SporeAgent: Reinforced Scene-level Plausibility for Object Pose Refinement

Dominik Bauer[1], Timothy Patten[1,2] and Markus Vincze[1]
[1]TU Wien Vienna, Austria
[2]University of Technology Sydney Sydney, Australia
{bauer,patten,vincze}@acin.tuwien.ac.at

Abstract

Observational noise, inaccurate segmentation and ambiguity due to symmetry and occlusion lead to inaccurate object pose estimates. While depth- and RGB-based pose refinement approaches increase the accuracy of the resulting pose estimates, they are susceptible to ambiguity in the observation as they consider visual alignment. We propose to leverage the fact that we often observe static, rigid scenes. Thus, the objects therein need to be under physically plausible poses. We show that considering plausibility reduces ambiguity and, in consequence, allows poses to be more accurately predicted in cluttered environments. To this end, we extend a recent RL-based registration approach towards iterative refinement of object poses. Experiments on the LINEMOD and YCB-VIDEO datasets demonstrate the state-of-the-art performance of our depth-based refinement approach. See github.com/dornik/sporeagent.

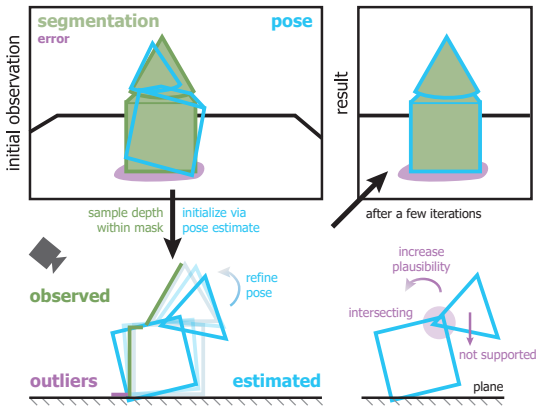

Figure 1. Objects are segmented and initial poses are estimated (upper left). The scene is initialized and all objects therein are refined iteratively. In each iteration, we more closely align the observed source and the target model point clouds (lower left), while increasing physical plausibility of the scene (lower right).

1. Introduction

Applications such as robotic grasping or augmented reality require the estimation of accurate object poses. This allows an observed scene to be represented by a set of 3D meshes and enables interaction with these scene objects. Given an initial pose estimate, the task of object pose refinement is to closely align an object's mesh with the corresponding image patch or point cloud to improve accuracy.

Image-based approaches aim to align with an RGB(D) image patch or instance segmentation mask [15, 28, 20, 7]. Approaches using point clouds align the sampled surface of the object mesh with an observed point cloud that is computed from a depth image [5]. RGB images provide high resolution and textural information but distance and scale are ambiguous. Using absolute depth information resolves this ambiguity. However, depth sensors offer lower resolution and suffer from artifacts such as depth shadowing and quantization errors. While approaches combining both modalities achieve high accuracy [15, 10, 23], they are bound by the limitations of visual observability. Indistinguishable views of an object due to (self)occlusion and symmetry lead to ambiguous refinement targets.

To resolve such ambiguous situations, recent approaches additionally consider the physical configuration of the complete scene [22, 16, 3]. This additional source of information exploits the pose estimates of other scene objects, limiting the possible object interactions. These approaches are, however, limited to resolving collisions [22] or require physics simulation [16, 3].

In contrast to prior work, we guide our refinement approach towards non-intersecting, non-floating and statically stable scenes by using a contact-based definition of physical plausibility [2]. We extend a Reinforcement Learning-based point cloud registration approach [4] by considering surface distance as proxy for scene-level interaction information and reinforce a plausibility-based reward. Geometrical symmetry is considered to stabilize training using Imitation and Reinforcement Learning (IL+RL).

Exploiting this additional source of information in combination with point cloud based alignment, our ***Scene-***

*level **P**lausibility for **O**bject Pose **Re**finement* Agent (SporeAgent) achieves accuracy comparable to RGBD approaches and outperforms competing RGB and depth-only approaches. While RGB(D)-based methods need large amounts of training data to converge, we are able to use a fraction of training images, using only 1/100th of the real training images provided for YCB-VIDEO.

In summary, we achieve the presented results by:

- integrating physical plausibility with IL+RL-based iterative refinement of object poses;
- simultaneously refining all objects in a scene, recovering from pose initialization errors due to ambiguous alignment using scene-level physical plausibility;
- considering geometrical symmetry, further dealing with indistinguishable views in the depth domain and
- achieving state-of-the-art accuracy, while requiring limited amounts of training data.

After an overview of previous work in Section 2, we briefly introduce the related tasks of point cloud registration and object pose refinement in Section 3. Extensions to IL+RL-based point cloud registration tackling the discussed challenges in pose refinement are proposed in Section 4. We evaluate our approach on the LINEMOD and YCB-VIDEO datasets and provide an ablation study of the proposed extensions in Section 5. Section 6 concludes the paper and gives an outlook to future work.

2. Related Work

Closely related to depth-based pose refinement, the task of point cloud registration is recently approached as a learning task. Building on the core ideas of ICP [5], learned feature representations improve the matching between corresponding points in the source and the target point clouds [24, 9, 26]. Yuan et al. [27] learn Gaussian Mixture Model parameters for probabilistic registration. A further line of work uses a learned global feature representation to represent the iterative registration state, combined with an application of the Lukas-Kanade algorithm [1] or Reinforcement Learning (RL) [4]. Our refinement approach is based on the latter RL approach as it suited to incorporate domain knowledge such as physical plausibility.

Learning-based object pose refinement, in contrast, is commonly based on RGB images [15, 20, 7, 28] where the goal is to align a rendering of the object under estimated pose to an observed image patch. Seminal work by Li et al. [15] uses optical flow features to predict the refinement transformation to more closely align the given image patches. The approaches of Shao et al. [20] and Busam et al. [7] consider this as a RL task and learn a policy that predicts discrete refinement actions. An alternative use of RL is proposed in [13], where RL actions are selected as one of a pool of pose hypotheses to be refined in each iteration.

However, these approaches in point cloud registration and object pose refinement consider a single registration target (i.e., a single object) at a time. Especially in heavily cluttered scenes that may occur, for example, during robotic manipulation, consideration of the full scene of objects is shown to be beneficial. Labbé et al. [14] propose a global refinement scheme, incorporating all scene objects in a joint pose optimization and incorporate multiple views of the scene. By fusing multiple views into a voxel grid, Wada et al. [22] additionally consider the collision between scene objects in their refinement method. A formalization of this concept of considering physical interactions between static scene objects for evaluation of pose estimates is proposed in [2], which is used for our definition of physical plausibility. In contrast, the methods proposed by Mitash et al. [16] and Bauer et al. [3] apply physics simulation to scene estimates, leveraging the simulated dynamics of the scene objects to improve pose hypotheses and verify the best aligned combinations with respect to the observed depth image. We employ a similar depth-based pose scoring to determine the best pose during refinement.

3. Background: Point Cloud Registration using Imitation and Reinforcement Learning

The task of depth-based object pose refinement is closely related to point cloud registration. In this section, we give a brief overview of registration, link it to refinement and introduce terms used throughout the paper. We additionally discuss how registration – and by extension refinement – is approached as a reinforcement learning task.

3.1. Point Cloud Registration and Refinement

Assume we are given a source X and target point cloud Y that, for example, both represent a single object. The target is offset from the source by an unknown rigid transformation T of form $[R \in SO(3), t \in \mathbb{R}^3]$. The task of registration is to find a transformation $\hat{T}$ such that the source and target are aligned again, i.e., $\hat{T}T^{-1} = I$. This transformation may be found directly or iteratively by updating the source with intermediary transformations $X_{i+1} = \hat{T}_i X$. Moreover, an initial estimate $\hat{T}_0$ may be provided.

Translated to object pose refinement, the source is a partial and noise afflicted view of the object, represented by a point cloud, and the target is represented by a 3D mesh. The initial estimate is commonly computed using a dedicated object pose estimator.

3.2. Reinforced Point Cloud Registration

Iteratively determining a transformation $\hat{T}_i$ that, with each iteration i, more closely aligns two point clouds

(X, Y) may be interpreted as taking a sequence of refinement actions. We will refer to such a sequence $\{a_1, ..., a_{i+1}\}$ as a *refinement trajectory*. Considering the prediction of such trajectories as a RL task, the goal is to determine a policy that selects a suitable refinement action a_{i+1} given the current observation $O_i = (X_i, Y)$. In related work [20, 4], the action space for this task is assumed to consist of a set of discrete steps per transformation dimension. Hence, in each iteration, the policy must determine a step per axis, separately for rotation and translation. Such an action, for example, might relate to taking a small translation step along x-axis. The policy for this action space is represented by a discrete probability distribution $\pi(a|O)$ that is conditioned on the observation.

Such a policy is predicted by the network architecture presented in [4]. A simplified PointNet [17] is used as embedding, consisting of three 1D-convolution layers of dimension $[64, 256, 1024]$. Source and target are processed independently with shared weights. The concatenation of the max pooling of the final layer constitutes the state vector and is used as representation of the observation. The state vector serves as input for the policy network, consisting of two separate heads for rotation and translation actions and an additional head for computing a value estimate for the current state. The value estimate serves as baseline for the advantage computation used in RL. All heads are implemented using fully-connected layers of dimension $[512, 256, D]$ and the policy heads' output is interpreted as a discrete probability distribution over refinement actions.

An expert policy π^* is able to determine the optimal actions in every iteration by taking the largest step towards the ground-truth pose T. Formally, such an expert policy may be defined [4] by taking the largest rotation and translation actions per-axis to minimize the residuals

$$\delta_i^R = R\hat{R}_i^\top, \quad \delta_i^t = t - \hat{t}_i, \tag{1}$$

where $T = [R, t]$ is the ground-truth transformation and $\hat{T}_i = [\hat{R}_i, \hat{t}_i]$ is the current estimate. The actions predicted by this policy may be used in conjunction with IL approaches such as Behavioral Cloning, for which they serve as ground-truth labels during training.

A combination of IL with RL is proposed in [4]. An alignment-based reward r_a is reinforced via Proximal Policy Optimization (PPO) [19] in a discounted task. For an action a_{i+1} in iteration i, it is defined as

$$r_a = \begin{cases} -\rho^-, & CD(X_{i+1}, X^*) > CD(X_i, X^*) \\ -\rho^0, & CD(X_{i+1}, X^*) = CD(X_i, X^*) \\ \rho^+, & CD(X_{i+1}, X^*) < CD(X_i, X^*), \end{cases} \tag{2}$$

where CD is the Chamfer distance, $X^* = TX$ is the source under ground-truth pose and $(-\rho^-, -\rho^0, \rho^+)$ is a set of rewards for worsening, stagnating and improving the alignment with respect to the previous step. The loss from IL is blended with the PPO loss via a scaling factor α that affects the impact of PPO. We employ this combined approach as it is shown to quickly converge, while allowing further domain knowledge to be integrated via adaptation of the expert policy in Equation (1) and the reward in Equation (2).

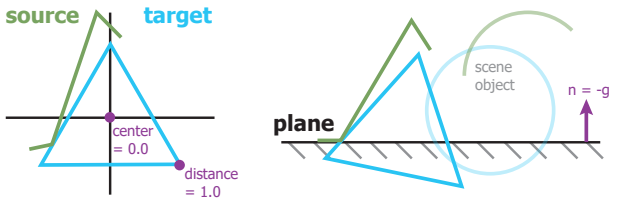

Figure 2. Representation for per-object refinement (left) and for computation of the scene-level information (right).

4. Reinforced Scene-level Plausibility for Object Pose Refinement

In applications such as robotic grasping or augmented reality, erroneous object pose estimates may result in missed grasps or might disrupt the immersion of the user. To improve upon single-shot pose estimation, object pose refinement aims to iteratively increase the alignment of an object under estimated pose with an observation.

Object pose refinement from a single view further increases the difficulty of the task. First, objects are commonly observed in the context of complex scenes, requiring instance segmentation of the object of interest. Second, many classes of objects (e.g, cylindrical, box-shaped) feature geometrical symmetries, which makes alignment ambiguous. Finally, objects are only partially observable from a single view. This issue is aggravated in cluttered scenes due to occlusion between objects.

To tackle each of these challenges, we extend the reinforced point cloud registration approach presented in Section 3 towards pose refinement. Consideration of geometrical symmetry in the expert policy and surface normals as additional input are discussed in Section 4.1. As presented in Section 4.2, we furthermore integrate contact-based physical plausibility as input and at the RL stage to deal with visual ambiguity.

4.1. From Registration to Object Pose Refinement

The baseline approach is designed for registration of noisy yet complete point clouds. In object pose refinement, however, the observed point cloud represents only a single partial view of the object. The required instance segmentation might include points belonging to the background or close-by objects. Hence, the noisy source only partially overlaps with the noise-free target that is sampled from the object's 3D mesh. The target may also have geometrical symmetries. The correct alignment between source and tar-

get point cloud is thus ambiguous, as any symmetrical pose will result in an indistinguishable observation of the object. The following extensions enable the registration to perform accurately on real data on the pose refinement task.

4.1.1 Extending Input and Embedding

To deal with inaccurate segmentation of the object, we adapt the input representation and extend it by further geometrical information. Surface normals are a strong geometrical cue for outlier points as the observed normals are subject to large changes at object borders. Source normals are estimated from the observed point cloud. For the target point cloud, they are retrieved from the mesh face from which the points are sampled.

Normalizing the input point clouds provides an inductive bias that any point outside the unit sphere is either misaligned or an outlier. This also reduces the range of the potential inputs. Formally, the normalization with respect to the target Y is defined by

$$X' = (X - \mu_Y)/d_Y, \quad Y' = (Y - \mu_Y)/d_Y, \qquad (3)$$

where μ_Y is the centroid of the target and d_Y is the maximal distance from the centroid to any point in the target. See Figure 2 (left) for an illustration. Supporting the uniform coverage of the input space further, we align objects into a canonical orientation in which the major symmetry axis is aligned with the z-axis. Cylindrical, cuboid and box-shaped objects are oriented uniformly as shown in Figure 3. Thereby, objects' targets will more closely align and thus leverage similar geometrical features. This supports training and results in higher refinement accuracy as discussed in the ablation study in Section 5.3.

To accommodate this additional input and further extensions to be discussed in the following sections, we add two additional layers to the baseline's embedding network, resulting in five 1D-convolution layers of dimension $[64, 128, 256, 512, 1024]$ in total.

Moreover, outlier points are pruned using a segmentation branch consisting of four 1D-convolution layers of dimension $[512, 256, 128, 1]$. The concatenation of the first and last layer's feature embedding as well as a one-hot class vector results in a $1088 + k$ dimensional input, where k is the number of classes. Points that are labeled *outliers* are ignored in the max pooling operation and thus do not contribute towards the state vector. This outlier removal is trained using the ground-truth instance labels and a binary cross-entropy loss, scaled by a factor β.

4.1.2 Symmetry-aware Expert Policy

The expert policy presented in Section 3 provides the goal trajectories for IL. It is defined to minimize the residual between the estimated and ground-truth pose. However, for symmetrical objects, the object under ground-truth pose is indistinguishable from a transformation by an alternative symmetrical pose. The expert labels are thus inconsistent with respect to the observation and the agent will receive a varying loss for seemingly equivalent actions.

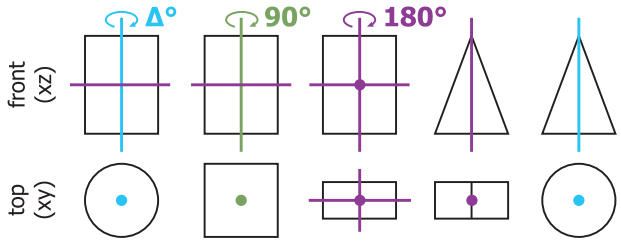

Figure 3. Geometrical symmetry classes. From left to right: Cylindrical, cuboid, box, front-back and rotational. Rotational symmetries (blue) are sampled with a resolution of Δdeg.

To address this issue, we consider multiple equivalent ground-truth poses of form $[R_s, t_s]$ for symmetrical objects. The expert should follow the shortest trajectory and thus move towards the symmetrical pose that is closest to the current estimate $\hat{T}_i$. Due to the symmetry axes coinciding with the origin in our canonical representation of the objects, the symmetry transformations do not involve any translation. The index s_i of the closest pose is thus determined by taking the residual rotation with the smallest angle

$$\begin{aligned} s_i &= \arg\min_s \arccos \frac{trace(R_s \hat{R}_i^\top) - 1}{2} \\ &= \arg\max_s trace(R_s \hat{R}_i^\top). \end{aligned} \qquad (4)$$

The expert actions are computed as before by retrieving the symmetry-aware residuals using Equation (1) with respect to $[R_{s_i}, t_{s_i}]$ as ground truth.

Since most related approaches exploit RGB information, the available symmetry annotations for common pose estimation datasets [12] consider the texture of objects as well as their geometry. Hence, we need to annotate additional geometrical symmetries for use with depth-only observations. In canonical representation, this reduces to assigning objects to one of the five symmetry classes, as illustrated in Figure 3, in our experiments[1].

4.2. Scene-level Physical Plausibility

Visual information alone might still be ambiguous due to occlusion or observational noise. However, assuming the observed scene is static, physically plausible interactions between an object and its support limit the space of admissible object poses within the scene.

[1]The transformations to our canonical representation and the corresponding symmetry annotations per object are provided with our code.

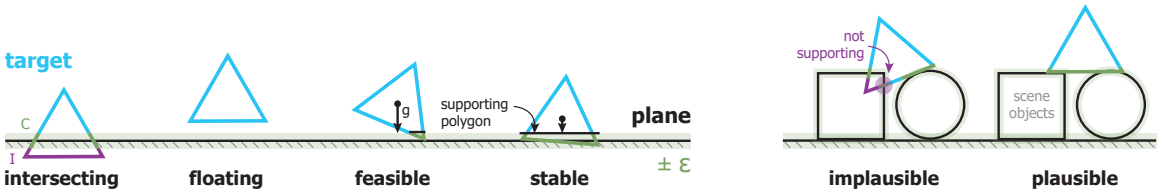

Figure 4. Definition of physical plausibility based on critical points for a single object (left) and a scene (right). If feasible, the CoM projected in gravity direction must intersect the support polygon (convex hull of supported points) to be considered stable.

4.2.1 Definition of Physical Plausibility

Individual aspects of physical plausibility such as collision are modeled using voxel grids [22] or by the dynamics computed using physics simulation [16, 3]. In contrast, physical plausibility defined in [2] is based on the interacting points between objects. As such, this approach is well suited to be incorporated with point cloud inputs.

As illustrated in Figure 4 (left), the signed distance between two objects serves as a basis for the definition of points that are critical for physical plausibility. The signed distance is the Euclidean distance to the closest surface point, with points inside the surface having negative sign. We refer to it as *surface distance* $d(x)$ of a point x.

Points farther within another object than a threshold ε are considered *intersecting*, as defined by

$$I = \{x \in X : d(x) < -\varepsilon\}. \tag{5}$$

Points within an absolute distance of ε are considered *in contact*, or formally

$$C = \{x \in X : |d(x)| < \varepsilon\}. \tag{6}$$

In addition to the definitions of [2], we consider a contact to be *supported*, if

$$S = \{c \in C : \cos(n_y(c) \cdot g) < 0\}, \tag{7}$$

where $n_y(c)$ is the normal of the closest point y to c in the scene and g the gravity direction. These critical points allow to define the plausibility conditions, illustrated in Figure 4.

If there is at least one intersecting point, the object is considered *intersecting*. If an object has no contact point, it is consider to be *floating*. If it is neither intersecting nor floating, the object is defined to be under a *feasible* pose within the current scene. Finally, if the center of mass (CoM) of the object – projected in the gravity direction – falls within the supporting polygon, the object is *stable*. The supporting polygon is defined as the convex hull of the supported points S. An object that is both feasible and stable under a given pose is considered to be *physically plausible*.

4.2.2 Computing Surface Distance and Critical Points

As the used definition of physical plausibility depends on the surface distance, we need to efficiently compute it for the whole scene in every refinement iteration. Moreover, we need to determine an initial supporting plane and gravity direction from the observation.

Assuming the supporting plane is sufficiently visible in the scene, we use a RANSAC-based approach to fit a plane to the subset of the source point cloud that is assigned a background label in the instance segmentation. The plane's pose with respect to the camera is then given by $\hat{P}$. To determine the surface distance between scene objects, we transform all corresponding object point clouds to this plane. The sources are already in camera space and may thus be transformed by $\hat{P}$. For the targets, we consider the current pose estimate $\hat{T}_i$ and transform targets by $\hat{P}\hat{T}_i^{-1}$.

The surface distance with respect to the plane is now trivially obtained by the z-coordinate of the queried point x. For the distance between objects, the nearest neighbor in each scene object is computed per queried point. The point lies within the corresponding scene object if

$$\cos(n_y(x) \cdot (y - x)) > 0, \tag{8}$$

that is, if the normal of the nearest neighbor $n_y(x)$ points in the same direction as the vector from x to the nearest neighbor y. To make this test more robust, we compute it for the k nearest neighbors and consider the point to lie inside if it holds for a certain fraction of k. Note that a similar quorum check is used to compute the supported points S.

The final surface distance per point x is computed by taking the minimum over the distances to all scene objects (and the plane). The critical points of the queried point cloud are then computed using Equations (5)–(7), with the gravity direction given by the normal of the supporting plane.

Initially, the plausibility information derived from the computed surface distances is inaccurate, since the initial poses of the scene objects themselves are inaccurate. We therefore consider all object poses in parallel – as all poses improve, so does the scene-level plausibility information.

4.2.3 Leveraging Plausibility for Refinement

We want to exploit the additional source of information and guide the agent towards plausible poses. The most straightforward approach is to add the surface distance as additional input. Intuitively, the agent should move away from points

that have a negative surface distance (intersecting) and keep some points close to zero distance (in contact). As the surface distance is available for both source and target, this also situates the object within the scene – if we assume two points to correspond, their surface distance under the estimated pose should also correspond.

To enforce this behavior and to also provide a training signal for the full trajectory, we add a plausibility-based reward term. To reinforce actions that lead to poses under which the object rest stably within the scene, we define

$$r_p = \begin{cases} +\rho_p, & \text{if stable,} \\ -\rho_p, & \text{otherwise.} \end{cases} \tag{9}$$

Note that a necessary condition for stability is feasibility. Thus, by reinforcing stability, feasibility is implicitly considered too. This reward is combined with the alignment-based reward in Equation (2) in a discounted task. Hence, if actions lead to a stable pose later on, the reward also reinforces these earlier actions via the discounted return.

4.2.4 Rendering-based Pose Scoring

Due to imprecise segmentation, the refinement may consider close-by objects or points sampled from the background. Consequently, the alignment after the final iteration might be lower than in an intermediary step. This is all the more important as poses of scene objects influence one another. Related work [10, 3] deals with this issue by rendering-based verification of pose hypotheses.

In each iteration, the object under currently estimated pose is rendered, giving a depth image $\hat{I}_d$ and a normal image $\hat{I}_n$. These are compared to the observed depth and normal images I_d and I_n, masked by the estimated segmentation for the corresponding object. Based on [3], we define the per-pixel score $e(p)$ by

$$e(p) = (e_d + e_n)/2, \quad \forall p \in \hat{I}_d > 0 \cup I_d > 0, \tag{10}$$

where the depth and the normal scores e_d, e_n are given by

$$\begin{aligned} e_d(p) &= 1 - \min(1, |\hat{I}_d(p) - I_d(p)|/\tau_d), \\ e_n(p) &= 1 - \min(1, (1 - cos\hat{I}_n(p) \cdot I_n(p))/\tau_n), \end{aligned} \tag{11}$$

with thresholds τ_d and τ_n clamping and scaling the respective error to the range $[0, 1]$. Note that for the normal error e_n, the cosine is clamped to $[0, 1]$. The mean over the per-pixel scores is used to score the corresponding pose and the one with maximal score is returned as refinement result.

5. Experiments

We evaluate the proposed depth-based pose refinement approach in comparison with state-of-the-art methods, provide an ablation study for the proposed extensions and discuss failure cases. Qualitative examples are shown in Figure 5. Further experiments are provided in the supplement.

Datasets: The single object scenario is evaluated on the LINEMOD dataset (LM) [11], consisting of 15 test scenes showing one object in a cluttered environment each but with only minor occlusion of the target object. The test split defined in related work [6, 18, 21] is used, omitting scenes 3 and 7. The YCB-VIDEO dataset (YCBV) [25] features more complex scene-level interactions and heavy occlusion with test scenes consisting of three to six YCB objects [8]. In contrast to most competing approaches, we do not use any additional synthetic training data. Moreover, on YCBV, we only use 1/100th of the real training images.

Metrics: Evaluation metrics used in related work are the Average Distance of Model Points (ADD) and Average Distance of Model Points with Indistinguishable Views (ADI) [11]. The ADD is defined as the mean distance between corresponding model points under estimated and ground-truth pose. The ADI computes the mean distance over the nearest points and as such is better suited for symmetrical objects. The AD uses ADI for objects that are symmetrical and ADD otherwise. Note that the distinction between (non)symmetrical objects in related work also considers textural information in the color image which is not observable by our depth-based approach.

Baselines: We compare our approach to state-of-the art pose refinement methods that use RGB (DeepIM [15], PoseRBPF [10], PFRL [20], DPOD [28]), depth (Point-to-Plane ICP (P2Pl-ICP) [29], Iterative Collision Check with ICP (ICC-ICP) [22], VeREFINE [3], the ICP-based multi-hypothesis approach (Multi-ICP) in [25]) or a combination of both modalities (DenseFusion [23], RGBD versions of DeepIM [15] and PoseRBPF [10]). The RGB-based method of Shao et al. [20] (PFRL) is conceptionally close to our approach as it leverages RL to refine the pose by aligning the observed mask. In terms of used modality, the ICP-based approaches are considered for direct comparison. Note that the initialization by PoseCNN [25] only provides a single pose hypothesis; we indicate the use of additional pose hypotheses by *Multi-ICP* with *. P2Pl-ICP and VeREFINE (using P2Pl-ICP) are given a budget of 30 refinement iterations. The correspondence threshold for P2Pl-ICP, given as index in the results, is defined as fraction of the object size. By *(RGB)D* we indicate that a method uses depth for refinement but is initialized by a method that additionally uses color information.

Implementation Details: Following related work [20, 15, 10, 14], we use the instance segmentation masks and poses estimated by PoseCNN [25] for initialization.

For the refinement actions, we use step sizes of $[0.0033, 0.01, 0.03, 0.09, 0.27]$ in positive and negative direction plus a "stop" action with step size 0. Step sizes are interpreted in radians for rotation and in units of the normalized representation for translation. Rotational symmetries are sampled with a resolution of 5deg. The threshold ε

Figure 5. Initial (purple) and final pose (blue) on LINEMOD (left, cropped) and YCB-VIDEO (right). Best viewed digitally.

for computing the critical points for plausibility is experimentally determined with $1cm$. The reward function uses $\rho = (-0.6, -0.1, 0.5)$ for alignment and $\rho_p = 0.5$ for plausibility. The RL loss term is scaled by $\alpha = 0.1$ on LM and by 0.2 on YCBV. The outlier-removal loss term is scaled by $\beta = 7$. The remaining network parameters are chosen as in [4]. The segmentation masks are provided as a single channel image, missing information where they overlap, and the clamps in YCBV are confused in many frames. To compute the verification score, we thus merge both masks and use thresholds $\tau_d = 2cm$ and $\tau_n = 0.7$.

During training, we generate a replay buffer of 128 object trajectories each, delaying network updates until it is filled and sampling batches from the shuffled buffer. The training samples are augmented by an artificial segmentation error, random pose error for initialization and a random error in the pose of the plane. The segmentation error selects a random pixel within the ground-truth mask. The nearest neighbors to this pixel are determined and $p\%$ of the foreground and $100 - p\%$ of the background neighbors are sampled. This simulates occlusion and inaccurate segmentation, including parts of the background. Error magnitudes are given with the experiments on the respective dataset. We train SporeAgent for 100 epochs, starting with a learning rate of 10^{-3} and halving it every 20 epochs.

5.1. Single Object: LINEMOD

For training, we uniformly randomly choose the foreground fraction $p \in [50, 100\%]$. The initial pose is sampled by rotation about a uniformly random rotation vector by a random magnitude in $[0, 90deg]$. Similarly, the translation is in direction of a uniformly random vector of magnitude $[0.0, 1.0]$ units in the normalized space of the respective object. Additionally, the estimated plane is jittered by a random rotation in $[0, 5deg]$ and a random translation in $[0, 2cm]$, sampled as for to the initial pose.

Table 1 shows the mean class recalls for varying precision thresholds with respect to the object diameter d. As shown, SporeAgent outperforms related approaches by a large margin of 3.8% using the $0.05d$ threshold and 10.8% using the $0.02d$ as compared to the next best methods.

	AD (↑) $< 0.10d$	AD (↑) $< 0.05d$	AD (↑) $< 0.02d$	Modality
PoseCNN [25]	62.7	26.9	3.3	RGB
PFRL [20]	79.7	–	–	RGB
DeepIM [15]	88.6	69.2	30.9	RGB
P2Pl-ICP$_{0.2}$ [29]	90.6	81.2	36.8	(RGB)D
VeREFINE$_{0.2}$ [3]	95.4	87.5	*39.5*	(RGB)D
P2Pl-ICP$_{0.3}$ [29]	92.6	79.8	29.9	(RGB)D
VeREFINE$_{0.3}$ [3]	*96.1*	85.8	32.5	(RGB)D
Multi-ICP* [25]	**99.3**	*89.9*	35.6	(RGB)D
SporeAgent (ours)	**99.3**	**93.7**	**50.3**	(RGB)D

Table 1. Results on LINEMOD (mean over per-class results).

	ADD (↑) AUC	AD (↑) AUC	ADI (↑) AUC	Modality
PoseCNN [25]	51.5	61.3	75.2	RGB
CosyPose [14]	–	84.5	89.5	RGB
PoseRBPF [10]	59.9	–	77.5	RGB
DeepIM [15]	71.7	81.9	88.1	RGB
PoseRBPF [10]	**80.8**	88.5	93.3	RGBD
DeepIM [15]	*80.7*	**90.4**	**94.0**	RGBD
ICC-ICP [22]	67.5	77.0	85.6	(RGB)D
P2Pl-ICP$_{1.0}$ [29]	68.2	79.2	87.8	(RGB)D
VeREFINE$_{1.0}$ [3]	70.1	81.0	88.8	(RGB)D
Multi-ICP* [25]	77.4	86.6	92.6	(RGB)D
SporeAgent (ours)	79.0	*88.8*	*93.6*	(RGB)D

Table 2. Results on YCB-VIDEO (mean over per-class results).

5.2. Full Scene: YCB-VIDEO

As YCBV already features large amounts of occlusion, we increase the foreground fraction p to the range $[80, 100\%]$ during training. In addition to this, the training samples are more challenging and we thus reduce the initial pose error to $[0, 75deg]$ and $[0, 0.75]$ translation units. The same plane error as on LM is used.

The results in Table 2 report the Area Under the precision-recall Curve (AUC) for a varying precision threshold of $[0, 10cm]$, averaged over per-class results. On the ADI metric, our approach achieves accuracy on par with RGBD-based methods, being only second to the RGBD version of DeepIM [15]. Although our approach uses depth information and, as such, may not consider textural symmetries reflected by high ADD and AD scores, we are able to achieve higher accuracy than competing RGB-based approaches. Compared to the best other evaluated depth-based approach (Multi-ICP [25]), we improve accuracy by 1 to 2.2%. Over the best single-hypothesis depth-based approach (VeREFINE [3]), the improvement is even higher with 4.8 to 8.9%. Moreover, note that compared to the best performing learning-based approaches, we require orders of magnitude less real and no synthetic training data.

We would like to emphasize that the results reported for PoseCNN are recomputed from the poses provided by the authors of [25], as those poses serve as initialization to our method. They however differ slightly from the scores reported in their paper. Results for Multi-ICP are also computed from the poses provided by the authors of [25, 15].

	AD (↑) $< 0.10d$	AD (↑) $< 0.05d$	AD (↑) $< 0.02d$
SporeAgent	99.3	93.7	50.3
no normals	99.1	93.5	**50.8**
no outlier removal	99.3	93.6	50.1
no stability reward	99.3	93.7	50.0
no surface distance	**99.4**	**94.0**	46.3
no pose scoring	98.9	92.5	46.9
Baseline (IL)	98.8	90.6	39.7

Table 3. Ablation on LINEMOD.

	ADD (↑) AUC	AD (↑) AUC	ADI (↑) AUC
SporeAgent	**79.0**	**88.8**	**93.6**
no canonical	76.4	86.7	92.6
no symmetry	78.2	88.0	93.4
no outlier removal	78.6	88.5	93.5
no stability reward	78.2	88.1	93.4
no surface distance	77.3	87.3	92.9
no pose scoring	78.3	88.1	93.0
Baseline (IL) + normals	72.2	82.2	90.7

Table 4. Ablation on YCB-VIDEO.

fraction	ADD (↑) AUC	AD (↑) AUC	ADI (↑) AUC
1/400th	75.4	86.2	92.5
1/200th	77.9	87.9	93.3
1/100th	**79.0**	**88.8**	**93.6**

points	ADD (↑) AUC	AD (↑) AUC	ADI (↑) AUC
256	77.9	87.8	93.3
512	78.6	88.4	93.5
1024	**79.0**	**88.8**	**93.6**

Table 5. Results by fraction of training split used (left) and number of points sampled for refinement (right) on YCB-VIDEO.

5.3. Ablation Study

To evaluate the impact of each of the proposed parts of our method, we train separate models on LM and YCBV with each part removed individually.

Table 3 shows results on LM. We see that the scene-level information is essential to achieve high recall under restrictive thresholds. Moreover, the rendering-based scoring is able to determine the best-aligned intermediary pose – without the scoring, the last iterations jitter between tight alignment and observational noise.

The ablation study on YCBV, shown in Table 4, highlights the benefit of representing the target point clouds in a canonical frame. As on LM, the consideration of the physical plausibility improves accuracy. The scene-level information is able to support the refinement, even when computed from initially inaccurate object poses.

Furthermore, we evaluate the impact of the number of training samples and point samples used on YCBV. As shown in Table 5 (left), the performance of Multi-ICP [25] is achieved already using 1/400th of the available real training images. Using 1/100th, SporeAgent outperforms RGBD-based PoseRBPF on the AD AUC and ADI AUC metrics. The number of points may be reduced to improve runtime and memory footprint while maintaining high accuracy, indicated by the results in Table 5 (right).

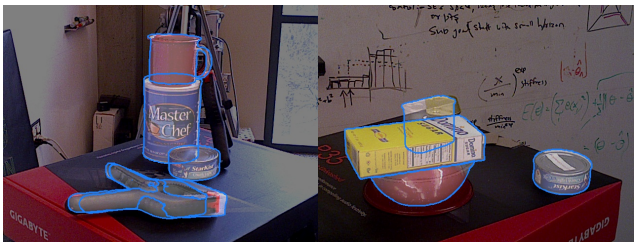

Figure 6. Failure cases on YCB-VIDEO. Best viewed digitally.

5.4. Analysis of Failure Cases and Future Work

Nevertheless, we observe specific systematic failure cases that could not be resolved by considering physical plausibility. For example, as shown in Figure 6, the two differently sized clamps in YCBV are typically confused by PoseCNN (our initialization) and may thus end-up stuck within one another. Similarly, the bowl in YCBV may be estimated in an upside-down pose due to occlusion. In this scenario, refinement gets stuck between the plane and the scene object resting on the bowl. We hypothesize that such systematic errors can be best addressed by tighter integration of detection, pose estimation and refinement, as for example proposed by Labbé et al. [14] for CosyPose.

In addition, when inaccurate segmentation, occlusion or missing depth values (e.g., of metallic surfaces like caps of cans) decimate the number of foreground points in the source point cloud too much, accuracy of our approach significantly drops as the remainder of the source mostly consists of background points. Additional consideration of color information might lessen this issue.

Further robustness may be achieved by consideration of multiple pose hypotheses per instance, as proposed in related work [14, 16, 3]. For example, in the case of the confused clamp classes, one hypothesis per class might be refined and the most probable one selected using the rendering-based scoring.

6. Conclusion

We present *SporeAgent*, a reinforcement learning approach to object pose refinement. SporeAgent jointly refines the poses of multiple objects in an observed depth frame in parallel. By considering object symmetries and physical plausibility of the scene, we achieve state-of-the-art results as shown in our evaluation on the LINEMOD and YCB-VIDEO datasets. The provided ablation study illustrates the benefit of each proposed part and our analysis of failure cases motivates future improvements towards further robustness to noise and inaccuracy in detection, segmentation and initialization.

Acknowledgements: This work was supported by the TU Wien Doctoral College TrustRobots and the Austrian Science Fund (FWF) under grant agreements No. I3968-

N30 HEAP and No. I3969-N30 InDex.

References

[1] Yasuhiro Aoki, Hunter Goforth, Rangaprasad Arun Srivatsan, and Simon Lucey. Pointnetlk: Robust & efficient point cloud registration using pointnet. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7163–7172, 2019.

[2] Dominik Bauer, Timothy Patten, and Markus Vincze. Physical plausibility of 6d pose estimates in scenes of static rigid objects. In *Eur. Conf. Comput. Vis. Workshops*, pages 648–662, 2020.

[3] Dominik Bauer, Timothy Patten, and Markus Vincze. Verefine: Integrating object pose verification with physics-guided iterative refinement. *IEEE Robot. Autom. Lett.*, 5(3):4289–4296, 2020.

[4] Dominik Bauer, Timothy Patten, and Markus Vincze. Reagent: Point cloud registration using imitation and reinforcement learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 14586–14594, 2021.

[5] Paul J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256, 1992.

[6] Eric Brachmann, Frank Michel, Alexander Krull, Michael Ying Yang, Stefan Gumhold, et al. Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3364–3372, 2016.

[7] Benjamin Busam, Hyun Jun Jung, and Nassir Navab. I like to move it: 6d pose estimation as an action decision process. *arXiv preprint arXiv:2009.12678*, 2020.

[8] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. The ycb object and model set: Towards common benchmarks for manipulation research. In *Int. Conf. on Adv. Robot.*, pages 510–517. IEEE, 2015.

[9] Christopher Choy, Wei Dong, and Vladlen Koltun. Deep global registration. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2514–2523, 2020.

[10] Xinke Deng, Arsalan Mousavian, Yu Xiang, Fei Xia, Timothy Bretl, and Dieter Fox. Poserbpf: A rao–blackwellized particle filter for 6-d object pose tracking. *IEEE Trans. Robot.*, 2021.

[11] 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 *ACCV*, pages 548–562, 2012.

[12] Tomáš Hodaň, Martin Sundermeyer, Bertram Drost, Yann Labbé, Eric Brachmann, Frank Michel, Carsten Rother, and Jiří Matas. BOP challenge 2020 on 6D object localization. *Eur. Conf. Comput. Vis. Workshops*, 2020.

[13] Alexander Krull, Eric Brachmann, Sebastian Nowozin, Frank Michel, Jamie Shotton, and Carsten Rother. Poseagent: Budget-constrained 6d object pose estimation via reinforcement learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6702–6710, 2017.

[14] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. Cosypose: Consistent multi-view multi-object 6d pose estimation. In *Eur. Conf. Comput. Vis.*, pages 574–591, 2020.

[15] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. Deepim: Deep iterative matching for 6d pose estimation. In *Eur. Conf. Comput. Vis.*, pages 683–698, 2018.

[16] Chaitanya Mitash, Abdeslam Boularias, and Kostas E Bekris. Improving 6d pose estimation of objects in clutter via physics-aware monte carlo tree search. In *Int. Conf. Robot. Autom.*, pages 3331–3338, 2018.

[17] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 652–660, 2017.

[18] 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 *Int. Conf. Comput. Vis.*, pages 3828–3836, 2017.

[19] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[20] Jianzhun Shao, Yuhang Jiang, Gu Wang, Zhigang Li, and Xiangyang Ji. Pfrl: Pose-free reinforcement learning for 6d pose estimation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 11454–11463, 2020.

[21] Bugra Tekin, Sudipta N Sinha, and Pascal Fua. Real-time seamless single shot 6d object pose prediction. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 292–301, 2018.

[22] Kentaro Wada, Edgar Sucar, Stephen James, Daniel Lenton, and Andrew J Davison. Morefusion: multi-object reasoning for 6d pose estimation from volumetric fusion. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 14540–14549, 2020.

[23] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3343–3352, 2019.

[24] Yue Wang and Justin M Solomon. Deep closest point: Learning representations for point cloud registration. In *Int. Conf. Comput. Vis.*, pages 3523–3532, 2019.

[25] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. In *Robot.: Sci. Syst.*, 2018.

[26] Zi Jian Yew and Gim Hee Lee. Rpm-net: Robust point matching using learned features. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 11824–11833, 2020.

[27] Wentao Yuan, Ben Eckart, Kihwan Kim, Varun Jampani, Dieter Fox, and Jan Kautz. Deepgmr: Learning latent gaussian mixture models for registration. *arXiv preprint arXiv:2008.09088*, 2020.

[28] Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. Dpod: 6d pose object detector and refiner. In *Int. Conf. Comput. Vis.*, pages 1941–1950, 2019.

[29] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3d: A modern library for 3d data processing. *arXiv preprint arXiv:1801.09847*, 2018.