IEEE Robotics and Automation Letters (RAL) paper presented at the
2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

VeREFINE: Integrating Object Pose Verification with Physics-guided Iterative Refinement

Dominik Bauer, Timothy Patten and Markus Vincze

***Abstract*— Accurate and robust object pose estimation for robotics applications requires verification and refinement steps. In this work, we propose to integrate hypotheses verification with object pose refinement guided by physics simulation. This allows the physical plausibility of individual object pose estimates and the stability of the estimated scene to be considered in a unified optimization. The proposed method is able to adapt to scenes of multiple objects and efficiently focuses on refining the most promising object poses in multi-hypotheses scenarios. We call this integrated approach *VeREFINE* and evaluate it on three datasets with varying scene complexity. The generality of the approach is shown by using three state-of-the-art pose estimators and three baseline refiners. Results show improvements over all baselines and on all datasets. Furthermore, our approach is applied in real-world grasping experiments and outperforms competing methods in terms of grasp success rate. Code is publicly available at github.com/dornik/verefine.**

I. Introduction

Autonomous robots need to interact with their physical environment to fulfill a plethora of tasks. This requires the manipulation of individual objects, for example, to fetch an item or to stow it away. A popular approach to enable such manipulations uses object pose estimation and grasp pose annotation [1], [2], [3], [4]. Previous work on object detection and pose estimation achieves high accuracy on popular datasets such as `LINEMOD` [5] or `YCB-VIDEO` [6]. However, the performance of these algorithms deteriorates when the objects' 3D models are inaccurate or lighting and viewing conditions change [7], [8]. To deal with this problem, hypotheses verification and object pose refinement are commonly used in object pose estimation pipelines.

The idea of hypotheses verification is to evaluate the fit of different estimates with the observed scene: The best fitting estimates are selected and estimates below a threshold are pruned. While this improves accuracy and reliability it also introduces the problem of increased complexity arising from the number of possible combinations in multi-object scenes. The usability of such approaches in robotics is limited by their runtime. For example, Mitash et al. [9] combine object pose verification with physics simulation, resulting in frame times of up to 30s. Krull et al. [10] integrate object pose refinement and verification with reinforcement learning to efficiently allocate a refinement budget. However, the authors report a frame time of up to 34s.

This work was supported by the TU Wien Doctoral College TrustRobots and the Austrian Science Fund (FWF) under grant agreement No. I3968-N30 HEAP and No. I3969-N30 InDex.

The authors are with the Vision for Robotics Laboratory, Automation and Control Institute, TU Wien, 1040 Vienna, Austria (e-mail: {bauer,patten,vincze} @acin.tuwien.ac.at).

Fig. 1: Grasping `YCB-VIDEO` objects with a Toyota HSR. Initial pose estimates using [4] in simulation environment (top) are improved using VeREFINE (mid and bottom).

In contrast to hypothesis verification that only accepts or rejects object pose estimates, object pose refinement improves the estimates themselves. This is achieved by minimizing the discrepancy between the observed scene and the object in an estimated pose. The most popular pose refinement method is the Iterative Closest Point (ICP) algorithm [11]. However, if the initial estimate or the visual observation are inaccurate, ICP converges to a local minimum (wrong pose) or even diverges. Alternatively, physics simulation has been used to ensure plausibility and improve accuracy [9], [12] of object pose estimates. But applying physics simulation to objects is an unstable process. It may cause objects to topple over and create worse estimates given the inaccuracy of the simulated environment.

The goal of both verification and refinement is to maximize the fit of the estimate to the observed scene. We hypothesize that, by integrating these approaches into one step, we are able to improve the overall accuracy of the pose estimates, while achieving more graceful degradation by limiting divergence of individual strategies. To this end we present VeREFINE, an integrated approach that combines hypotheses *Ver*ification, object pose *Refine*ment and physics simulation into a unified framework. Our contributions are

- improving accuracy by integrating refinement with physics simulation in an iterative loop,
- improving robustness by efficient rendering-based veri-

fication of object pose estimates,
- improving accuracy and runtime using regret minimization to exploit promising hypotheses, and
- the combination into reliable scene-level refinement and verification for multi-object scenes.

We evaluate our framework on three publicly available datasets, `Extended APC` [9], `LINEMOD` [5] and `YCB-VIDEO` [6], and out-perform state of the art in pose estimation and refinement in terms of robustness and accuracy. We compare to the related approach by Mitash et al. [9], achieving a significant reduction in runtime while increasing the accuracy of the pose estimates. We demonstrate the robustness of our method with respect to initial pose errors and missing depth values due to occlusion and material properties. Finally, we evaluate the proposed framework in a robotic grasping experiment resulting in significantly increased success rates compared to other methods.

After reviewing related work in Sec. II, we discuss the refinement methods in Sec. III and the complete VeREFINE approach in Sec. IV. Sec. V presents experiments and results. Sec. VI concludes the paper.

II. Related Work

The proposed approach builds on previous work in hypotheses verification, object pose refinement and their combination with physics simulation.

Hypotheses verification approaches for object pose estimation show that considering multiple pose hypotheses per object improves overall estimation performance. Drost et al. [13] use a clustering-based verification stage to refine pose estimates. In [14], a pool of 200 object pose hypotheses is generated using a Point Pair Features (PPF) pipeline. Each hypothesis is refined using Projective ICP and a two-step verification to determine the best estimate. In [6], an initial estimate is perturbed to generate a set of hypotheses for better coverage of the solution space. All hypotheses are refined before scoring and selection. In contrast, Wang et al. [4] estimate a pose confidence score jointly with per-pixel object pose estimates. The highest scoring estimate is selected and refined. Krull et al. [10] train a CNN to predict two different hypotheses scores for use during refinement and for the selection of the final estimate. On the scene level, a scoring function that considers geometrical cues, clutter and conflicting hypotheses for multiple objects is proposed in [15]. For efficient evaluation of the search space, [16] consider equivalent combinations of hypotheses to reduce the search tree to a directed acyclic graph and explore using Monte Carlo Tree Search (MCTS). Physics simulation is incorporated in MCTS to additionally consider the supporting relations between objects in [9], [17]. We propose to apply rendering-based verification to guide refinement, allowing refinement steps to be allocated to promising hypotheses. This naturally extends to multi-object scenes, which reduces the solution space as compared to search-based methods.

Previous work on object pose refinement exploit depth, RGB and object segmentation as input modalities. A seminal approach is ICP [11]. More recently, deep learning approaches for object pose refinement have been proposed. RGB-based methods render intermediary object pose estimates and use CNNs to compute a pose update [18], [19], [20]. The refinement method by Wang et al. [4] requires RGB-D images and instance segmentation as input. The depth cues are processed using a PointNet and combined with the RGB-based features from a CNN. We show that our proposed approach is applicable to both learning-free and learning-based methods. It boosts their performance by improving initial estimates using physics simulation and guides refinement through rendering-based verification.

Application of physics reasoning and simulation in related vision tasks indicates that it creates strong cues for object pose and admissible scene configurations. The segmentation method by Jia et al. [21] uses rule-based physical stability reasoning to combine or split candidate patches, represented by bounding boxes, to generate physically plausible scenes. A similar reasoning is applied to voxelized scene representations to segment and estimate the shape of objects in [22]. In a robotics context, Furrer et al. [12] show the benefit of using physics simulation for object stacking. They propose a method for determining the target pose of irregular stones such that a structurally stable stack can be built by a robot. Mitash et al. [9] use physics-based verification and MCTS for object pose estimation given multiple hypotheses in multi-object scenes. For each hypotheses combination, this approach runs one iteration of Trimmed ICP and a physics simulation, making it sensitive to the estimation of the supporting plane and the physical properties of the simulated objects. Our proposed solution of interleaving physics simulation and refinement is more robust to these challenges and prevents diverging simulation. We allow more promising estimates to be refined multiple times while saving these additional iterations on less promising estimates. Moreover, in [9], a solution is processed one object after another. Feedback on the impact on the overall solution quality is given by a scene-level reward but only allows to select among the refined hypotheses. In contrast, by incorporating the scene-level feedback in the refinement process, our approach adapts the estimates to the overall solution. Furthermore, the approach of [9] needs to grow a search tree of combinations of hypotheses, spending expensive refinements on exploring the search space. More efficiently, our approach uses an object-based representation of the search space, which is initialized using a rendering-based verification score. Thereby, no additional computation needs to be spent on initial exploration of the search space.

III. Integrating Hypotheses Verification with Physics-guided Iterative Refinement

The goal of this work is to accurately and robustly explain scenes of varying complexity in terms of object poses for applications such as robotic grasping. An RGB-D observation, instance detection, instance segmentation and a set of initial object pose estimates are assumed to be given.

In the following, we present the building blocks of our VeREFINE approach by considering increasingly complex

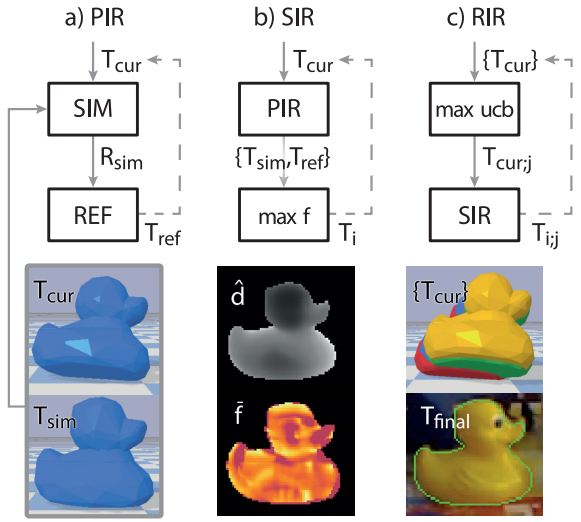

Fig. 2: Proposed integration approaches given a simulation environment and an initial object pose estimate (T_{cur}). (a) Integration of physics simulation (SIM) and iterative refinement (REF) into Physics-guided Iterative Refinement (PIR). (b) Supervision using verification score $\bar{f}$ (SIR). (c) Regret minimization using UCB score (RIR).

scenarios. For individual objects, we propose an iterative physics-guided refinement loop (Sec. III-A). To improve the robustness of this approach, supervision of the refinement loop through rendering-based verification is presented (Sec. III-B). Given multiple estimates, a regret minimization approach is introduced to efficiently allocate refinement towards promising estimates (Sec. III-C). We extend the discussed methods to consider multi-object scenes with multiple initial estimates each, where occlusion and support relationships between objects need to be considered (Sec. IV).

A. Physics-guided Iterative Refinement (PIR)

Object pose refinement methods depend heavily on the quality of the initial estimates. In contrast to previous approaches that apply physics simulation as a post-hoc step after iterative refinement [9], we propose to interleave object pose refinement and physics simulation in a Physics-guided Iterative Refinement (PIR) loop, illustrated in Figure 2a. The physical plausibility of the initial estimate used for refinement is improved using simulation, helping the refinement to relate the correct parts of the model to the observation. The iterative feedback loop allows the refinement to, in turn, initialize physics simulation with better estimates, thus limiting divergence.

In each iteration, the current object pose estimate $T_{cur} = [R_{cur}, t_{cur}]$ initializes the object in the simulation environment, shown in Figure 2a (mid). In the simplest case, the environment consists of a supporting plane. In more complex scenes, it also includes other estimated objects. The simulation is progressed and the resulting object pose T_{sim} is returned. As indicated in Figure 2a, only the orientation part R_{sim} is used to update the estimate. This is motivated by the observation that, when physics simulation leads to large displacements, it causes the iterative refinement to lose track of corresponding object parts. We found only using the orientation contains this divergent behavior while still improving the refinement process. The estimate $[R_{sim}, t_{cur}]$ is used to initialize an iteration of the object pose refinement algorithm that returns the final estimate T_{ref} after one iteration of PIR. In the experiments, multiple iterations of PIR are used to obtain T_{ref}.

B. Supervised Iterative Refinement (SIR)

Due to divergent behavior in physics simulation or iterative refinement, the final estimate after applying these methods might generate a worse explanation of the observation than the initial or intermediary estimates. We solve this by continuously evaluating the observation fit of the intermediary estimates. This integration of verification into the refinement process allows us to supervise divergent behavior and select the best fitting estimate as the final one. The verification score $\bar{f}$ measures the observation fit and is computed from the average discrepancy between the estimate and the observation in terms of depth and surface normals, given by

$$\bar{f}(T) = \frac{1}{2}\left(\frac{1}{N}\sum^{N} f_d(T) + \frac{1}{N}\sum^{N} f_{\mathbf{n}}(T)\right)$$
$$f_d(T) = \begin{cases} 1 - \frac{|d-\hat{d}_T|}{\tau}, & \text{where } |d-\hat{d}_T| < \tau \\ 0, & \text{otherwise} \end{cases} \quad (1)$$
$$f_{\mathbf{n}}(T) = \begin{cases} 1 - \frac{1-\mathbf{n}\cdot\hat{\mathbf{n}}_T}{\alpha}, & \text{where } 1-\mathbf{n}\cdot\hat{\mathbf{n}}_T < \alpha \\ 0, & \text{otherwise} \end{cases}$$

where d is a valid depth value and $\mathbf{n}$ is a corresponding surface normal in the segmented scene. $\hat{d}_T$ and $\hat{\mathbf{n}}_T$ are the N corresponding pixel values obtained by rendering depth and normal images of the object under the estimated pose using OpenGL. Parameters τ and α are soft thresholds for the maximal admissible discrepancy. Figure 2b (bottom) shows an example of $\bar{f}$ applied to an estimate.

In each PIR iteration i, we evaluate the estimates returned by physics simulation $T_{i;sim}$ and refinement $T_{i;ref}$ and proceed with the estimate that achieves the better score. After the last iteration, the final estimate T that gives the best score $\bar{f}$ overall is selected from all processed estimates. As such, in cases where the individual approaches could diverge, Supervised Iterative Refinement (SIR) can recover to the highest scoring intermediary estimate.

The supervision requires evaluations of $\bar{f}$ for $T_{i;sim}$ and $T_{i;ref}$ each iteration. To enable fast evaluation, computations are carried out on the GPU in two rendering passes using OpenGL. The first pass writes $\hat{d}_T$ and $\hat{\mathbf{n}}_T$ to a texture. The second pass uses this texture and the observation to compute f_d and f_n. The summed values of N, f_d and f_n are read-back from a higher-level mipmap, drastically reducing the read-back time. The final averaging is done on the CPU and yields $\bar{f}$. In our experiments, one evaluation of $\bar{f}$ using a NVIDIA GTX 1080Ti takes 1-2ms. This is a significant speed-up compared to 7-9ms when reading-back the full depth and normal information from the GPU to evaluate $\bar{f}$ on the CPU.

C. Regret-minimizing Iterative Refinement (RIR)

Considering multiple pose hypotheses per object raises the questions: On which hypotheses to spend refinement steps and which hypothesis to select in the end. Promising hypotheses should be exploited by applying more refinement steps while other hypotheses should still be explored to find better candidates.

We propose a Multi-armed Bandit (MAB) to model this exploitation-exploration problem, where the pull of arm j represents running one SIR iteration for hypothesis j. The Upper Confidence Bound policy (UCB) [23] minimizes the regret of choosing a sub-optimal arm of a MAB with respect to a given reward. In each iteration, the arm with maximal ucb_j is selected according to

$$ucb_j = \mu_j + c \cdot \sqrt{\frac{\ln p}{n_j}} \tag{2}$$

where μ_j is the mean reward of playing arm j, p is the total number of plays and n_j is the number of times the arm has been played. c is a parameter of the algorithm that controls the balance of exploitation and exploration. With $c > 0$, all hypotheses are eventually explored in the limit. The larger the μ of the best known hypotheses as compared to the others, the more it will be exploited. As a result, the RIR converges towards refining the best known hypothesis if its mean reward is significantly larger. However, if multiple hypotheses yield a similar reward, the RIR will alternate between them. In this case, reducing c forces RIR to exploit the best hypothesis even if the reward difference is small. The choice of c is thus dependent on the variation of hypotheses: If the underlying pose estimator yields similar hypotheses, it is beneficial to force exploitation using a small c. If the hypotheses expose high variance, a higher c and thus more exploration yields, in general, better results.

In our approach, illustrated in Figure 2c, the verification score $\bar{f}$ is chosen as reward function. Applying the UCB policy to the resulting reward statistics efficiently allocates a fixed refinement budget, spending more refinements on promising hypotheses while saving refinements on those that have a low $\bar{f}$. This results in the same total amount of refinements but in a regret-minimizing way. The resulting Regret-minimizing Iterative Refinement (RIR) procedure starts by ranking the initial estimates based on $\bar{f}$. For each subsequent RIR iteration, SIR is applied and the verification score is used as reward signal. As with SIR, the final selection is based on the observation fit across all encountered estimates.

The formulation based on a MAB and a rendering-based score allows our approach to be quickly applied to new datasets and can be used to extend existing and future refinement methods. In contrast, the related approach in [10] uses reinforcement learning and a CNN-based verification score regression, which need to be expensively re-trained.

IV. Physics Simulation and Regret Minimization in Cluttered Multi-Object Scenes

In cluttered multi-object scenes, the proposed verification score and physics simulation need to deal with occlusions and support relationships. Thus, the order in which objects are considered is important. Moreover, with each of the N objects having n hypotheses, the number of combinations of hypotheses grows exponentially. To tame this problem, we discuss clustering strategies to reduce the number of combinations that need to be considered and present two approaches to efficiently evaluate the remaining search space.

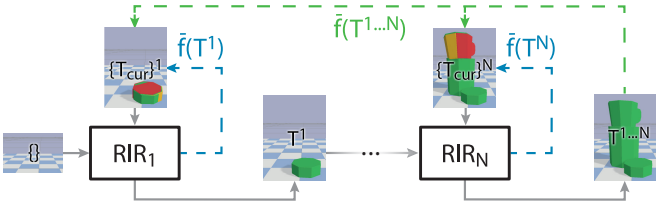

Fig. 3: Proposed approaches for cluttered multi-object scenes. The best estimate per object is added to the simulation environment used for the subsequent objects, allowing consideration of occlusions and support relationships. VF_b (blue) fully refines each object using the object fit as reward. VF_d (green) repeats this process iteratively, refining each object only once per iteration and uses the scene fit as reward.

A. Object Clustering and Dependency Graph

Mitash et al. [9] isolate objects that might interact based on the segmented point clouds. This reduces the number of objects that need to be jointly considered and thus the number of combinations. Furthermore, they argue that not all combinations of objects have to be considered. Instead, occlusion and support relationships between objects are used to compute a dependency list. A search tree is built from this list, where at layer i, object i is represented by all of its n hypotheses. This yields a tree of $(n^{N+1}-1)/(n-1)$ nodes. For a scene of 5 objects with 5 hypotheses each this produces a search tree of 3905 nodes (excluding the root node).

In contrast, we address more general scenarios by explicitly considering ambiguous dependencies, for example, the case where an object is occluded by another object but also supporting the same object. To resolve such ambiguities, we first decompose the independent clusters into support dependency lists. The first object in each support dependency list, the base object, is assumed to be in contact with the ground plane and supports the remaining objects in the list. The support dependency lists are then ordered front-to-back based on their respective base objects. Instead of using the resulting dependency list to grow a search tree using MCTS as in [9], we exploit our single-object approaches to reduce the solution space and allow for iterative refinement on a scene level. The proposed representation requires only $N \cdot n$ nodes to represent the same search space as before – or only 25 instead of 3905 nodes in the example.

B. VeREFINE breadth (VF_b)

Given an ordering, as determined in Sec IV-A, we explore all object hypotheses by representing each object in the dependency list and its hypotheses using a RIR bandit. The scene is incrementally built by computing the best estimate

for the considered object in the current environment. The object is added to the environment with the computed pose, allowing more accurate estimation of the next objects' poses. We call this approach of first exploring all hypotheses per object *VeREFINE breadth* (VF_b), shown in Figure 3 (blue). This results in N RIR bandits with n nodes each.

C. VeREFINE depth (VF_d)

An alternative approach, and to introduce a feedback loop that is missing in VF_b, is to iterate through the dependency list. For each iteration, the objects' RIR bandits are progressed only once. The best known hypotheses after each iteration are evaluated as a complete scene. The resulting scene fit is computed by $\bar{f}$ and is used as reward for selected hypotheses instead of the per-object reward. Thereby, hypotheses that contribute to a better overall scene fit are selected more often. This scene-first approach, called *VeREFINE depth* (VF_d), is illustrated in Figure 3 (green). As the procedure results in a changing reward distribution, the UCB policy is replaced with Discounted-UCB (D-UCB) [24]. The reward and plays statistics are discounted by a small factor each iteration, which reduces the impact of previous iterations and adapts to a changing reward distribution over time. This is shown to reduce the cumulative regret of the D-UCB policy as compared to UCB for abruptly and continuously changing reward distributions [25].

The RIR bandits are initialized using the rendering-based verification score as in the single-object scenario, acting as a heuristic in the first iteration through the dependency list to select better initial estimates. Therefore, instead of spending refinement steps to grow the search tree as in the MCTS-based approach [9], both our proposed approaches efficiently allocate refinement steps to more promising estimates.

V. Experiments

This section presents the evaluation of VeREFINE on the `Extended APC` (xAPC), `LINEMOD` (LM) and `YCB-VIDEO` (YCBV) datasets. Improvement over state-of-the-art refinement methods is shown by comparison with Iterative Closest Point (ICP) and DenseFusion Refinement (DF-R). For pose estimation, we use Point Pair Features (PPF) and DenseFusion (DF). In addition, we compare against the approach by Mitash et al. [9] (PHYSIM-MCTS). It uses Super4PCS (PCS) and hypotheses clustering for pose estimation and Trimmed ICP (TrICP) for refinement. The impact of the individual parts of our method is evaluated in an ablation study on LM.

Datasets: The LM dataset [5] is used to evaluate the single-object setting. It consists of 15 scenes with individual toys and household objects. A test set is defined based on the BOP19 challenge [26], albeit adapted to learning-based methods. These methods use the training split defined in [27], [28], [29], which excludes scenes 3 and 7 but includes 15% of the test frames used in [26]. We therefore exclude both scenes and the frames used in training from the test set for a total of 2219 test frames. xAPC [9] and YCBV [6] are used for the multi-object setting. Both datasets exhibit clutter as well as isolated, 2- and 3-object support relationships. xAPC uses Amazon Picking Challenge objects and features three objects per scene. The whole dataset is used for testing. YCBV contains 92 scenes. The 12 test scenes consist of 3 to 6 objects from the YCB object set [30]. The test set defined in [26] is used for our evaluation.

Metrics: The procedure defined for the BOP 2019 challenge [26] is used for evaluation. This considers three different error functions, namely, the Maximum Symmetry-Aware Projection Distance (MSPD), the Maximum Symmetry-Aware Surface Distance (MSSD) and the Visible Surface Discrepancy (VSD). The reported values per error function are the average recall rates over 10 thresholds in percent. The overall performance score (AR) is the average recall rate over all sub-scores. On xAPC, we additionally report the average rotation and translation errors for comparison with [9].

Baselines: Mitash et al. [9] (PHYSIM-MCTS) evaluate on the xAPC dataset. For comparability, we use the code provided by the authors to generate bounding boxes, a pool of 25 hypotheses per object and the results reported for their method. A maximum of 150 TrICP iterations is used for evaluation of all approaches. Note that, for PHYSIM-MCTS, we only count the refinement iterations in the expansion step to ensure a fair comparison. The best performing methods on LM are the PPF-based methods by Vidal et al. [14] and Drost et al. [13]. As neither provide code, we use the code of a comparable PPF-based method by Alexandrov et al. [31] to produce a pool of hypotheses. We train Mask R-CNN [32] to provide detections and segmentation masks. In addition, we evaluate the RGB-D-based method DenseFusion [4]. It features a fast inference time and a learning-based refinement method. Precomputed detections and segmentation masks by [6] are used. A pool of object pose estimates is generated using the provided code and weights. The hypotheses pool consists of the highest confidence per-pixel estimate and additional uniformly-random sampled estimates.

We set the parameters for the verification score in Equation (1) to $\tau = 20$mm and $\alpha = 45$deg on all datasets. PyBullet [33] is used as physics simulator with a time-step of 1/60sec, 10 solver iterations, 4 sub-steps and assuming an equal mass of 1kg for all objects. 3D plane segmentation is employed to determine a supporting plane and its normal is used to compute the gravity direction.

The generality of our approach is shown by applying it to three baseline iterative refinement approaches, namely, TrICP, point-to-point ICP and DF-R. TrICP uses the implementation in PCL [34] with the same settings as [9]. The simulation uses 60 steps in this case. We use the basic point-to-point ICP implementation from PCL with 50 iterations. For our approaches, we distribute the ICP iterations evenly over 5 PIR iterations. DF-R uses the weights provided by [4], trained to use 2 iterations. They are distributed over 2 PIR iterations. As ICP and DF-R are found to be more sensitive to interference with the iterative refinement, only 3 simulation steps are used.

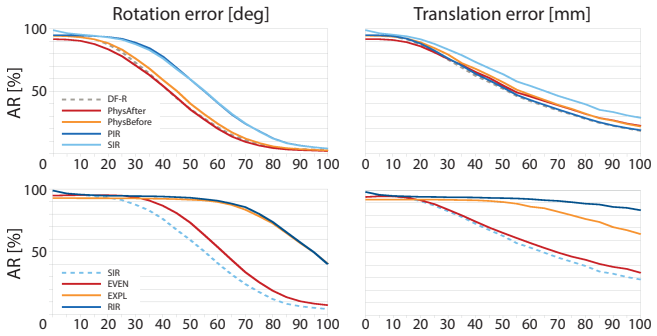

Fig. 4: Ablations on LM using single hypotheses (top) and 5 hypotheses (bottom). EVEN and EXPL use our verification score to determine the best estimate and PIR for refinement. PhysBefore and PhysAfter apply simulation before and after refinement. AR values at 5mm/deg steps are reported and linearly interpolated in between.

A. Ablation Study

The following ablations aim to motivate several design choices. The experiments start with the ground-truth annotations of the LM dataset as initial estimates and introduce errors of increasing magnitude. For the ablation, the ground-truth ground plane is used for physics simulation. Two types of errors are applied. (1) Rotation error is created by uniformly-random sampling a rotation axis from the unit sphere and rotating the ground-truth estimate by a varying angle about this axis. (2) Translation error is introduced by offsetting the ground truth by a translation vector that is sampled from the unit sphere, scaled by a varying distance.

1) Physics Simulation and Iterative Refinement: As shown in Figure 4 (top), our interleaved approach to combine physics simulation with refinement (PIR) is consistently the best performing simulation approach under rotation error. For translation error, it is limited as it only considers the rotation part from simulation to contain divergence. The benefit of using only rotation is illustrated by comparison with applying full simulation after refinement (PhysAfter) as used in [9]. Rotation error in the initial estimate causes this approach to diverge and perform even worse than the baseline method (DF-R) without physics simulation. Figure 4 (top) also shows the benefit of supervising the refinement process. Our approach (SIR) consistently improves the accuracy of pose estimates, most notably under translation error.

2) Regret Minimization: There are two major approaches to deal with multiple hypotheses. The first is to score all initial hypotheses, exploiting only the best scoring hypothesis for refinement (EXPL). The second approach is to refine all hypotheses evenly and selecting the best scoring refined hypothesis (EVEN). As shown in Figure 4 (bottom), EVEN performs well for low error magnitudes while EXPL is robust to high error magnitudes. Our regret-minimizing approach (RIR) balances between these two extreme approaches and is thus able to outperform the alternatives. Moreover, a comparison with SIR shows the benefit of considering multiple hypotheses.

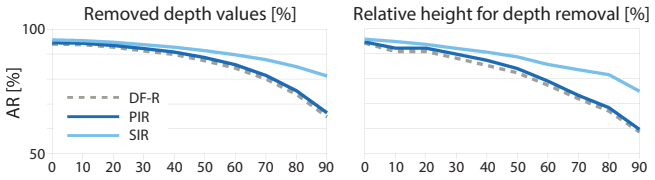

Fig. 5: Robustness study on LM using single hypotheses with a fixed error magnitude of 5mm and 5deg. AR values are measured every 10% and linearly interpolated in between.

TABLE I: Comparison using DF-R[4] and ICP[11] on `LINEMOD`.

DF	VSD	MSPD	MSSD	AR	T[ms]	#ref/obj
[4]	70.6	76.8	77.7	75.0	2	2
PIR	73.3	79.3	80.1	77.6	4	2
SIR	74.0	85.9	86.4	82.1	14	2
[4]	76.9	82.6	82.9	80.8	11	10
RIR	**78.3**	**89.7**	**89.6**	**85.9**	48	10
PPF	**VSD**	**MSPD**	**MSSD**	**AR**	**T[ms]**	**#ref/obj**
[11]	79.8	93.2	93.0	88.7	248	50
PIR	78.1	92.1	92.1	87.4	274	50
SIR	79.9	93.7	93.2	88.9	302	50
[11]	80.0	93.4	93.2	88.9	617	150
RIR	**81.0**	**95.1**	**94.5**	**90.2**	892	150

B. Robustness Analysis

To highlight the robustness of our approach, we perform experiments with missing depth values to consider two types of errors. (1) Occlusion is simulated by removing rectangular patches that are centered at uniformly-random sampled positions of the observed object. (2) Missing parts of objects from the depth channel, e.g., due to reflective material, are considered by removing depth values that correspond to the object above a certain height. Error is introduced similar to the ablation study but kept fixed at 5mm for translation and 5deg for rotation. The depth error increases from 0 to 90%.

As shown in Figure 5, our approach increases robustness to both types of error in comparison to the baseline. This indicates that the remaining depth information, together with physics simulation, limit the degradation of performance.

C. Comparison to State of the Art

1) Single-Object Scenario: The single-object scenario is evaluated on the LM dataset using DF and PPF as object pose estimators and DF-R and ICP as refinement methods. The refiners are run for the same number of iterations for comparison with RIR. Results are shown in Table I.

The performance of PIR indicates that physics simulation is beneficial given less accurate initial estimates using DF as compared to PPF. This agrees with our hypothesis that simulation improves implausible initial estimates while being vulnerable to divergence in inaccurate simulation environments. The biggest relative improvement is achieved by SIR, improving over DF-R by 7.1% AR. As indicated by the results using PPF, SIR is able to limit the divergence of the physics simulation observed for PIR. The top-performing approach in both conditions is RIR, improving over the

TABLE II: Comparison with Mitash et al. [9] using Trimmed ICP [34] on `Extended APC` with 150 iterations each.

PCS	VSD	MSPD	MSSD	AR	$\bar{r}$ [deg]	$\bar{t}$ [cm]	T[s]
[9]	48.5	51.6	68.3	56.2	**5.7**	1.3	29.9
RIR	51.8	52.0	63.0	55.6	10.5	1.4	5.5
VF_b	54.4	54.3	66.7	58.5	8.0	**1.2**	5.5
VF_d	**56.7**	**57.3**	**69.6**	**61.2**	7.5	**1.2**	6.2

TABLE III: Comparison using DF-R[4] on `YCB-VIDEO`.

DF	VSD	MSPD	MSSD	AR	T[ms]	#ref/obj
[4]	74.2	69.9	77.6	73.9	17	2
PIR	74.9	70.8	78.2	74.7	20	2
SIR	76.5	72.9	80.2	76.5	49	2
[4]	71.2	66.3	75.6	71.0	71	10
RIR	77.9	73.9	80.6	77.5	228	10
VF_b	78.3	73.8	80.6	77.6	495	10
VF_d	**78.5**	**74.1**	**80.9**	**77.8**	521	10

baselines using the same number of refinement iterations by 5.1% and 1.3% AR, respectively.

Regarding runtime, we observe the application of physics simulation results in a small relative increase per frame of 1ms per simulation. Note that, using DF-R as refinement method, SIR and RIR still achieve 71fps and 21fps. The significantly improved accuracy more than outweighs the increase in runtime in the final robotic application, as shown in the grasping experiments (Sec. V-D).

2) Multi-Object Scenario: An evaluation on the YCBV and xAPC datasets highlights the performance in multi-object scenarios. For comparison with RIR, VF_b and VF_d, the baseline DF-R is also run for the same number of iterations.

The results on YCBV are shown in Table III. The supervision through SIR is again the biggest source of relative improvement as compared to DF-R with an increase of 2.6% AR. The increased number of refinement iterations decreases the performance of DF-R. This could be due to the confidence score of DF suggesting a sub-optimal initial estimate for exploitation or due to divergence of the refinement method itself. In either case, RIR does not exhibit divergent behavior and is able to outperform the baseline method given the same number of refinement iterations by 6.5%. Table II shows that on xAPC, the performance of RIR improves over the approach by Mitash et al. [9] on the VSD and MSPD metrics by 3.3% and 0.4% and significantly speeds-up the runtime.

The results on both datasets show that our scene-level approaches successfully deal with the occlusion and support relationships in multi-object scenarios. Both VF_b and VF_d outperform [9] by a significant margin of 2.3% and 5.0% AR, respectively, with VF_d performing the best overall. All our approaches are approximately five times faster, with TrICP accounting for 5s per frame with VF_d. This highlights the benefit of the initialization of the solutions, the efficient search space formulation and our GPU-based computation of the verification score. As YCBV contains highly cluttered scenes that introduce occlusion but features only few support relationships, the relative increases over RIR are less pronounced with 0.1% and 0.3%. Overall, our scene-level approaches perform best on YCBV with VF_d achieving an increase of 6.8% over DF-R given the same number of iterations.

Fig. 6: Refined estimates using RIR (left), retrieved annotated grasps (mid) and successful grasp attempt (right).

TABLE IV: Results of grasping experiments in percentage of *found* collision-free grasp poses and *success*ful grasps.

DF	mustard	spam	foam	jello	banana	success	found	#ref/obj
[4]	10	3	1	7	0	42%	46%	2
SIR	9	7	2	7	0	50%	70%	2
[4]	10	6	5	9	1	62%	70%	10
[9]	9	10	2	6	0	54%	78%	10
RIR	**10**	**10**	**9**	**10**	**4**	**86%**	**90%**	10

D. Robotic Grasping Experiment

Our work is motivated by the performance deterioration of object detection and pose estimation methods when deployed on robots [7], [8]. To evaluate whether the proposed approach is able to reduce this problem, its performance is evaluated in a grasping experiment using a Toyota HSR and YCBV objects. Reproducible experimental conditions are ensured by using the GRASPA scene layouts [35] to place 5 objects as shown in Figure 1. 10 grasps are attempted per object and method – 5 are attempted for a given pose and an additional 5 for a rotation to a symmetric pose. Multiple grasp poses are annotated by hand for each object as shown in Figure 6 (mid).

In each experiment, Mask R-CNN [32] is executed to detect objects and to provide instance segmentation masks. The evaluated methods are queried to compute an object pose estimate from this information and the RGB-D image. To this end, for [9] and our approaches, we generate a hypotheses pool using DF [4]. Using the resulting pose estimate per object, the annotated grasp poses are transformed to the scene. Trajectories for all collision-free grasps are planned using MoveIt [36]. If at least one plan is found, this is counted as a *found* grasp. A grasp is considered a *success*ful grasp if the plan can be executed, i.e., the object is grasped and remains stable in the robot's gripper.

As shown in Table IV, our proposed approach generates object pose estimates that result in more successful and reliable grasps. The most striking improvements are achieved on the "061_*foam*_brick" and "011_*banana*" objects. Due to their proximity to other objects, object poses must be accurate to allow collision-free grasps. The *banana* is the most difficult object, resulting from inaccuracy in the instance segmentation and the low height. The experiment runtime is

dominated by grasp planning and execution. All compared settings have overall comparable execution time, therefore, the runtime difference for pose estimation, verification and refinement is insignificant for practical applications.

VI. Conclusion

This work presented an approach for the tight integration of hypotheses verification, refinement and physics simulation for object pose estimation. The rendering-based hypotheses verification and the proposed physics-guided extension to iterative refinement methods benefit from this integration by allowing them to share useful information. The comparison with state-of-the-art methods and a robotic grasping experiment show that our integrated approach creates more accurate and more reliable object pose estimates. Furthermore, we are able to increase performance over related work while significantly reducing the runtime.

An open issue is the presence of a-priori unknown objects. With interactions between known and unknown objects, the results of simulation will diverge from the true object pose. Incorporating shape estimation would enable unknown objects to be considered in simulation. Moreover, the use of physics simulation requires structures on which objects can rest and an estimate for the gravity vector. For robotic applications, static objects in the robot's environment map could be considered as supporting structures. An IMU could be used to determine the gravity direction to become robust to non-planar support.

References

[1] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. T. Birchfield, "Deep object pose estimation for semantic robotic grasping of household objects," in *Proc. Conf. Robot Learn.*, 2018, pp. 306–316.

[2] S. S. Srinivasa *et al.*, "HERB: A home exploring robotic butler," *Auton. Robots*, vol. 28, no. 1, p. 5–20, 2010.

[3] S. Chitta, E. G. Jones, M. Ciocarlie, and K. Hsiao, "Mobile manipulation in unstructured environments: Perception, planning, and execution," *IEEE Robot. Autom. Mag.*, vol. 19, no. 2, pp. 58–71, 2012.

[4] C. Wang *et al.*, "DenseFusion: 6D object pose estimation by iterative dense fusion," in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit.*, 2019, pp. 338–3347.

[5] S. Hinterstoisser *et al.*, "Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes," in *Proc. Asian Conf. Comput. Vision*, 2012, pp. 548–562.

[6] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes," in *Proc. Robot.: Sci. Syst.*, 2018.

[7] M. R. Loghmani, B. Caputo, and M. Vincze, "Recognizing objects in-the-wild: Where do we stand?" in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 2170–2177.

[8] P. Ammirato, P. Poirson, E. Park, J. Košecká, and A. C. Berg, "A dataset for developing and benchmarking active vision," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 1378–1385.

[9] C. Mitash, A. Boularias, and K. E. Bekris, "Improving 6D pose estimation of objects in clutter via physics-aware Monte Carlo tree search," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 3331–3338.

[10] A. Krull, E. Brachmann, S. Nowozin, F. Michel, J. Shotton, and C. Rother, "PoseAgent: Budget-constrained 6d object pose estimation via reinforcement learning," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 6702–6710.

[11] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, 1992.

[12] F. Furrer *et al.*, "Autonomous robotic stone stacking with online next best object target pose planning," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 2350–2356.

[13] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3D object recognition," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2010, pp. 998–1005.

[14] J. Vidal, C.-Y. Lin, and R. Martí, "6D pose estimation using an improved method based on point pair features," in *Proc. Int. Conf. Control, Autom. Robot.*, 2018, pp. 405–409.

[15] A. Aldoma, F. Tombari, L. Di Stefano, and M. Vincze, "A global hypothesis verification framework for 3d object recognition in clutter," *IEEE Tran. Pattern Anal. Mach. Intell.*, vol. 38, no. 7, pp. 1383–1396, 2016.

[16] D. Bauer, T. Patten, and M. Vincze, "Monte Carlo tree search on directed acyclic graphs for object pose verification," in *Proc. Int. Conf. Comput. Vision Syst.*, 2019, pp. 386–396.

[17] ——, "6D object pose verification via confidence-based Monte Carlo tree search and constrained physics simulation," in *OAGM & ARW Joint Workshop*, 2019, pp. 153–158.

[18] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox, "DeepIM: Deep iterative matching for 6D pose estimation," in *Proc. Eur. Conf. Comput. Vision*, 2018, pp. 683–698.

[19] F. Manhardt, W. Kehl, N. Navab, and F. Tombari, "Deep model-based 6D pose refinement in RGB," in *Proc. Eur. Conf. Comput. Vision*, 2018, pp. 800–815.

[20] S. Zakharov, I. Shugurov, and S. Ilic, "DPOD: Dense 6D pose object detector in RGB images," in *Proc. IEEE Int. Conf. Comput. Vision*, 2019, pp. 1941–1950.

[21] Z. Jia, A. C. Gallagher, A. Saxena, and T. Chen, "3D reasoning from blocks to stability," *IEEE Tran. Pattern Anal. Mach. Intell.*, vol. 37, no. 5, pp. 905–918, 2014.

[22] B. Zheng, Y. Zhao, J. Yu, K. Ikeuchi, and S.-C. Zhu, "Scene understanding by reasoning stability and safety," *Int. J. Comput. Vision*, vol. 112, no. 2, pp. 221–238, 2015.

[23] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Mach. Learn.*, vol. 47, no. 2-3, pp. 235–256, 2002.

[24] L. Kocsis and C. Szepesvári, "Discounted ucb," in *Proc. 2nd PASCAL Challenges Workshop*, 2006.

[25] A. Garivier and E. Moulines, "On upper-confidence bound policies for switching bandit problems," in *Proc. Int. Conf. Algorithmic Learn. Theory*, 2011, pp. 174–188.

[26] T. Hodaň, E. Brachmann, B. Drost, F. Michel, M. Sundermeyer, J. Matas, and C. Rother, "BOP: Benchmark for 6D object pose estimation," `https://bop.felk.cvut.cz/challenges/bop-challenge-2019/`, 2019, [Online; accessed 13-May-2020].

[27] E. Brachmann, F. Michel, A. Krull, M. Ying Yang, S. Gumhold *et al.*, "Uncertainty-driven 6D pose estimation of objects and scenes from a single RGB image," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 3364–3372.

[28] M. Rad and V. Lepetit, "BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth," in *Proc. IEEE Int. Conf. Comput. Vision*, 2017, pp. 3828–3836.

[29] B. Tekin, S. N. Sinha, and P. Fua, "Real-time seamless single shot 6d object pose prediction," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 292–301.

[30] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in manipulation research: Using the Yale-CMU-Berkeley object and model set," *IEEE Robot. Autom. Mag.*, vol. 22, no. 3, pp. 36–52, 2015.

[31] S. V. Alexandrov, T. Patten, and M. Vincze, "Leveraging symmetries to improve object detection and pose estimation from range data," in *Proc. Int. Conf. Comput. Vision Syst.*, 2019, pp. 397–407.

[32] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vision*, 2017, pp. 2980–2988.

[33] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," `http://pybullet.org`, 2016–2019.

[34] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 1–4.

[35] F. Bottarel, G. Vezzani, U. Pattacini, and L. Natale, "GRASPA 1.0: GRASPA is a robot arm grasping performance benchmark," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 836–843, 2020.

[36] I. A. Sucan and S. Chitta, "MoveIt," 2013.