VSD), Maximum Symmetry-aware Surface Distance (MSSD), and Maximum Symmetry-aware Projection Distance (MSPD). For each of the pose errors, an average recall is computed, *i.e.*, $AR_{VSD}$, $AR_{MSSD}$, $AR_{MSPD}$, based on a set of error thresholds. The AR refers to the average of the three recalls.

## 4.2. Implementation Details

We use a KPConv-FPN [56] backbone for feature extraction. Adam optimizer [26] is employed to optimize the network with the learning rate of $5 \times 10^{-5}$. The weight decay is set to $1 \times 10^{-6}$. GCPose is trained for 80 epochs with a batch size of 16 and the learning rate is decayed by 0.95 every 5 epochs. The network is trained on NVIDIA Tesla V100 GPUs.

Following the practice in OVE6D [5], our GCPose is trained using the synthetic 3D objects from ShapeNet [6]. During the test, we first obtain the class labels and segmentation masks of objects by the MaskRCNN [14]. Taking a depth image with mask and object model as input, GCPose is applied to estimate the 6D pose for the object. In addition, the performance using ground truth class labels and segmentation masks is also reported in this paper.

## 4.3. Comparison with the State-of-the-Art Methods

### 4.3.1 T-LESS Results

We firstly report pose estimation results on T-LESS dataset in Table 2 in terms of the AR metric. To facilitate comparison to previous works, we re-evaluate the pose estimation results of OVE6D [5] in AR metric using the official trained model. Our method achieves state-of-the-art performance under an open-set pose estimation setting. GCPose outperforms the recent learning-based OVE6D [5] by 15.6% without ICP refinement [67] and is also 13.3% higher than its ICP refinement counterpart. And GCPose also outperforms traditional HybridPPF [28] with ICP refinement by 2.4%. These results indicate that GCPose performs well on symmetric and texture-less objects. Moreover, given the ground truth masks, GCPose achieves 73.8% AR, which reveals great potential for performance improvement with better segmentation masks.

| Method | $AR_{VSD}$ | $AR_{MSSD}$ | $AR_{MSPD}$ | AR |
|---|---|---|---|---|
| DrostPPF† [12] | 0.375 | 0.478 | 0.480 | 0.444 |
| VidalPPF† [57] | 0.464 | 0.575 | 0.574 | 0.538 |
| HybridPPF† [28] | 0.580 | 0.689 | 0.696 | 0.655 |
| MegaPose [30] | - | - | - | 0.543 |
| OVE6D [5] | 0.521 | 0.511 | 0.538 | 0.523 |
| OVE6D† [5] | 0.513 | 0.561 | 0.565 | 0.546 |
| **GCPose** (Ours) | **0.643** | **0.691** | **0.702** | **0.679** |
| OVE6D* [5] | 0.624 | 0.592 | 0.626 | 0.614 |
| OVE6D*† [5] | 0.601 | 0.637 | 0.637 | 0.625 |
| **GCPose*** (Ours) | **0.714** | **0.739** | **0.762** | **0.738** |

Table 2: **Comparisons on `T-LESS` dataset.** †: the results are further refined by the ICP algorithm. *: using the ground truth label and mask of the object.

| Method | $AR_{VSD}$ | $AR_{MSSD}$ | $AR_{MSPD}$ | AR |
|---|---|---|---|---|
| DrostPPF† [12] | 0.678 | 0.786 | 0.789 | 0.751 |
| OVE6D [5] | 0.560 | 0.751 | 0.762 | 0.701 |
| OVE6D† [5] | 0.736 | 0.882 | 0.886 | 0.835 |
| **GCPose** (Ours) | **0.749** | **0.885** | **0.887** | **0.841** |
| MatchNorm* [9] | 0.319 | 0.490 | 0.529 | 0.446 |
| MatchNorm*† [9] | 0.616 | 0.680 | 0.737 | 0.678 |
| OVE6D* [5] | 0.794 | 0.886 | 0.902 | 0.860 |
| OVE6D*† [5] | 0.806 | 0.952 | 0.956 | 0.905 |
| **GCPose*** (Ours) | **0.869** | **0.965** | **0.974** | **0.936** |

Table 3: **Comparisons on `LINEMOD` dataset.** †: the results are further refined by the ICP algorithm. *: using the ground truth label and mask of the object.

### 4.3.2 LINEMOD and Occluded-LINEMOD Results

We report pose estimation results on LINEMOD and Occluded-LINEMOD, as shown in Table 3, 4. GCPose still achieves state-of-the-art performance under an open-set pose estimation setting. Specifically, GCPose surpasses OVE6D [5] by 14.0% on LINEMOD dataset and 15.6% on Occluded-LINEMOD dataset, respectively. GCPose outperforms the recent MatchNorm [9] by a large margin of 49.0% on LINEMOD dataset and 37.6% on Occluded-LINEMOD dataset, respectively. Moreover, our method is superior to OVE6D and MatchNorm, which both use ICP refinement. In contrast to previous works using handcrafted geometry features, GCPose also outperforms DrostPPF

| Method | $AR_{VSD}$ | $AR_{MSSD}$ | $AR_{MSPD}$ | AR |
|---|---|---|---|---|
| DrostPPF† [12] | 0.437 | 0.563 | 0.581 | 0.527 |
| VidalPPF† [57] | 0.473 | 0.625 | 0.647 | 0.582 |
| HybridPPF† [28] | 0.517 | 0.675 | 0.703 | 0.631 |
| MegaPose [30] | - | - | - | 0.583 |
| OVE6D [5] | 0.373 | 0.540 | 0.575 | 0.496 |
| OVE6D† [5] | 0.526 | 0.658 | 0.697 | 0.627 |
| **GCPose** (Ours) | **0.543** | **0.691** | **0.721** | **0.652** |
| MatchNorm* [9] | 0.263 | 0.384 | 0.450 | 0.365 |
| MatchNorm*† [9] | 0.478 | 0.542 | 0.612 | 0.544 |
| OVE6D* [5] | 0.539 | 0.640 | 0.703 | 0.627 |
| OVE6D*† [5] | 0.633 | 0.755 | 0.798 | 0.728 |
| **GCPose*** (Ours) | **0.656** | **0.768** | **0.799** | **0.741** |

Table 4: **Comparisons on `Occluded-LINEMOD` dataset.** †: the results are further refined by the ICP algorithm. *: using the ground truth label and mask of the object.

| Method | $AR_{VSD}$ | $AR_{MSSD}$ | $AR_{MSPD}$ | AR |
|---|---|---|---|---|
| DrostPPF† [12] | 0.741 | 0.793 | 0.791 | 0.775 |
| VidalPPF† [57] | 0.811 | 0.910 | 0.907 | 0.876 |
| HybridPPF† [28] | **0.872** | 0.945 | 0.944 | 0.920 |
| MegaPose [30] | - | - | - | 0.712 |
| **GCPose** (Ours) | 0.871 | **0.960** | **0.947** | **0.926** |
| MatchNorm* [9] | 0.700 | 0.853 | 0.852 | 0.801 |
| MatchNorm*† [9] | 0.859 | 0.914 | 0.935 | 0.903 |
| **GCPose*** (Ours) | **0.888** | **0.979** | **0.980** | **0.949** |

Table 5: **Comparisons on `TUD-L` dataset.** †: the results are further refined by the ICP algorithm. *: using the ground truth label and mask of the object.

[12] and HybridPPF [28] on LINDEMOD and Occluded-LINEMOD datasets. These results further demonstrate that GCPose has a strong generalization ability to unseen objects.

#### 4.3.3 TUD-L Results

The pose estimation results on TUD-L dataset are summarized in Table 5. GCPose outperforms the recent MatchNorm [9] by 14.8% without ICP refinement and is also 4.6% higher than its ICP refinement counterpart. Besides, GCPose is also superior to traditional HybridPPF [28] with ICP refinement, which further indicates the superiority of our learning-based geometry features.

#### 4.3.4 Scaling of Training Data

To investigate the potential of GCPose with more CAD models for training, we scale up the training set to both ShapeNet [6] and ABC [27]. ABC dataset consists of one million CAD models and each model is a collection of explicitly parametrized curves and surfaces. For training efficiency, we only pick the first 100k out of 1000k CAD models from ABC. The pose estimation results are reported in Table 6. GCPose achieves consistent performance improvement on four datasets after being trained with more CAD models.

| Dataset | Method | w/ ABC | $AR_{VSD}$ | $AR_{MSSD}$ | $AR_{MSPD}$ | AR |
|---|---|---|---|---|---|---|
| T-LESS [20] | GCPose | | 0.643 | 0.691 | 0.702 | 0.679 |
| | | ✓ | **0.681** | **0.735** | **0.757** | **0.724** |
| | GCPose* | | 0.714 | 0.739 | 0.762 | 0.738 |
| | | ✓ | **0.760** | **0.801** | **0.828** | **0.796** |
| LM [18] | GCPose | | 0.749 | 0.885 | 0.887 | 0.841 |
| | | ✓ | **0.763** | **0.907** | **0.908** | **0.859** |
| | GCPose* | | 0.869 | 0.965 | 0.974 | 0.936 |
| | | ✓ | **0.870** | **0.971** | **0.978** | **0.940** |
| O-LM [4] | GCPose | | 0.543 | 0.691 | 0.721 | 0.652 |
| | | ✓ | **0.557** | **0.731** | **0.764** | **0.684** |
| | GCPose* | | 0.656 | 0.768 | 0.799 | 0.741 |
| | | ✓ | **0.657** | **0.793** | **0.827** | **0.760** |
| TUD-L [21] | GCPose | | 0.871 | 0.960 | 0.947 | 0.926 |
| | | ✓ | **0.875** | **0.978** | **0.974** | **0.942** |
| | GCPose* | | 0.888 | 0.979 | 0.980 | 0.949 |
| | | ✓ | **0.904** | **0.981** | **0.985** | **0.957** |

Table 6: **Comparisons on `T-LESS`, `LINEMOD (LM)`, `Occluded-LINEMOD (O-LM)`, and `TUD-L` datasets by scaling up training datasets.** *: using the ground truth label and mask of the object.

#### 4.3.5 Visualization

To visualize the learned symmetry-aware geometry features, we reduce the dimension of point-wise features extracted by our backbone network using the t-SNE algorithm. Then the object-model point cloud is colorized according to the low-dimension features. As shown in Figure 5, the geometry feature embeddings among different parts of an object surface are distinguishable clearly and show meaningful similarity among the local structure of symmetry. It demonstrates that our method can predict reasonable geometry representations for unseen objects, which are employed to solve the 6D poses in this paper. We also give some qualitative results of GCPose on T-LESS and LINEMOD datasets, as shown in Figure 4. Compared to OVE6D [5], our method can generate more precise poses.

### 4.4. Ablation Study

We conduct several ablation experiments on the T-LESS dataset. To eliminate the effect of inaccurate segmentation, we use ground truth labels and masks.

**Data Augmentation and Normalization.** Our method is trained on synthetic data and tested on real-world data.

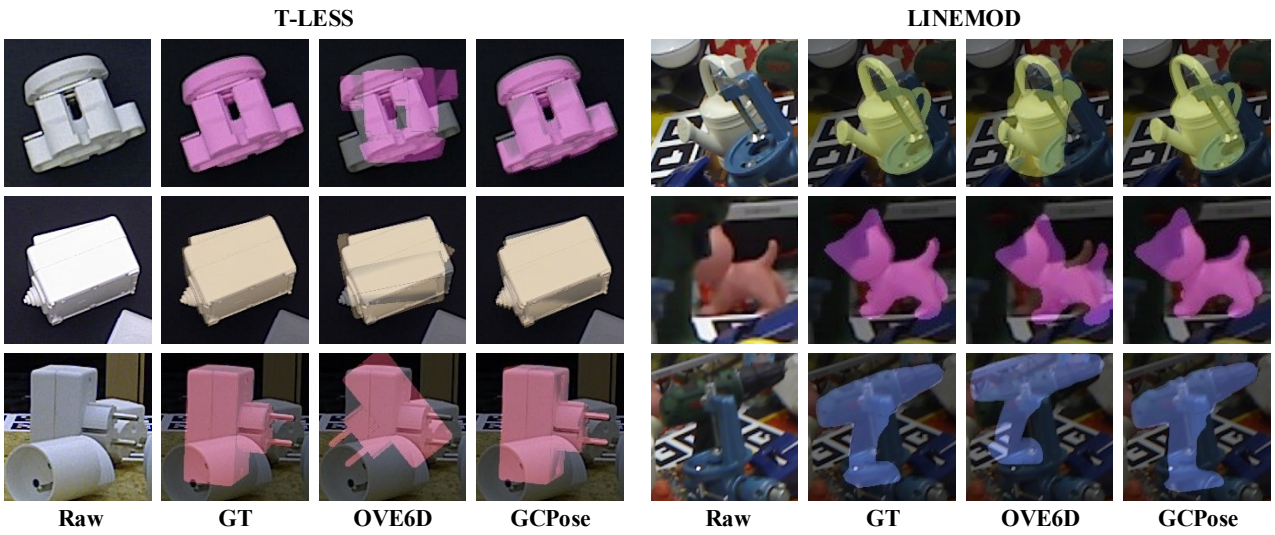



Figure 4: **Qualitative results.** We visualize the results of OVE6D [5] and the proposed GCPose on the T-LESS (left) and the LINEMOD (right) datasets. Raw indicates the original images and GT means the visualization of ground truth poses. GCPose performs well on a wide range of objects, containing view-point changes, and occlusions.

| Data Aug | Norm | GA Feat | SA Loss | AR |
|---|---|---|---|---|
| | | | | 0.635 |
| ✓ | | | | 0.669 (+3.4%) |
| ✓ | ✓ | | | 0.698 (+6.3%) |
| ✓ | ✓ | ✓ | | 0.708 (+7.3%) |
| ✓ | ✓ | ✓ | ✓ | **0.738** (+10.3%) |

Table 7: **Ablation experiments:** the impact of the proposed special data augmentation (Data Aug), normalization (Norm), global-aware fine-level features (GA Feat), and symmetry-aware matching loss (SA Loss).

So special data augmentation strategies are necessary to improve the generalization to real-world scenes. Besides, we standardize the scale of the input point clouds to facilitate the learning of geometry features. As shown in Table 7, special data augmentation and normalization bring 3.4% and 2.9% AR improvement, respectively.

**Global-Aware Fine-level Features.** As described in Section 3.3, we enhance fine-level point features with global structural cues to make features more discriminative for establishing correspondence. As shown in Table 7, the performance of GCPose is improved from 69.8% to 70.8% using global-aware fine-level features.

**Symmetry-Aware Matching Loss.** To eliminate the ambiguity of the learning of geometry features, as described in Section 3.4, we design a symmetry-aware matching loss to supervise the many-to-many correspondences. The geometry features output by GCPose are meaningful and the features of symmetric structures are visually similar (Figure 5). As shown in Table 7, the symmetry-aware matching loss provides 3.0% AR improvement.

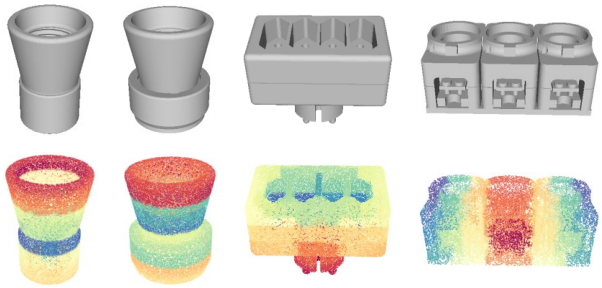


Figure 5: **t-SNE visualization of learned symmetry-aware geometry features on several unseen objects from T-LESS.** The geometry features are distinguishable among different parts within an object and show meaningful similarity among the local structure of symmetry.

## 5. Conclusion

In this paper, we propose a pose estimation framework based on object-agnostic geometry features, which can generalize to arbitrary unseen objects without re-training. Compared with previous methods, GCPose achieves state-of-the-art performance on four benchmarks. We hope that our simple yet effective framework could serve as a strong baseline for unseen object 6D pose estimation and motivate researchers to explore more in this direction.

**Acknowledgments.** This work is supported by National Key R&D Program of China under Grant No. 2023YFE0204200 and the Fundamental Research Funds for the Central Universities under Grant No. 226-2022-00051.

## References


[1] Xuyang Bai, Zixin Luo, Lei Zhou, Hongbo Fu, Long Quan, and Chiew-Lan Tai. D3feat: Joint learning of dense detection and description of 3d local features. In *CVPR*, 2020. 2, 3

[2] 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. 2

[3] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611. Spie, 1992. 5

[4] 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. 6, 7

[5] Dingding Cai, Janne Heikkilä, and Esa Rahtu. Ove6d: Object viewpoint encoding for depth-based 6d object pose estimation. In *CVPR*, 2022. 2, 3, 4, 5, 6, 7, 8

[6] 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. *arXiv:1512.03012*, 2015. 2, 3, 4, 6, 7

[7] Kai Chen and Qi Dou. Sgpa: Structure-guided prior adaptation for category-level 6d object pose estimation. In *ICCV*, 2021. 2

[8] Alvaro Collet, Manuel Martinez, and Siddhartha S Srinivasa. The moped framework: Object recognition and pose estimation for manipulation. *The international journal of robotics research*, 30(10), 2011. 1

[9] Zheng Dang, Wang Lizhou, Guo Yu, and Mathieu Salzmann. Learning-based point cloud registration for 6d object pose estimation in the real world. In *ECCV*, 2022. 6, 7

[10] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppfnet: Global context aware local features for robust 3d point matching. In *CVPR*, 2018. 3

[11] Yan Di, Fabian Manhardt, Gu Wang, Xiangyang Ji, Nassir Navab, and Federico Tombari. So-pose: Exploiting self-occlusion for direct 6d pose estimation. In *ICCV*, 2021. 2

[12] Bertram Drost, Markus Ulrich, Nassir Navab, and Slobodan Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *CVPR*, 2010. 6, 7

[13] Rasmus Laurvig Haugaard and Anders Glent Buch. Surfemb: Dense and continuous correspondence distributions for object pose estimation with learnt surface embeddings. In *CVPR*, 2022. 3

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 5, 6

[15] Yisheng He, Haibin Huang, Haoqiang Fan, Qifeng Chen, and Jian Sun. Ffb6d: A full flow bidirectional fusion network for 6d pose estimation. In *CVPR*, 2021. 1, 2

[16] Yisheng He, Wei Sun, Haibin Huang, Jianran Liu, Haoqiang Fan, and Jian Sun. Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation. In *CVPR*, 2020. 2

[17] Yisheng He, Yao Wang, Haoqiang Fan, Jian Sun, and Qifeng Chen. Fs6d: Few-shot 6d pose estimation of novel objects. In *CVPR*, 2022. 2

[18] Stefan Hinterstoisser, Stefan Holzer, Cedric Cagniart, Slobodan Ilic, Kurt Konolige, Nassir Navab, and Vincent Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In *ICCV*, 2011. 6, 7

[19] Tomas Hodan, Daniel Barath, and Jiri Matas. Epos: Estimating 6d pose of objects with symmetries. In *CVPR*, 2020. 2, 3

[20] Tomáš Hodan, Pavel Haluza, Štepán Obdržálek, Jiri Matas, Manolis Lourakis, and Xenophon Zabulis. T-less: An rgb-d dataset for 6d pose estimation of texture-less objects. In *WACV*, 2017. 6, 7

[21] 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. 6, 7

[22] Lin Huang, Tomas Hodan, Lingni Ma, Linguang Zhang, Luan Tran, Christopher Twigg, Po-Chen Wu, Junsong Yuan, Cem Keskin, and Robert Wang. Neural correspondence field for object pose estimation. In *ECCV*, 2022. 1

[23] Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. Predator: Registration of 3d point clouds with low overlap. In *CVPR*, 2021. 2, 3

[24] Shun Iwase, Xingyu Liu, Rawal Khirodkar, Rio Yokota, and Kris M Kitani. Repose: Fast 6d object pose refinement via deep texture rendering. In *ICCV*, 2021. 2

[25] Xiaoke Jiang, Donghai Li, Hao Chen, Ye Zheng, Rui Zhao, and Liwei Wu. Uni6d: A unified cnn framework without projection breakdown for 6d pose estimation. In *CVPR*, 2022. 2

[26] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015. 6

[27] Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. Abc: A big cad model dataset for geometric deep learning. In *CVPR*, 2019. 7

[28] Rebecca König and Bertram Drost. A hybrid approach for 6dof pose estimation. In *ECCV*, 2020. 6, 7

[29] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. Cosypose: Consistent multi-view multi-object 6d pose estimation. In *ECCV*, 2020. 2

[30] 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. *arXiv:2212.06870*, 2022. 2, 6, 7

[31] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o (n) solution to the pnp problem. *IJCV*, 2009. 1, 2

[32] Jiaxin Li and Gim Hee Lee. Usip: Unsupervised stable interest point detection from 3d point clouds. In *ICCV*, 2019. 3

[33] Xiaolong Li, He Wang, Li Yi, Leonidas J Guibas, A Lynn Abbott, and Shuran Song. Category-level articulated object pose estimation. In *CVPR*, 2020. 2

[34] 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. 2

[35] Jiehong Lin, Zewei Wei, Zhihao Li, Songcen Xu, Kui Jia, and Yuanqing Li. Dualposenet: Category-level 6d object pose and size estimation using dual pose network with refined learning of pose consistency. In *ICCV*, 2021. 2

[36] Lahav Lipson, Zachary Teed, Ankit Goyal, and Jia Deng. Coupled iterative refinement for 6d multi-object pose estimation. In *CVPR*, 2022. 1

[37] 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. 3

[38] Eric Marchand, Hideaki Uchiyama, and Fabien Spindler. Pose estimation for augmented reality: a hands-on survey. *IEEE transactions on visualization and computer graphics*, 22(12), 2015. 1

[39] Ningkai Mo, Wanshui Gan, Naoto Yokoya, and Shifeng Chen. Es6d: A computation efficient and symmetry-aware 6d pose regression framework. In *CVPR*, 2022. 3, 5

[40] 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. 2

[41] 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. 2

[42] Kiru Park, Timothy Patten, and Markus Vincze. Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation. In *ICCV*, 2019. 1

[43] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. Pvnet: Pixel-wise voting network for 6dof pose estimation. In *CVPR*, 2019. 1, 2

[44] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017. 3

[45] Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, and Kai Xu. Geometric transformer for fast and robust point cloud registration. In *CVPR*, 2022. 2, 3, 4, 5

[46] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 5

[47] Dahu Shi, Xing Wei, Liangqi Li, Ye Ren, and Wenming Tan. End-to-end multi-person pose estimation with transformers. In *CVPR*, 2022. 4

[48] Dahu Shi, Xing Wei, Xiaodong Yu, Wenming Tan, Ye Ren, and Shiliang Pu. Inspose: instance-aware networks for single-stage multi-person pose estimation. In *ACM MM*, 2021. 4

[49] Ivan Shugurov, Fu Li, Benjamin Busam, and Slobodan Ilic. Osop: A multi-stage one shot object pose estimation framework. In *CVPR*, 2022. 2, 3

[50] Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2), 1967. 5

[51] Yongzhi Su, Mahdi Saleh, Torben Fetzer, Jason Rambach, Nassir Navab, Benjamin Busam, Didier Stricker, and Federico Tombari. Zebrapose: Coarse to fine surface encoding for 6dof object pose estimation. In *CVPR*, 2022. 1

[52] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *CVPR*, 2021. 5

[53] 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. 2

[54] Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. Circle loss: A unified perspective of pair similarity optimization. In *CVPR*, 2020. 5

[55] 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. 2

[56] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *ICCV*, 2019. 4, 6

[57] 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. *Sensors*, 18(8), 2018. 6, 7

[58] Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martín-Martín, Cewu Lu, Li Fei-Fei, and Silvio Savarese. Densefusion: 6d object pose estimation by iterative dense fusion. In *CVPR*, 2019. 2, 3

[59] 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. 1, 2, 3

[60] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *CVPR*, 2019. 2

[61] Xing Wei, Yifan Bai, Yongchao Zheng, Dahu Shi, and Yihong Gong. Autoregressive visual tracking. In *CVPR*, 2023. 4

[62] Yijia Weng, He Wang, Qiang Zhou, Yuzhe Qin, Yueqi Duan, Qingnan Fan, Baoquan Chen, Hao Su, and Leonidas J Guibas. Captra: Category-level pose tracking for rigid and articulated objects from point clouds. In *ICCV*, 2021. 2

[63] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv:1711.00199*, 2017. 2, 3

[64] Zi Jian Yew and Gim Hee Lee. Regtr: End-to-end point cloud correspondences with transformers. In *CVPR*, 2022. 2, 3

[65] Hao Yu, Fu Li, Mahdi Saleh, Benjamin Busam, and Slobodan Ilic. Cofinet: Reliable coarse-to-fine correspondences for robust pointcloud registration. *NeurIPS*, 2021. 2, 3, 5

[66] Xiaodong Yu, Dahu Shi, Xing Wei, Ye Ren, Tingqun Ye, and Wenming Tan. Soit: Segmenting objects with instance-aware transformers. In *AAAI*, 2022. 4

[67] Zhengyou Zhang. Iterative point matching for registration of free-form curves and surfaces. *IJCV*, 1994. 6