

# StablePose: Learning 6D Object Poses from Geometrically Stable Patches

Yifei Shi\* Junwen Huang\* Xin Xu Yifan Zhang Kai Xu<sup>†</sup>  
National University of Defense Technology

## Abstract

We introduce the concept of geometric stability to the problem of 6D object pose estimation and propose to learn pose inference based on geometrically stable patches extracted from observed 3D point clouds. According to the theory of geometric stability analysis, a minimal set of three planar/cylindrical patches are geometrically stable and determine the full 6DoFs of the object pose. We train a deep neural network to regress 6D object pose based on geometrically stable patch groups via learning both intra-patch geometric features and inter-patch contextual features. A sub-network is jointly trained to predict per-patch poses. This auxiliary task is a relaxation of the group pose prediction: A single patch cannot determine the full 6DoFs but is able to improve pose accuracy in its corresponding DoFs. Working with patch groups makes our method generalize well for random occlusion and unseen instances. The method is easily amenable to resolve symmetry ambiguities. Our method achieves the state-of-the-art results on public benchmarks compared not only to depth-only but also to RGBD methods. It also performs well in category-level pose estimation.

## 1. Introduction

The problem of object pose estimation is to determine the 6D rigid transformation from the local object coordinate system to the camera reference frame. Robust and accurate object pose estimation is of primary importance in a variety of applications ranging from robotic manipulation and localization to augmented reality. Recent advances either predict correspondences between observations and template models [36], or regress pose directly [49]. In these tasks, RGB features learned with convolutional neural networks have been predominantly adopted with notable success [17].

Object pose inference with only color information, however, find difficulty in handling texture-less objects or unseen surface texture/appearance. In human perception, object pose hinges on object geometry [42]. Humans recognize shapes and their poses simultaneously and in a coupled way [11] in order to achieve a so-called invariant object

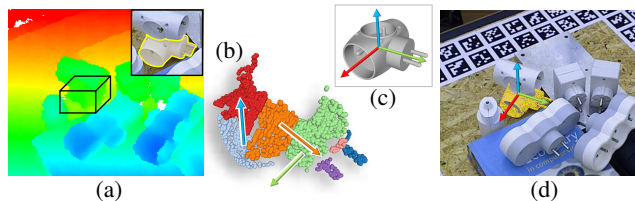


Figure 1: Given the 3D point cloud (b) of a detected object (a), StablePose is trained to predict its 6D pose based on a geometrically stable patch group containing the blue, the orange and the green patches. Each patch determines a few DoFs and they altogether pin down all six DoFs (c, d).

recognition [23]. Geometry enables a natural and powerful pose perception not only mitigating the distraction of color and appearance but also facilitating generalization to random occlusion and unseen instances.

3D geometric information does have been utilized for pose inference, both in traditional [9, 44] and in learning-based methods [52, 6], especially with the proliferation of depth cameras. The most straightforward use is to perform ICP-based pose refinement with the geometric information [40, 30]. Most deep learning approaches learn depth features to enhance color features [46]. Some others infer object poses from geometric features learned on 3D point clouds [12] or voxels [45]. These geometric features, however, are learned without an explicit guidance on the correlation between shape and pose, making pose reasoning based on them lack of interpretability and weak in generality.

We propose to learn object pose inference based on 3D surface patches extracted from the point cloud of a single-view depth image. In particular, we focus on planar and cylindrical patches which are omnipresent on the surface of household objects. This design choice stems from two key insights. *First*, patches are neither too local to capture meaningful geometric information, nor too global to be repeatable and generalizable across object instances. *Second*, each patch determines a specific set of DoFs of object pose. A minimal set of *geometrically stable patches* can lock all six DoFs according to the theory of geometric stability (or slippage) analysis [13]. It is therefore possible to accurately reason about 6D object poses over a small group of geometrically stable patches (Figure 1). Each stable group usually

\*Joint first authors

<sup>†</sup>Corresponding author: kevin.kai.xu@gmail.com

contains up to three patches, which facilitates fast learning. This also enables pose prediction with a redundant set of the stable groups, leading to robust pose estimation generalizing under occlusion and to unseen objects.

We design StablePose, a deep neural network trained to regress 6D object pose based on geometrically stable patch groups. Given a patch-sample 3D point cloud, the network extracts both intra-patch geometric features and inter-patch contextual features. It then predicts a 6D pose for each stable patch group through aggregating the intra- and inter-patch features. A dense *point-to-point* pose loss is used to train this network. A crucial design of StablePose is that a subnetwork is trained to predict per-patch poses. This auxiliary task is a relaxation of the group pose prediction since a single patch cannot determine the full 6DoFs and thus the pose loss downgrades to a weaker *point-to-patch* loss. Imposing such weak constraint for each patch in a stable group individually improves pose accuracy in the respective DoFs and altogether reinforces the group pose constraint, similar in spirit to the principle of geometrically stable ICP [14].

Given a 3D point cloud, we first extract a set of geometrically stable patch groups via performing stability analysis. We then use StablePose to predict a 6D pose for each stable group. The final object pose takes the average of all group poses weighted by group stability. To resolve ambiguities introduced by symmetries and achieve high pose accuracy, StablePose is trained to handle asymmetric objects and objects with discrete and continuous symmetries separately. Through extensive evaluation, we show that StablePose outperforms state-of-the-art learning-based methods by 19.1% for depth-only input and 9.1% for RGBD input on the T-LESS benchmark [18]. Furthermore, our method generalizes well for category-level pose estimation, obtaining competitive results on the NOCS-REAL275 benchmark [48] and 18.6% improvement on a more challenging ShapeNet based dataset. Our work makes the following contributions:

- We, for the first time, introduce the concept of geometric stability into 6D object pose estimation.
- We propose a deep network learning to infer 6D object pose based on geometrically stable patch groups. It attains high accuracy and robustness with a weak task of patch-wise, under-determined pose estimation.
- We devise several key designs to accommodate a broad range of cases encompassing asymmetric or symmetric objects, objects with occlusion and unseen objects.

## 2. Related work

**Pose estimation from RGB** The most common solution to object pose estimation from RGB images is to detect and match keypoints and solve a PnP. This approach has been well studied with a huge body of learned or non-learned

methods (e.g., [36, 43, 32, 51, 39] and a survey [10]). They are, however, less capable in the texture-less case where keypoints are hard to detect.

More recent works focus on predicting 6D object pose directly with trained deep neural networks [49]. Li et al. [28] proposed a method which matches the rendered images of the object model against the input image for pose refinement. Similar idea was later explored in many follow-up works [27, 47, 31]. SSD-6D [24] and Deep-6DPose [8] integrate object detection, segmentation and pose regression from single-view RGB images in a unified network. Instead of directly regressing poses, another line of works learns to output interim results of pixel-wise 3D coordinates [2, 48, 17, 21], based on which object poses can then be recovered. Note, however, EPOS [17] samples 3D patches from *template models* which are used as spatial clusters for 3D coordinates regression of image pixels. In contrast, our patches are sampled from *observation point clouds* and serve as DoF determinant of 6D object pose.

**Pose estimation with RGB-D** The most straightforward use of depth is to perform pose refinement with ICP-based geometric alignment [40, 30] or by congruent set based registration [29]. A more sophisticated approach is to utilize depth for 2D-3D feature fusion [46]. 3D-keypoint-based approaches compute object poses by solving a least-square optimization to match detected 3D key-points and their counterparts of the template model [41, 15]. Point pair feature (PPF) based methods [9, 44, 50] achieve high accuracy at the cost of high computation complexity.

The most relevant work to ours is [25] where a convolutional auto-encoder is trained to regress descriptors of locally-sampled RGB-D patches for 6D vote casting. During testing, scene patch descriptors are matched against model view patches and cast 6D object votes which are then filtered to refined hypotheses. While their method learns *patch matching*, our method tries to learn *pose regression* from geometrically stable patch groups.

**Stability analysis** Stability or slippage analysis is a powerful tool of shape analysis. It is originally posed to subsample points from a point set while maintaining the stability of DoFs in ICP-based alignment [14, 33, 3]. This is done by filtering out redundant points while keeping sufficient points for each alignment DoF. Another application of stability analysis is to extract slippage signatures for discovering slippable components on a 3D object [13, 1]. Inspired by these works, our work, for the first time, introduces the concept of stability into object pose estimation.

## 3. Stability Analysis for Pose Estimation

We describe stability analysis of 3D shapes, also known as slippage analysis, and investigate its relation to object pose estimation. The geometric stability of a 3D shape can

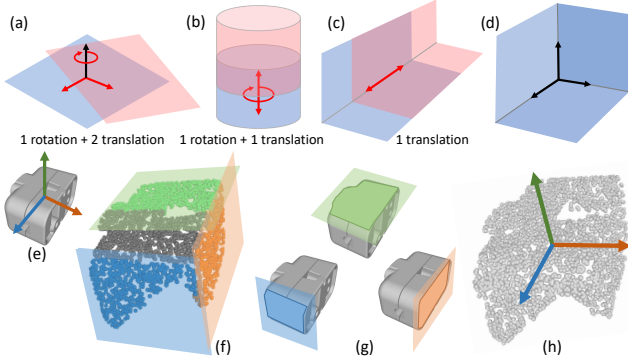


Figure 2: (a-d): Geometric stability of different shapes. The unstable/slippable transformations of each shape are annotated with red arrows. A group of three non-coplanar planes (d) is geometrically stable under rigid transformation. (e-h): Using geometric stability for object pose estimation. Given the observation of an object (f), a geometrically stable patch group is extracted and matched to the model shape (g), thus determining the 6D pose of the object (h).

be characterized by the rigid transformations (a translation and a rotation) that minimizes point-to-plane error metric. Intuitively, if a 3D shape is transformed by a rigid transformation without introducing significant motion along the normal direction at each surface point, it is called geometrically unstable (or slippable) along that transformation. If there is no such transformation, the shape is stable. See Figure 2(a-d) for a few examples with different stability.

Given a 3D shape sampled into a point set, we can compute a stability measure base on the eigenvalues of the  $6 \times 6$  covariance matrix of the 6D rigid transformation minimizing point-to-plane error. *Please refer to the appendix in the supplemental material for details of this computation.*

**Stability and object pose** Given the geometric observation of an object, we solve the problem of 6D object pose estimation on the basis of stability analysis. Specifically, we first extract a group of planar or cylindrical patches which are geometrically stable. The patches in the stable group are then aligned to the corresponding patches on the 3D model in canonical pose. Each alignment determines a subset of the six DoFs. All patches in the stable groups together pin down the 6D pose of the object. See Figure 2(e-h) for an illustration. In practice, however, finding patch correspondence itself is a challenging problem. In what follows, we propose a deep neural network which predicts 6D object poses without relying on patch correspondences.

## 4. Method

**Overview** Figure 3 provides an overview of our method. The input to our method is an RGBD image capturing one or multiple objects. The RGB image is used only for object detection but *not* for pose estimation. The output is the

6DoF pose of each object. Our method starts from detecting objects and predicting their 2D masks in the depth image. For each detected object, we obtain a 3D point cloud by unprojecting its depth mask. We then extract planar and cylindrical patches from the point cloud (Section 4.1) and sample a set of geometrically stable patch groups based on stability analysis (Section 4.2). Each stable group is fed into a deep network to predict the 6D pose of the corresponding object and the final pose is obtained by averaging the 6D poses obtained with all stable groups (Section 4.3). To resolve the ambiguities caused by symmetry, our network handles asymmetric objects and objects with discrete and continuous symmetries separately, each trained with a proper loss.

### 4.1. Object Detection and Patch Extraction

The first step is to detect objects and extract patches. Any RGB-based object detection method can be used here. We adopt the detector proposed in Pix2Pose [30] which is also used in many recent pose estimation works. We then crop the depth image based on the detected object mask and unproject the cropped depth image into a 3D point cloud. For each object point cloud, we extract both planar patches  $\mathcal{P}^P = \{p_i^P\}$  and cylindrical patches  $\mathcal{P}^C = \{p_i^C\}$ . Let  $\mathcal{P} = \mathcal{P}^P \cup \mathcal{P}^C$ . Again, many existing methods can be utilized to extract patches from point clouds. We found through experiment that CAPE [34] is fast and relatively robust for our data modality (i.e., point cloud converted from single-view depth image of objects with occlusion).

### 4.2. Stable Patch Group Sampling

One could predict object pose based on all patches. Ideally, the more patches are used, the more global information is encoded and the better the pose can be estimated [25]. In the case of single-view observations, however, using all patches may not be a good choice since the patch count varies from different views due to occlusion. Based on the fact that 6D object pose can be determined with a minimum of three geometrically stable patches (e.g., nearly mutual orthogonal planar patches in Figure 2), we opt for learning pose estimation based on *patch triplets*.

Given an object point cloud, we enumerate all triplets out of  $\mathcal{P}$  as patch group candidates. We then analyze the stability of each group. We collect those groups whose stability measure passes the threshold into the set of geometrically stable patch groups  $\mathcal{G} = \{G_k = (p_{k1}, p_{k2}, p_{k3})\}$ . To enhance the generality of our pose estimation network, we train it with patch groups sampled from point clouds in multiple views for each object.

### 4.3. Object Pose Estimation

#### 4.3.1 Patch Feature Extraction

**Intra-patch geometric feature** To learn geometric feature for each individual patch, we utilize PointNet++ [35].

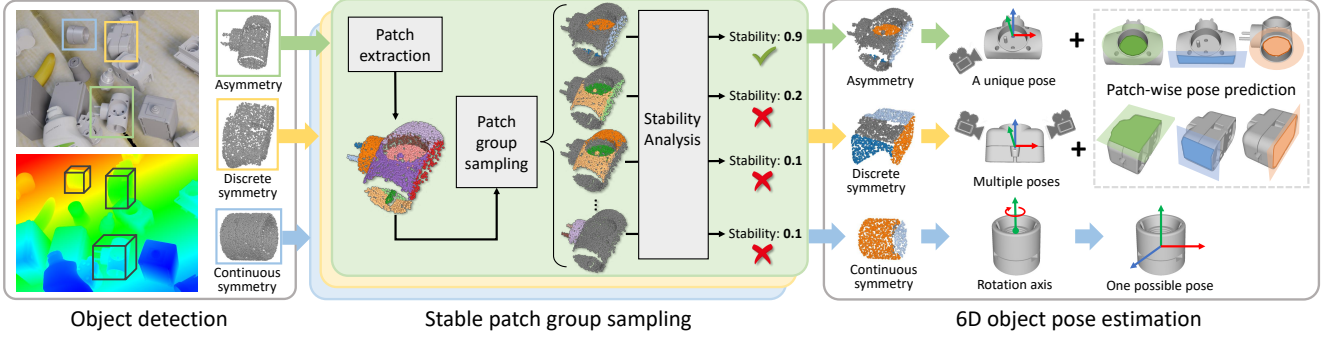


Figure 3: Method overview. Given a single-view RGB-D image, we first detect and segment objects in the RGB image and then crop the depth point cloud. Based on object labels, the objects are categorized according to symmetry property. For each object point cloud, we extract planar and cylindrical patches and sample a set of geometrically stable patch groups based on stability analysis. Our network predicts for stable group a 6D pose, with patch-wise pose estimation as an auxiliary task. For objects with continuous symmetry, it computes the rotation axis and then outputs one possible 6D pose.

The input is the point cloud of a patch (resampled to 2000 points) and the output is a feature vector of 1024D. Note that this learned feature encodes for a patch not only its local geometry but also its relative position in the whole object. The latter benefits spatial reasoning in pose estimation.

**Inter-patch contextual feature** To help our network reasoning about object pose globally, we additionally learn inter-patch contextual features using all patches. Since we have extracted per-patch features, we concatenate the per-patch features for every two patches and then aggregate them using the Relation Network [37]. Relation network provides a mechanism for non-local message passing between patches. The final feature is the adaptive pooling of the pair-wise features after message passing, which encodes inter-patch structure and global shape context.

#### 4.3.2 Pose Estimation of Asymmetric Objects

**Group pose prediction** Given an object, its 6D pose is predicted on the basis of its geometrically stable patch groups  $\mathcal{G}$ . As shown in Figure 5, for each  $G_k$  in  $\mathcal{G}$ , a three-layer multi-layer perceptrons (MLPs) takes the concatenation of its per-patch geometric features and the inter-patch contextual feature as input and outputs a 6D pose. To train the network, we impose a *dense-point pose loss* which measures the discrepancy between the predicted 6D pose and the ground-truth on a per-point basis:

$$\mathcal{L}_k^{\text{group}} = \sum_j \|\mathbf{T}_k^G \mathbf{x}_j^m - \bar{\mathbf{T}} \mathbf{x}_j^m\|, \quad (1)$$

where  $\mathbf{T}_k^G = [\mathbf{R}_k^G | \mathbf{t}_k^G]$  is the predicted pose for group  $G_k$  and  $\bar{\mathbf{T}} = [\bar{\mathbf{R}} | \bar{\mathbf{t}}]$  the ground-truth object pose.  $\mathcal{X}^m = \{\mathbf{x}_j^m\}$  are the points sampled on the template 3D model.

**Patch pose prediction** In addition to the group pose prediction, we introduce an auxiliary task of patch-wise pose

prediction. This task is a relaxation of the group pose prediction since a single patch cannot determine all six DoFs and thus the point-to-point loss in Eq. (1) needs to be replaced by a weaker “point-to-patch” loss. For a stable patch group, imposing such weak constraint for each of its three patches individually improves the pose accuracy in respective DoFs and altogether reinforces the group pose constraint (see Figure 4). This process resembles the stability-based ICP for 3D shape registration [38]. Further, these weak tasks decouple the learning into different sets of DoFs, making the network easier to train with faster convergence.

For each patch  $p_i \in \mathcal{P}$ , we devise a three-layer MLP which takes the concatenation of its geometric feature and the contextual feature as input and produces a patch pose  $\mathbf{T}_i^P = [\mathbf{R}_i^P | \mathbf{t}_i^P]$ . The network is trained by minimizing a *point-to-plane loss* for planar patches or a *point-to-axis loss* for cylindrical patches. In particular, we define the dense point-to-plane loss for planar patches as

$$\mathcal{L}_i^{\text{planar}} = \sum_j \|[(\mathbf{T}_i^P)^{-1} \mathbf{x}_{ij} - (\bar{\mathbf{T}})^{-1} \mathbf{c}_i] \cdot (\bar{\mathbf{R}}^{-1} \mathbf{n}_i)\|, \quad (2)$$

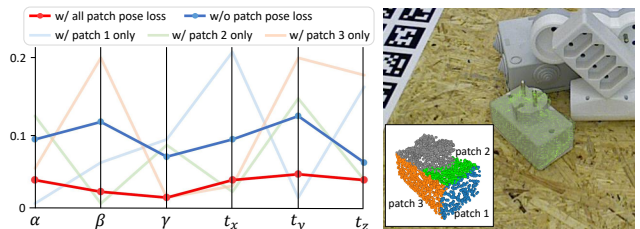


Figure 4: A parallel coordinates visualization [22] of object pose errors in the six DoFs. Minimizing the pose loss of one patch optimizes a specific subset of DoFs (see the curves with light shading). Minimizing the group pose loss but not per-patch pose loss leads to high overall pose error (the blue curve). The best overall accuracy is obtained by minimizing both group and all-patch pose losses (the red curve).

where  $\mathbf{c}_i$  represents the center of patch  $p_i$ , and  $\mathbf{n}_i$  is the normal of the plane that  $p_i$  lies in.  $\mathbf{x}_{ij}$  is a point on  $p_i$ . For cylindrical patches, point-to-patch error can be reduced to point-to-axis error and the corresponding patch pose loss is

$$\mathcal{L}_i^{\text{cylind}} = \sum_j \|d((\mathbf{T}_i^{\text{P}})^{-1} \mathbf{x}_{ij}, \bar{\phi}_i) - r_i\|, \quad (3)$$

where  $\bar{\phi}_i$  is axis of the patch cylinder transformed with  $\bar{\mathbf{T}}^{-1}$  and  $r_i$  the radius of the cylinder.  $d(\mathbf{x}, \phi)$  denotes the distance between point  $\mathbf{x}$  and line  $\phi$ .

**The overall loss** The 6D pose loss for asymmetric objects sums up the pose losses over all stable patch groups and those of all planar and cylindrical patches:

$$\mathcal{L}^{\text{asym}} = \sum_{k=1}^{|\mathcal{G}|} \mathcal{L}_k^{\text{group}} + \sum_{i=1}^{|\mathcal{P}^{\text{P}}|} \mathcal{L}_i^{\text{planar}} + \sum_{i=1}^{|\mathcal{P}^{\text{C}}|} \mathcal{L}_i^{\text{cylind}} \quad (4)$$

### 4.3.3 Pose Estimation of Symmetric Objects

**Discrete symmetry case** Object symmetries introduce ambiguities to pose estimation. A common strategy of resolving symmetry ambiguities in training a pose estimation network is “back-propagate with the best” [46, 27]. In particular, suppose the object possesses a discrete set of symmetries  $\{\mathbf{T}_i^{\text{S}}\}_{i=1, \dots, M}^1$  and its 6D pose estimated by the current neural network is  $\mathbf{T}$ . In the next round of training, we choose the transformation from  $\{\mathbf{T}, \mathbf{T}\mathbf{T}_1^{\text{S}}, \dots, \mathbf{T}\mathbf{T}_M^{\text{S}}\}$  such that it leads to the minimal alignment error between the observation and the template model, and back-propagate this error. Using this scheme, however, the network tends to memorize a particular pose of the object and lacks a global understanding of object symmetries resulting in weak generality on occluded or unseen objects. To overcome this issue, we propose a new training method for handling symmetry ambiguities — “back-propagate with all”.

In our approach, instead of estimating only a single object pose, we predict  $M + 1$  object poses, each against one of the ground-truth poses in  $\{\bar{\mathbf{T}}, \bar{\mathbf{T}}\mathbf{T}_1^{\text{S}}, \dots, \bar{\mathbf{T}}\mathbf{T}_M^{\text{S}}\}$ , using again a three-layer MLP. Since there are multiple outputs, the correspondence between the predictions and ground-truths should be determined on-the-fly, so that the loss of each individual prediction could be evaluated and back-propagated. To this end, we devise an optimal assignment process [26] which finds a benefit-maximizing correspondence by solving the following optimization:

$$\begin{aligned} & \arg \max_{\Pi} \sum_{m=1}^{M+1} \sum_{k=1}^{M+1} B_{m,k} \Pi_{m,k}, \\ \text{s.t. } & \sum_{m=1}^{M+1} \Pi_{m,k} = 1, \sum_{k=1}^{M+1} \Pi_{m,k} = 1, m, k \in \{1, \dots, M+1\}. \end{aligned}$$

<sup>1</sup>A symmetry transformation  $\mathbf{T}^{\text{S}} \in \text{SE}(3)$  of a 3D shape  $\mathbf{X}$  satisfies  $\mathbf{T}^{\text{S}}\mathbf{X} = \mathbf{X}$  and  $\mathbf{T}^{\text{S}} \neq \mathbf{I}$ .

$\Pi$  is a permutation matrix with  $\Pi_{m,k} \in \{0, 1\}$  indicating whether the  $m$ -th predicted pose matches the  $k$ -th ground-truth pose.  $M + 1$  is the total number of possible object poses given  $M$  object symmetries.  $\mathbf{B}$  is a benefit matrix in which  $B_{m,k}$  represents the benefit of matching the  $m$ -th predicted pose to the  $k$ -th ground-truth. A higher similarity in pose leads to a larger benefit. The benefit  $B_{m,k}$  between two poses  $\mathbf{T}_m$  and  $\mathbf{T}_k$  is computed as the point-wise Euclidean error between point sets transformed by the two poses respectively:

$$B_{m,k} = \sum_j \|\mathbf{T}_m \mathbf{x}_j - \mathbf{T}_k \mathbf{x}_j\|, \quad (5)$$

where  $\mathbf{X} = \{\mathbf{x}_j\}$  are sample points on the template model.

Once the correspondences are determined, the pose loss can be evaluated by accumulating the asymmetry loss in Eq. (4) over all  $M + 1$  possible poses under  $M$  symmetries:

$$\mathcal{L}^{\text{dsym}} = \sum_{\mathbf{T}^{\text{S}} \in \mathcal{T}^{\text{S}}} \mathcal{L}^{\text{asym}}(\mathbf{T} \rightarrow \mathbf{T}\mathbf{T}^{\text{S}}). \quad (6)$$

$\mathcal{L}^{\text{asym}}(\mathbf{T} \rightarrow \mathbf{T}\mathbf{T}^{\text{S}})$  means that the loss is computed by replacing the predicted group/patch pose  $\mathbf{T}$  with  $\mathbf{T}\mathbf{T}^{\text{S}}$ , with  $\mathbf{T}^{\text{S}} \in \mathcal{T}^{\text{S}}$  and  $\mathcal{T}^{\text{S}} = \{\mathbf{I}, \mathbf{T}_1^{\text{S}}, \dots, \mathbf{T}_M^{\text{S}}\}$ .

**Continuous symmetry case** For object with a continuous rotational symmetry, the number of ground-truth pose is infinite. Therefore, instead of predicting poses directly, we opt to train a three-layer MLP to regress the rotation axis represented by the object center  $\mathbf{c}^{\text{r}}$  and the orientation vector of the axis  $\mathbf{a}^{\text{r}}$ , based on each stable group. Let us define  $\mathbf{T}(\theta, \mathbf{c}^{\text{r}}, \mathbf{a}^{\text{r}})$  as the transformation of rotating an angle of  $\theta$  about the axis  $(\mathbf{c}^{\text{r}}, \mathbf{a}^{\text{r}})$ . The regression loss of rotation axis for each patch group  $G_k$  is then defined as the point-wise Euclidean error between model point sets transformed by the predicted and ground-truth rotational transformations respectively:

$$\mathcal{L}_k^{\text{rot}} = \frac{1}{|\Theta|} \sum_{\theta \in \Theta} \sum_j \|\mathbf{T}(\theta, \mathbf{c}^{\text{r}}, \mathbf{a}^{\text{r}}) \mathbf{x}_j - \mathbf{T}(\theta, \bar{\mathbf{c}}^{\text{r}}, \bar{\mathbf{a}}^{\text{r}}) \mathbf{x}_j\|, \quad (7)$$

where  $(\bar{\mathbf{c}}^{\text{r}}, \bar{\mathbf{a}}^{\text{r}})$  is the ground-truth rotation axis. The loss is computed over a set of rotation angles  $\Theta = \{k \cdot \pi/8\}_{k=1, \dots, 16}$ . The overall loss sums up the losses of both per-group axis prediction and per-patch pose prediction:

$$\mathcal{L}^{\text{csym}} = \sum_{k=1}^{|\mathcal{G}|} \mathcal{L}_k^{\text{rot}} + \sum_{i=1}^{|\mathcal{P}^{\text{P}}|} \mathcal{L}_i^{\text{planar}} + \sum_{i=1}^{|\mathcal{P}^{\text{C}}|} \mathcal{L}_i^{\text{cylind}}. \quad (8)$$

Having obtained the rotation axis, we can compute the 6D object pose  $[\mathbf{R}|\mathbf{t}]$ . The translation DoFs are determined:  $\mathbf{t} = \mathbf{c}^{\text{r}}$ . The quaternion corresponding to  $\mathbf{R}$  is

$$q^{\text{R}} = \left( \cos \frac{\gamma}{2}, \sin \frac{\gamma}{2} a_x, \sin \frac{\gamma}{2} a_y, \sin \frac{\gamma}{2} a_z \right)$$

where  $\mathbf{a}^{\text{r}} = (a_x, a_y, a_z)$  is the axis vector and  $\gamma$  can be an arbitrary angle (we set  $\gamma = \frac{\pi}{2}$ ) since there is one rotational DoF that cannot be determined by the rotation axis.

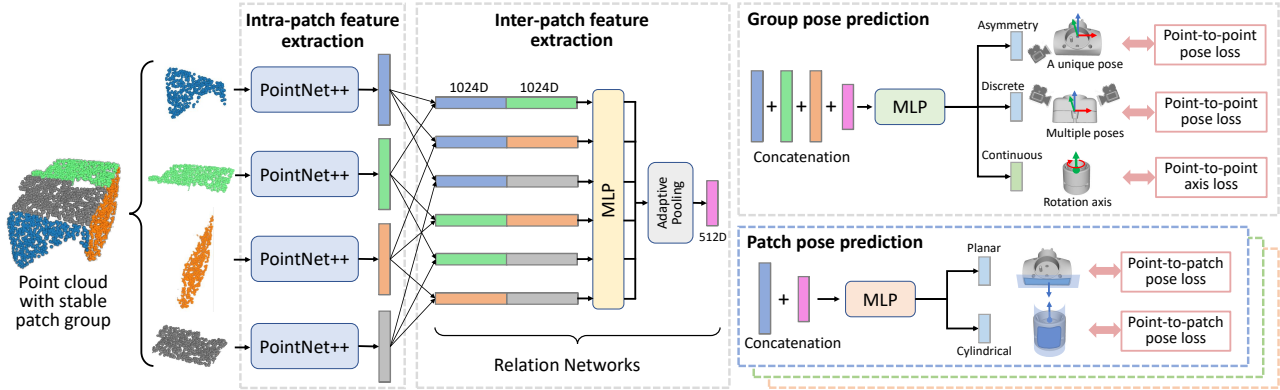


Figure 5: The network architecture of StablePose which is composed of four subnetworks.

#### 4.3.4 Inference

During inference, the 6D pose for each stable patch group is first regressed with the proper network according to the object’s symmetry property. The final object pose is the weighted average over all group poses, with the weight being the stability measure of a group. Higher stability leads to higher weight. Note that the weighed average of rotation and translation are computed separately. The weighted combination of rotations is computed in quaternion form.

#### 4.4. Implementation details

The number of points sampled is 500 on template models in Eq. (1), (5) and (7), and 100 for each patch in Eq. (2) and (3). The dimensions of several critical features are shown in Figure 5. Rotations are represented in quaternion form. The network is trained first with patch-wise pose losses only for 2~3 epoches as a warmup and then with all losses turned on. The batch size is 16. For some small objects with less than three patches extracted, we use PointNet++ [35] to extract feature for the object points and then regress the pose/axis directly with a three-layer MLP. For objects with no stable patch group extracted, we simply use the inter-patch contextual feature to make pose/axis prediction. The ratio of these two cases is 10~30% for the datasets we have tested.

### 5. Results and Evaluation

**Datasets** We conduct evaluation on three public datasets: T-LESS [18], LineMOD-O [2] and NOCS-REAL275 [48]. T-LESS is challenging as the objects are texture-less mechanical parts with similar appearance and geometry and most possess symmetries. LineMOD-O is one of the most widely used datasets for 6D object pose estimation with RGBD images of texture-less household objects with heavy occlusion. NOCS-REAL275 is used to evaluate category-level object pose estimation. The test set contains unseen objects of five categories with different shape and size from those in the training set. To better evaluate the generality on unseen objects with larger

shape variations, we created a synthetic dataset based on ShapeNet [4], named ShapeNetPose. This new dataset contains rendered RGBD images of objects from 22 categories. The test objects are usually significantly different from the training ones in terms of shape and appearance; *please refer to the supplemental material for an overview of this dataset.*

**Metrics** For instance-level pose estimation, to facilitate comparison to previous works, we adopt the 6D localization recall under two popular pose-error functions: Average Closest Point Distance (ADI) [16] and Visible Surface Discrepancy (VSD) [20]. ADI considers both visible and invisible parts of the 3D model surface. Suppose  $\mathbf{T}$  is the predicted pose and  $\bar{\mathbf{T}}$  the ground-truth, ADI measures the mean Euclidean distance from each point on the 3D model transformed by  $\mathbf{T}$  to the closest point on the model transformed by  $\bar{\mathbf{T}}$ . We use the standard ADI recall metric  $e_{ADI} < 0.1d$ , where  $d$  is the object diameter [20, 27]. VSD considers visible parts only. To evaluate the VSD error of an estimated pose  $\mathbf{T}$  w.r.t. its ground-truth  $\bar{\mathbf{T}}$ , we first render the template model into two depth maps  $D$  and  $\bar{D}$  using the two poses, respectively. VSD measures the differences between  $D$  and  $\bar{D}$  at image locations where the object is visible in the input observation. We use the standard VSD recall metric  $e_{VSD} < 0.3$  with  $\tau = 20\text{mm}$  and  $\delta = 15\text{mm}$  [20, 27]. For T-LESS, the results are reported on 6D localization both for varying number of instances of varying number of objects in single-view RGBD images (VIVO) [20] and for a single instance of a single object (SISO) [19]. For LineMOD-O, since each RGB-D image contains at most one instance for one object, we only report results for SISO.

#### 5.1. Qualitative Results

Figure 6 shows some visual results of CosyPose [27] and our StablePose on T-LESS and LineMOD-O. These examples encompass objects with heavy occlusion (e.g. column 1–5), discrete symmetry (column 2 and 3) and continuous symmetry (column 4). In all these challenging cases, our method is able to estimate the 6D object poses accu-

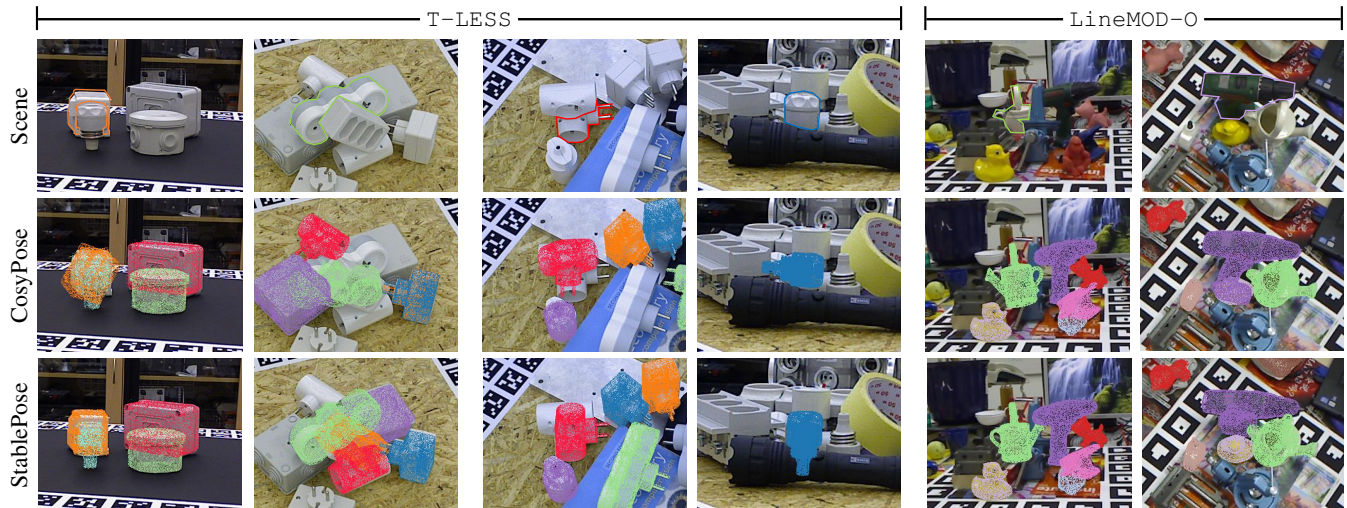


Figure 6: Visual results of 6D object pose estimation by *CosyPose* (single-view) [27] and *StablePose* on T-LESS and LineMOD-O. Please pay special attention to the results of the challenging objects highlighted in the input scene.

rately. Especially for those objects with severe occlusion, our poses are quite accurate thanks to the per-patch pose prediction as a reinforcement of the global pose prediction; see a visual ablation study in the supplemental material.

## 5.2. Quantitative Comparisons

**Comparing to depth-only methods** We first compare our method to several baselines that take only depth image as input: *Drost-PPF* [9], *Vidal-PPF* [44], *PointNet++* [44] and *PPFNet* [7]. Both *Drost-PPF* and *Vidal-PPF* are state-of-the-art methods based on the point pair features. They simultaneously detect objects and estimate poses. We also build two baselines by using *PointNet++* and *PPFNet* as point feature extraction backbone, respectively, followed by a three-layer MLP for regressing 6D pose trained with dense-pixel loss [46]. Like our method, a pretrained object detector [30] is used for these methods. The experiments are conducted on T-LESS and LineMOD-O; see the results in Table 1 and Table 2. *StablePose* outperforms all the baselines over both datasets for all metrics. *StablePose* outperforms the learning-based methods significantly. *Drost-PPF* and *Vidal-PPF* are comparably accurate but 3~4 times slower than *StablePose* inference. Moreover, the learned *StablePose* model can generalize to handle unseen objects without a template model (see Table 4) which is difficult, if not impossible, for non-learning-based methods. Note, for the experiments on LineMOD-O, we use ADI/ADD, instead of ADI, as LineMOD-O contains many non-symmetric objects [20].

**Comparing to RGBD methods** We compare to the following RGBD baselines: *DenseFusion* [46], *Pix2Pose* [30] and *CosyPose* (single-view) [27]. *DenseFusion* achieves 6D pose estimation by fusing features of RGB and depth. *Pix2Pose* trains an auto-encoder to regress pixel-wise 3D coordinates. *CosyPose* is the state-of-the-art method which

predicts poses using RGB image followed by an ICP refinement with depth image. The comparison is conducted on T-LESS and LineMOD-O; see results in Table 1 and Table 2. Although using only depth in pose prediction, *StablePose* beats the RGBD methods on T-LESS by a large margin. On LineMOD-O, our method outperforms all methods except *CosyPose* which was trained using a much larger training set of RGB images. The training of *StablePose* requires a much smaller dataset and hence is significantly faster. This implies that learning on depth/geometric input is more data-efficient for the task of 6D pose estimation.

## 5.3. Parameter Setting and Ablation Studies

In Table 3, we study the parameter setting and design choices of our method.

**Patch count of a stable group** Our method selects *three patches* for each stable group. Here we evaluate other possibilities: *one-patch*, *two-patch* and *five-patch*. All baselines adopt the same network setting as the main method. For one-patch and two-patch cases, the final pose takes average over all groups with uniform weights since stability measure can hardly be computed for less than three patches. The results show that patch triplet is the best choice for constructing stable groups for pose estimation.

**Patch group sampling** To evaluate the necessity of *stability-based* patch group sampling, we compare to two baselines: *size-based sampling* and *distance-based sampling*. The former selects patch triplets by point count: Any three patches whose total point count is larger than a threshold (1000) form a group. The latter picks patch triplets such that the sum up of pair-wise patch distances exceeds a threshold (8 cm). Table 3 shows that stability-based sampling performs the best. This confirms the idea that geometrically stable patches complement better to each other in terms of determining the DoFs of object pose.

Table 1: Performance comparison on T-LESS.

	Pose Est.	$e_{\text{ADI}}$ (VIVO)	$e_{\text{ADI}}$ (SISO)	$e_{\text{VSD}}$ (VIVO)	$e_{\text{VSD}}$ (SISO)	Training data	Training time	Inference time
Drost-PPF [9]	D	-	-	-	0.57	-	-	1.3s
Vidal-PPF [44]	D	-	-	-	0.72	-	-	1.6s
PointNet++ [35]	D	0.74	0.78	0.50	0.54	37K	15h	0.4s
PPFNet [7]	D	0.76	0.79	0.44	0.49	37K	15h	0.4s
DenseFusion [46]	RGBD	0.13	0.15	0.08	0.10	37K	15h	0.1s
Pix2Pose [30]	RGBD	-	-	-	0.30	37K	80h	0.6s
CosyPose [27]	RGBD	0.68	0.75	0.63	0.64	1M	200h	1.1s
StablePose	D	<b>0.86</b>	<b>0.88</b>	<b>0.69</b>	<b>0.73</b>	37K	15h	0.4s

Table 2: Performance comparison on LineMOD-O.

	$e_{\text{ADI/ADD}}$ (SISO)	$e_{\text{VSD}}$ (SISO)	Training data
Drost-PPF [9]	-	0.55	-
Vidal-PPF [44]	-	0.62	-
Pix2Pose [30]	0.32	-	10K
CosyPose [27]	<b>0.68</b>	<b>0.83</b>	1M
StablePose	0.63	0.71	10K

Table 3: Parameter setting and ablation studies.

	$e_{\text{ADI}}$ (SISO)	$e_{\text{VSD}}$ (SISO)
one-patch	0.70	0.57
two-patch	0.76	0.63
five-patch	0.73	0.69
size-based sampling	0.85	0.69
distance-based sampling	0.77	0.68
w/o patch-wise pose	0.73	0.62
baseline symmetry handling	0.75	0.54
StablePose	<b>0.88</b>	<b>0.73</b>

**Network design and training scheme** We also compare to baselines of *without inter-patch contextual feature*, *without patch-wise pose estimation* and “*back-propagate with the best*” in symmetry handling, by ablating each of these algorithmic components. The degraded performance of the baselines validates our design choices. Crucially, the performance drops significant for *without patch-wise pose estimation*, which suggests that this auxiliary task indeed provides substantial constraints for improving pose accuracy.

#### 5.4. Category-Level Pose Estimation

The generality of our method can be best reflected by *category-level 6D pose estimation* in which the object instance is unseen during training. We test our method on NOCS-REAL275 and ShapeNetPose, and compare with two state-of-the-art methods: NOCS [48] and CASS [5]. Note, both NOCS and CASS predict pose with RGBD, while our method does so with only depth input. The experiments are evaluated using  $10^\circ 10\text{cm}$ ,  $IoU_{25}$ ,  $R_{err}$  and  $T_{err}$  as in [48]. In Table 4, we show the results on ShapeNetPose (*the results on NOCS-REAL275 is in the supplemental material*). The better cross-instance generality of StablePose is due to the repeatability of patch groups

Table 4: Comparison on category-level pose estimation over ShapeNetPose.

	Pose Est.	$10^\circ 10\text{cm}$	$IoU_{25}$	$R_{err}$	$T_{err}$
NOCS [48]	RGBD	12.8	61.7	33.5	19.3
CASS [5]	RGBD	13.9	67.3	32.9	17.6
Ours	D	<b>21.4</b>	<b>92.1</b>	<b>20.9</b>	<b>9.6</b>

across object instances and the pose prediction learning over a redundant set of patch groups.

## 6. Discussion, Limitations and Future Works

With our work, we hope to deliver the following key messages. 1) Shape and pose of objects are tightly coupled in visual perception. 2) 6D object pose can be pinned down by a minimal set of geometrically stable patches sampled on the object surface. 3) Although each patch determines only a subset of the six DoFs, their under-determined predictions can cumulatively constrain the object pose, resulting in high accuracy and robustness. We have realized these ideas with StablePose, a multi-task deep neural network with good generalization. The network achieves the state-of-the-art performance on several public benchmarks.

Our current solution has the following limitations on which future investigations could be conducted. *First*, object detection still relies on RGB input. Designing a single-stage object detection and pose estimation for depth-only input is an interesting problem to study. *Second*, although our method works on redundant set of patch groups, the quality of patch extraction is of great importance. It is difficult to extract valid patches from very small or incompletely scanned objects. In the future, we plan to realize implicit patch stability analysis in a more end-to-end fashion without explicit patch extraction and stable group sampling.

## Acknowledgements

We thank the anonymous reviewers for their valuable comments. This work was supported in part by National Key Research and Development Program of China (2018AAA0102200), NSFC (61825305, 62002379) and the Zhejiang Lab’s International Talent Fund for Young Professionals.



## References

- [1] Martin Bokeloh, Alexander Berner, Michael Wand, Hans-Peter Seidel, and Andreas Schilling. Slippage features. 2008. [2](#)
- [2] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6d object pose estimation using 3d object coordinates. In *European conference on computer vision*, pages 536–551. Springer, 2014. [2](#), [6](#)
- [3] Benedict J Brown and Szymon Rusinkiewicz. Global non-rigid alignment of 3-d scans. In *ACM SIGGRAPH 2007 papers*, pages 21–es. 2007. [2](#)
- [4] 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 preprint arXiv:1512.03012*, 2015. [6](#)
- [5] Dengsheng Chen, Jun Li, Zheng Wang, and Kai Xu. Learning canonical shape space for category-level 6d object pose and size estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11973–11982, 2020. [8](#)
- [6] Wei Chen, Xi Jia, Hyung Jin Chang, Jinming Duan, and Ales Leonardis. G2l-net: Global to local network for real-time 6d pose estimation with embedding vector features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4233–4242, 2020. [1](#)
- [7] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppfnet: Global context aware local features for robust 3d point matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 195–205, 2018. [7](#), [8](#)
- [8] Thanh-Toan Do, Ming Cai, Trung Pham, and Ian Reid. Deep-6dpose: Recovering 6d object pose from a single rgb image. *arXiv preprint arXiv:1802.10367*, 2018. [2](#)
- [9] Bertram Drost, Markus Ulrich, Nassir Navab, and Slobodan Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 998–1005. Ieee, 2010. [1](#), [2](#), [7](#), [8](#)
- [10] Guoguang Du, Kai Wang, and Shiguo Lian. Vision-based robotic grasping from object localization, pose estimation, grasp detection to motion planning: A review. *arXiv preprint arXiv:1905.06658*, 2019. [2](#)
- [11] Martha J Farah and Katherine M Hammond. Mental rotation and orientation-invariant object recognition: Dissociable processes. *Cognition*, 29(1):29–46, 1988. [1](#)
- [12] Ge Gao, Mikko Lauri, Yulong Wang, Xiaolin Hu, Jianwei Zhang, and Simone Frntrop. 6d object pose regression via supervised learning on point clouds. *arXiv preprint arXiv:2001.08942*, 2020. [1](#)
- [13] Natasha Gelfand and Leonidas J Guibas. Shape segmentation using local slippage analysis. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 214–223, 2004. [1](#), [2](#)
- [14] Natasha Gelfand, Leslie Ikemoto, Szymon Rusinkiewicz, and Marc Levoy. Geometrically stable sampling for the icp algorithm. In *Fourth International Conference on 3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings.*, pages 260–267. IEEE, 2003. [2](#)
- [15] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11632–11641, 2020. [2](#)
- [16] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *Asian conference on computer vision*, pages 548–562. Springer, 2012. [6](#)
- [17] Tomas Hodan, Daniel Barath, and Jiri Matas. Epos: Estimating 6d pose of objects with symmetries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11703–11712, 2020. [1](#), [2](#)
- [18] Tomáš Hodan, Pavel Haluza, Štěpán Obdržálek, Jiri Matas, Manolis Lourakis, and Xenophon Zabulis. T-less: An rgb-d dataset for 6d pose estimation of texture-less objects. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 880–888. IEEE, 2017. [2](#), [6](#)
- [19] Tomáš Hodaň, Jiří Matas, and Štěpán Obdržálek. On evaluation of 6d object pose estimation. In *European Conference on Computer Vision*, pages 606–619. Springer, 2016. [6](#)
- [20] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 19–34, 2018. [6](#), [7](#)
- [21] Yinlin Hu, Pascal Fua, Wei Wang, and Mathieu Salzmann. Single-stage 6d object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2930–2939, 2020. [2](#)
- [22] Alfred Inselberg and Bernard Dimsdale. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *Proceedings of the First IEEE Conference on Visualization: Visualization90*, pages 361–378. IEEE, 1990. [4](#)
- [23] Hamid Karimi-Rouzbahani, Nasour Bagheri, and Reza Ebrahimpour. Invariant object recognition is a personalized selection of invariant features in humans, not simply explained by hierarchical feed-forward vision models. *Scientific reports*, 7(1):1–24, 2017. [1](#)
- [24] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. SSD-6D: Making rgb-based 3d detection and 6d pose estimation great again. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1521–1529, 2017. [2](#)
- [25] Wadim Kehl, Fausto Milletari, Federico Tombari, Slobodan Ilic, and Nassir Navab. Deep learning of local rgb-d patches for 3d object detection and 6d pose estimation. In *European conference on computer vision*, pages 205–220. Springer, 2016. [2](#), [3](#)
- [26] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. [5](#)

- [27] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. Cosypose: Consistent multi-view multi-object 6d pose estimation. *arXiv preprint arXiv:2008.08465*, 2020. 2, 5, 6, 7, 8
- [28] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. Deepim: Deep iterative matching for 6d pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 683–698, 2018. 2
- [29] Chaitanya Mitash, Abdeslam Boularias, and Kostas Bekris. Robust 6d object pose estimation with stochastic congruent sets. *arXiv preprint arXiv:1805.06324*, 2018. 2
- [30] Kiru Park, Timothy Patten, and Markus Vincze. Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7668–7677, 2019. 1, 2, 3, 7, 8
- [31] Kiru Park, Timothy Patten, and Markus Vincze. Neural object learning for 6d pose estimation using a few cluttered images. *arXiv preprint arXiv:2005.03717*, 2020. 2
- [32] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. Pvnnet: Pixel-wise voting network for 6dof pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4561–4570, 2019. 2
- [33] Helmut Pottmann and Michael Hofer. Geometry of the squared distance function to curves and surfaces. In *Visualization and mathematics III*, pages 221–242. Springer, 2003. 2
- [34] Pedro F Proença and Yang Gao. Fast cylinder and plane extraction from depth cameras for visual odometry. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6813–6820. IEEE, 2018. 3
- [35] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017. 3, 6, 8
- [36] Mahdi Rad and Vincent Lepetit. Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3828–3836, 2017. 1, 2
- [37] Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. In *Advances in neural information processing systems*, pages 4967–4976, 2017. 4
- [38] Yifei Shi, Kai Xu, Matthias Niessner, Szymon Rusinkiewicz, and Thomas Funkhouser. Planematch: Patch coplanarity prediction for robust rgb-d reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 750–766, 2018. 4
- [39] Chen Song, Jiaru Song, and Qixing Huang. Hybridpose: 6d object pose estimation under hybrid representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 431–440, 2020. 2
- [40] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3d orientation learning for 6d object detection from rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 699–715, 2018. 1, 2
- [41] Supasorn Suwajanakorn, Noah Snavely, Jonathan J Tompson, and Mohammad Norouzi. Discovery of latent 3d keypoints via end-to-end geometric reasoning. In *Advances in neural information processing systems*, pages 2059–2070, 2018. 2
- [42] Michael J Tarr, Pepper Williams, William G Hayward, and Isabel Gauthier. Three-dimensional object recognition is viewpoint dependent. *Nature neuroscience*, 1(4):275–277, 1998. 1
- [43] Bugra Tekin, Sudipta N Sinha, and Pascal Fua. Real-time seamless single shot 6d object pose prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 292–301, 2018. 2
- [44] 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):2678, 2018. 1, 2, 7, 8
- [45] Kentaro Wada, Edgar Sucar, Stephen James, Daniel Lenton, and Andrew J Davison. Morefusion: Multi-object reasoning for 6d pose estimation from volumetric fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14540–14549, 2020. 1
- [46] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3343–3352, 2019. 1, 2, 5, 7, 8
- [47] Gu Wang, Fabian Manhardt, Jianzhun Shao, Xiangyang Ji, Nassir Navab, and Federico Tombari. Self6d: Self-supervised monocular 6d object pose estimation. *arXiv preprint arXiv:2004.06468*, 2020. 2
- [48] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2642–2651, 2019. 2, 6, 8
- [49] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017. 1, 2
- [50] Zelin Xu, Ke Chen, and Kui Jia. W-posenet: Dense correspondence regularized pixel pair pose regression. *arXiv preprint arXiv:1912.11888*, 2019. 2
- [51] Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. Dpod: 6d pose object detector and refiner. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1941–1950, 2019. 2
- [52] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1802–1811, 2017. 1