Scene-level Pose Estimation for Multiple Instances of Densely Packed Objects

Chaitanya Mitash, Bowen Wen, Kostas Bekris, and Abdeslam Boularias
Department of Computer Science
Rutgers University
{cm1074, bw344, kb572, ab1544}@cs.rutgers.edu

Abstract: This paper introduces key machine learning operations that allow the realization of robust, joint 6D pose estimation of multiple instances of objects either densely packed or in unstructured piles from RGB-D data. The first objective is to learn semantic and instance-boundary detectors without manual labeling. An adversarial training framework in conjunction with physics-based simulation is used to achieve detectors that behave similarly in synthetic and real data. Given the stochastic output of such detectors, candidates for object poses are sampled. The second objective is to automatically learn a single score for each pose candidate that represents its quality in terms of explaining the entire scene via a gradient boosted tree. The proposed method uses features derived from surface and boundary alignment between the observed scene and the object model placed at hypothesized poses. Scene-level, multi-instance pose estimation is then achieved by an integer linear programming process that selects hypotheses that maximize the sum of the learned individual scores, while respecting constraints, such as avoiding collisions. To evaluate this method, a dataset of densely packed objects with challenging setups for state-of-the-art approaches is collected. Experiments on this dataset and a public one show that the method significantly outperforms alternatives in terms of 6D pose accuracy while trained only with synthetic datasets.

Keywords: Pose estimation, Synthetic data training, Computer Vision

1 Introduction

Robot manipulation pipelines, such as in bin-picking, often integrate perception with planning [1, 2, 3]. Some systems directly compute picks without computing the pose of objects by semantic segmentation or directly learning grasp affordances [4, 5, 6, 7]. While pose-agnostic techniques are promising, in many tasks it is important to first compute the 6D pose of observed objects to achieve purposeful manipulation and placement, such as in the context of packing [2, 8, 9].

Estimating 6D object poses has been approached in various ways [10], such as matching of locally-defined features [11], or of pre-defined templates of object models [12], or via voting in the local object frame using oriented point-pair features [13, 14]. Most methods were developed and evaluated for setups where each object appears once and for relatively sparsely placed objects on tabletops. Pose estimation for multiple instances of the same object type and where objects may be densely packed or in highly unstructured but dense piles has received less attention despite its significance in application domains, such as logistics. This is partly due to the increased difficulty of such setups.

There is prior work [10] that provides a public dataset [15] with a considerable number of instances for the same object type. Nevertheless, it measures the recall for estimating pose of any object instance in the scene. This may be sufficient in certain tasks but it is a weaker requirement than identifying the 6D pose of most, if not all, object instances. Achieving scene-level pose estimation allows a robot to internally simulate the world, reason about the order with which objects can be manipulated, as well as their physical interactions and the stability of their configuration. Scene-level reasoning can also infer missing information by considering occlusions and physical interactions between objects.

3rd Conference on Robot Learning (CoRL 2019), Osaka, Japan.

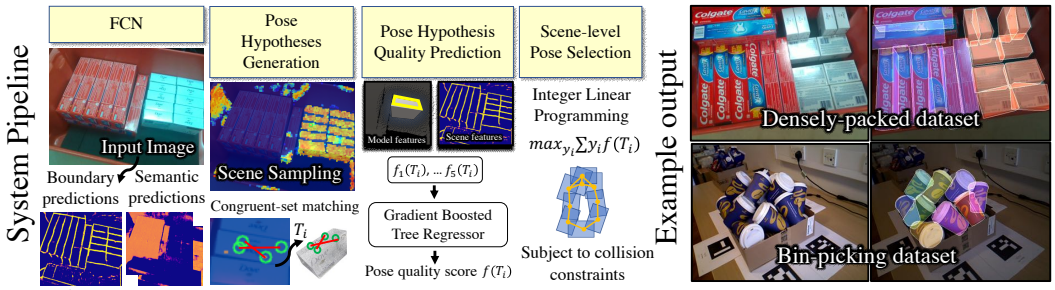

Figure 1: System pipeline and example output of the proposed approach on densely-packed scenes

The current work aims to improve the robustness of pose estimation for real-world applications, where object types appear multiple times, and in challenging, dense configurations, such as those illustrated in Figure 1, while allowing for scene-level reasoning. It aims to do so by proposing effective machine learning operations that depend less on manual data labeling and less on handcrafted combinations of multiple accuracy criteria into a single objective function. This allows the true automation of robot manipulation pipelines and brings the hope of wider-scale, real-world deployment.

Problem Setup: The considered framework receives as input: a) an RGB image x and a depth map of the scene; b) a set of mesh models $\{M_i\}_{i=1}^{K}$, one for each object type $i \in [1, K]$ present in the scene, and c) a set $\{N_i\}_{i=1}^{K}$ expressing an upper-bound on the number of instances for each object type i. The output is object poses as a set of rigid-body transformations $\{\{T_{ij}^*\}_{j=1}^{N_i}\}_{i=1}^{K}$, where each $T_{ij}^* = (t_{ij}^*, R_{ij}^*)$ captures the translation $t_{ij}^* \in R^3$ and rotation $R_{ij}^* \in SO(3)$ of the j^{th} instance of object type i in the camera's reference frame, for each of the K object types present in the scene.

Figure 1 summarizes the considered pipeline: a) CNNs are used to detect semantic object classes and visible boundaries of individual instances, b) then, a large set of candidate 6D pose hypotheses are generated for each object class, c) quality scores are computed for each hypothesis, and d) scene-level reasoning identifies consistent poses that maximize the sum of individual scores. In the context of this pipeline, the contribution of this work relative to state-of-the-art methods is two-fold:

A. Adversarial training with synthetic data for robust object class and boundary prediction: Machine learning approaches have become popular in pose estimation, both in end-to-end learning [16, 17] and as a pipeline component [18, 19]. They require, however, large amounts of labeled data. Recent approaches aim to solve single-instance pose estimation by training entirely in simulation [20, 19, 21]. The proposed method also utilizes labeled data generated exclusively in simulation to train a CNN for semantic segmentation. Nevertheless, CNNs are sensitive to the domain gap between synthetic and real data. The proposed training aims to mimic the physics of real-world test scenes and bridges the domain gap by using a generative adversarial network, as explained in Section 2. A key insight to improve robustness in the multi-instance case is that the network is simultaneously trained to predict object visibility boundaries. A thesis of this work is that boundaries learned on RGB images are more effective than boundaries detected on depth-maps [14] to guide and constrain the search for 6D poses, especially in tightly-packed setups.

B. Scene-level reasoning by automatically learning to evaluate pose candidate quality: This work finds the best physically-consistent set of poses among multiple candidates by formulating a constraint optimization problem and applying an ILP solver as shown in Section 3. The objective is to select pose hypotheses that maximize the sum of their individual scores, while respecting constraints, such as avoiding perceived collisions. Scene-level optimization has been previously approached as maximizing the weighted sum of various geometric features [22]. The weights characterizing the objective function, however, were carefully handcrafted. This work shows it is possible to learn the distance of a given candidate pose from a ground-truth one by using a set of various objective functions as features. A gradient boosted tree [23] is trained to automatically integrate these objectives and regress the distance to the closest ground truth pose. The objectives indicate how well a candidate hypothesis explain the predicted object segments, the predicted boundaries, the observed depth and local surface normals in the input data. Prior related work has used tree search [24, 25, 26] to reconstruct the scene by sequentially placing objects. These prior approaches, however, were restricted to a small number of objects or fewer degrees-of-freedom due to computational overhead. The proposed ILP solution is quite fast in practice and scales to a large number of objects. For the images of Figure 1, the scene-level optimization is achieved in a few milliseconds.

2 Semantic and Boundary Predictions

Fully Convolutional Networks (FCNs) [27, 28] are popular semantic segmentation tools. They have also been used for object contour detection [29] and predicting multiple instances of an object type [30, 31]. These networks are increasingly being trained in simulation [32, 33, 34, 20] to alleviate the need for large amounts of labeled data. The domain gap between the data generated in simulation and real data can lead to noisy predictions and greatly affect pose estimation accuracy. Several recent methods have been developed to bridge this domain gap [35, 36]. The current work subscribes to these ideas and: a) exploits the constraints available in robotic setups to simulate scenes with realistic poses, while b) uses adversarial training with unlabeled real images to bridge the gap between the labels predicted in synthetic data with those predicted in real ones.

This work proposes the use of a CNN to predict per-pixel semantic classification and a classification of whether a pixel is a visible object boundary. The data for training the CNN are generated in simulation with a physics engine and a renderer. The simulation samples a bin pose and a camera pose given the robot's workspace. Each scene is created by randomly sampling, within a pre-specified domain, the number and 6D poses of objects, the color of the bin, and the placement and intensity of the illumination sources. Finally, the scene is rendered to obtain a color image, a depth map, per-pixel class labels and visible instance boundary labels. The simulation generates a wide range of training data for domain randomization and robustness to domain gap issues. Nevertheless, domain gap still exists between synthetic data and data acquired through real sensors as it is hard to capture the domain of object material's interactions with various illumination sources in the environment.

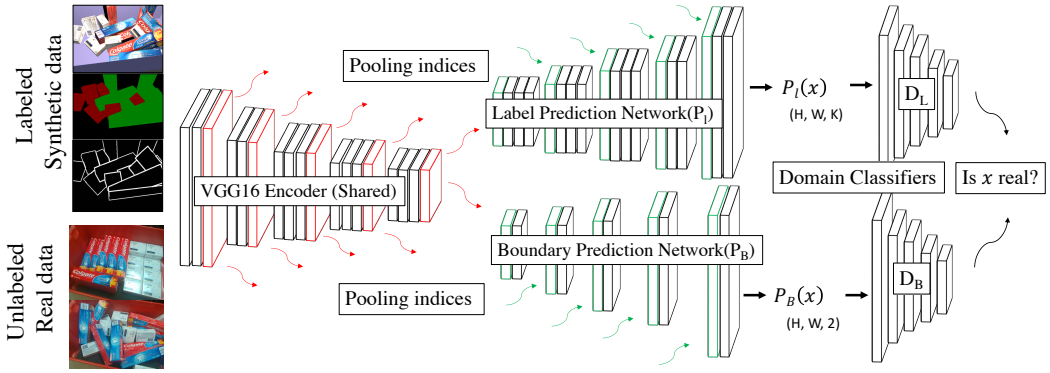

Figure 2: CNN architecture for semantic classes and boundary prediction.

The generative adversarial network (GAN), shown in Figure 2, performs the semantic and boundary detection tasks, while also adapting the output predictions on unlabeled real images to resemble the predictions on synthetic images. It consists of a shared VGG16 encoder that stacks five blocks of convolution, batch normalization and max pooling layers. The network branches out into two decoders for semantic and boundary classifications. These are fully convolutional decoders with unpooling indices passed from corresponding max pooling blocks in the encoder section. The outputs of both decoders are passed to a corresponding fully convolutional discriminator network.

The network is trained by taking as input a synthetic image x_s and its ground-truth label $Y(x_s)$. It also receives as input unlabeled real image x_r. The output $P_l(x_s)$ of the label prediction network on image x_s (or corresponding output on x_r) is then passed on to the label discriminator D_L whose task is to classify correctly if the prediction is on real or on synthetic data. Along with the objective of correctly labeling the synthetic image x_s, the label prediction network should also confuse D_L into classifying $P_l(x_r)$ as an output of a sample coming from the synthetic domain. The objective of the semantic labeling network is defined as:

$$\mathcal{L}_{sem} = -\sum_{h,w}\sum_{k\in K} Y(x_s)_{(h,w,k)} \log P_l(x_s)_{(h,w,k)} - \lambda_a \sum_{h,w} \log D_L(P_l(x_r))_{(h,w,0)}$$

where $P_l(x_s)$ is the per-pixel K-channel (for K object classes) output from the labeling network, (h, w) are pixel coordinates, and λ_a is a weight factor. $D_L(P_l(x))_{(h,w,0)}$ is the predicted score of x in pixel (h, w) of being a synthetic image. The domain classifier's objective is specified as:

$$\mathcal{L}_{D_L} = -\sum_{h,w} \log D_L(P(x_s))_{(h,w,0)} - \sum_{h,w} \log D_L(P(x_r))_{(h,w,1)}$$

A similar GAN objective is used to simultaneously train the boundary predictor and discriminator.

3 Scene-level Pose Selection

This section formulates the scene-level objective and presents how to address it in a computationally efficient manner, despite its computational hardness. Given the semantic and boundary predictions, a set of 6D pose hypotheses is generated, $\{\{T_{ij}\}_{j=1}^{H_i}\}_{i=1}^{K}$, where H_i denotes the number of hypotheses for each object class ($H_i \gg N_i$). Pose candidates can be generated via learning [37], `RANSAC` [18] hough voting [13], or by incorporating distinctive geometric features [19].

The current work uses a stochastic representation of output from segmentation for hypotheses generation via congruent set matching [38, 19]. It iteratively samples a set of points (called a *base*) from the observed point cloud such that the points in each set belong with high probability to a single object instance. The sampled point sets are then matched to congruent sets on corresponding object model to generate candidate rigid transformations. A key feature of our hypotheses generation process, compared to [38, 19], is that the boundary predictions from the previous step are utilized to limit the selection of points within a single object instance. An adaptive sampling process is used to cover all instances of the same object by enforcing dispersion. The detailed formulation for the hypotheses generation process can be found in Appendix A.

The final objective is to select for an object category i, a subset of poses $\{T_{ij}^*\}_{j=1}^{N_i} \subset \{T_{ij}\}_{j=1}^{H_i}$, which maximizes the total sum of scores $f(T_{ij}^*)$ while avoiding collisions between objects when assigned to those poses. In other terms, the intersection between the volume occupied by any two object instances in the scene should be empty. The approach first identifies a set that contains all pairs of poses, which conflict with each other. This set is defined as $\mathcal{C} = \{\{(i,j),(i',j')\}$ s.t. $|\mathcal{V}(M_i,T_{ij}) \cap \mathcal{V}(M_i,T_{i'j'})| > \epsilon_v\}$ where $i,i' \in \{1,..,K\}, j \in \{1,..,H_i\}, j' \in \{1,..,H_i'\}$ and $\mathcal{V}(M,T)$ is the volume occupied by a model M when placed according to pose T. ϵ_v is the maximum volume of tolerated collisions between objects in the scene. This positive error term is necessary because the best poses among the sampled ones may induce slight collisions between objects, which can often be corrected afterwards.

Pose selection is formulated as a constraint optimization problem, which can be solved by `ILP`. A set of binary variables $\{\{y_{ij}\}_{j=1}^{H_i}\}_{i=1}^{K}$ is defined where $y_{ij} \in \{0,1\}$. Each variable y_{ij} can be seen as an indicator variable on whether pose hypothesis T_{ij} is included in the set of selected posed $\{T_{ij}^*\}_{j=1}^{N_i}$. Then, the optimization problem is as follows:

$$\begin{aligned} \max_{\{y_{ij}\}} \quad & \sum_{i=1}^{K}\sum_{j=1}^{H_i} y_{ij} f(T_{ij}) \\ \text{subject to:} \quad & \forall i \in \{1,\dots,K\} : \sum_{j=1}^{H_i} y_{ij} \leq N_i \\ & \forall \{(i,j),(i',j')\} \in \mathcal{C} : y_{ij} + y_{i'j'} \leq 1. \end{aligned}$$

The first constraint ensures that the number of poses selected for each category of object i do not exceed the number N_i of instances. The second constraint ensures that poses that are conflicting cannot be selected. This problem is equivalent to the Maximum-Weight Independent Set Problem (`MWISP`), with an additional constraint related to the number of selected poses. `MWISP` is NP-hard, and there are no $\frac{1}{n^{1-\epsilon}}$-approximations for any fixed $\epsilon > 0$ where n is the number of variables $\{y_{ij}\}$ [39]. In practice, however, and for the problems considered here, an exact solution can be found very fast (in milliseconds) using modern `ILP` solvers for scenes containing a few hundreds of candidate poses. This is because the poses tend to cluster into cliques around specific instances, with a small number of constraints between poses in different cliques. Moreover, the objective function is monotone sub-modular as it is a linear function of a subset [40]. Therefore, greedy optimization is guaranteed to find in linear time a solution that is at least $(1 - \frac{1}{e})$ fraction of the optimal. An approximate solution can also be found in linear time with an `LP` relaxation. After solving the `ILP`, the poses $\{T_{ij}^*\}_{j=1}^{N_i}$ are constructed from $\{T_{ij}\}_{j=1}^{H_i}$ by keeping those for which $y_{ij} = 1$.

4 Pose Hypothesis Quality Evaluation

Scene-level optimization, presented in Section 3, requires that a single quality score $f(T)$ is assigned to each pose T in the hypotheses set. The proposed score function considers five indicators $f_{1:5}(T)$ of a good alignment for each pose candidate T, shown in Table 1. A straightforward solution is to define f as a weighted sum of its components. Nevertheless, the resulting poses would heavily depend on the choice of the weights. Choosing the right weights manually for every new object type

Notation	
$B(T)$: Visible boundary pixels when the object model is placed at pose T and rendered S_B: Scene boundary pixels predicted by the CNN. $V(T)$: Visible portion of the the object model when placed at pose T and rendered. δ_S, δ_B: Surface and boundary matching distance thresholds. $\mathcal{D}_S(p, T, O_D)$: Depth distance between rendered image and observed depth image O_D at pixel p. $\mathcal{D}_B(p, T, S_B)$: Distance between rendered boundary and predicted boundary at pixel p.	
Model-to-Scene consistency features	
f_1	$\lvert B(T) \cap S_B \rvert / \lvert B(T) \rvert$: fraction of pixels in model boundary that match the scene boundary.
f_2	$\lvert B(T) \cap S_B \rvert / \lvert V(T) \cap S_B \rvert$: fraction of pixels in scene boundary within the visible model region that match the model boundary.
f_3	$\sum_{p \in V(T)} \mathbf{1}_{\mathcal{D}_S(p,T,O_D) < \delta_S} / \lvert V(T) \rvert$: fraction of pixels in visible model region that is sufficiently aligned in terms of depth to the observed data.
Scene alignment features	
f_4	$\sum_{p \in V(T)} P_l(p) \mathcal{S}(p, T, O_D)$: surface alignment score weighted by the corresponding label probability. Similarity score given by $\mathcal{S}(p, T, O_D) = (1 - \frac{1}{\delta_S} \mathcal{D}_S(p, T, O_D)) \mathcal{N}(p, T)$ and it considers depth distance and surface normal similarity.
f_5	$\sum_{p \in B(T)} (1 - \frac{1}{\delta_B} \mathcal{D}_B(p, T, S_B))$: boundary alignment score based on distance between point on model boundary and it's nearest point in the predicted boundary set.

Table 1: Description of features indicating good pose alignment with sensory input

is not trivial. Instead, a key aspect of this work is to learn the objective function $f(T)$ using $f_{1:5}(T)$ as features, i.e., $f(T) = h(f_{1:5}(T))$. The function h is learned by minimizing the following loss:

$$\min_{h} \sum_{(T, f_{1:5}(T), T^g) \in \mathcal{T}_{train}} \Big(h(f_{1:5}(T)) - e_{ADI}(T, T^g) \Big)^2, \tag{1}$$

where $e_{ADI}(T, T^g)$ is the distance between a given pose T and a ground-truth pose T^g. The distance $e_{ADI}(T, T^g)$ is computed using the ADI metric, which is frequently used in the literature for evaluating pose estimations [12]. This metric is explained in the experimental section.

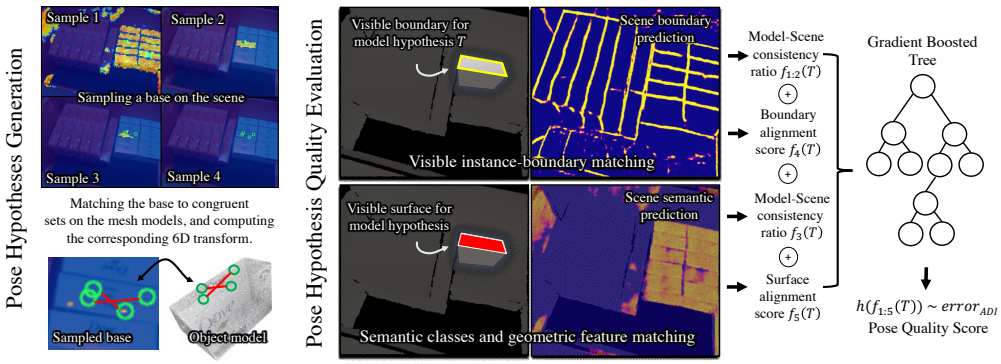

Figure 3: Regressing the hypothesis quality given various alignment features.

The training data set $\mathcal{T}_{train}$ is collected by simulating different scenes in a physics engine in the same way as described in Section. 2. For each scene, the CNN predicts semantic and object boundaries and a large number of pose hypotheses are sampled. For each hypothesis T, their alignment features $f_{1:5}(T)$ and corresponding scores $e_{ADI}(T, T^g)$ are computed based on the closest ground-truth pose. The regression learning problem is to find a function h that maps features $f_{1:5}(T)$ of a pose T to its actual distance from the corresponding ground-truth pose T^g.

This work adopts the Gradient Boosted Regression Trees (GBRT) to solve the optimization problem in Equation 1. GBRTs are well-suited for handling variables that have heterogeneous features [23]. GBRTs are also a flexible non-parametric approach that can adapt to non-smooth changes in the regressed function using limited data, which often occurs when dealing with objects that have different shapes and sizes. An implementation of GBRT is available in the Scikit-learn library [41].

5 Experiments

This section describes the experimental study performed on 2 datasets of scenes with multiple instances of objects. In this study, the recall for pose estimation is measured based on the error given by `ADI` [12], which measures distances between poses T_1 and T_2 given an object mesh model M:

$$e_{ADI}(T_1, T_2) = \text{avg}_{b_1 \in M} \text{min}_{b_2 \in M} ||T_1(b_1, M) - T_2(b_2, M)||_2,$$

where $T(b, M)$ corresponds to point b after applying transformation T on M. Given a ground-truth pose T^g, a true positive is a returned pose T that has $e_{ADI}(T, T^g) < k_l d_l$, where k_l is a fraction set to $\frac{1}{10}$, while d_l is the diameter of the object model calculated as the maximum distance between any two points on the model.

The two datasets include a public dataset called *bin-picking dataset* [15] and a new one developed as part of this submission. The *bin-picking dataset* contains two object types and 3 scenarios (Figure 2). These scenarios present clutter of objects with high occlusion rate. For this dataset, segmenting objects in color images is challenging but depth cues can be used to find the object boundaries. This leads to depth-based approaches achieving high estimation success on this dataset without color information. The new dataset, henceforth called the *densely-packed* dataset, comprises of 30 unique scenes with two object categories. Each scene contains 15 to 19 different instances of these objects, which were manually labeled with 6D pose annotations. There are two types of scenes in this dataset, as illustrated below. In one type, denoted as scenario 1, the instances are tightly packed next to each other. This case is particularly challenging because the surfaces of multiple instances are aligned, which makes it difficult to use depth information for segmentation. The dataset and the code is shared alongside the paper.

Table 2: Object type and scenarios in the (left) *densely-packed* and (right) *bin-picking* dataset.

Approach	o1	o2	Mean
OURS	**79.9**	**85.2**	**82.1**
Hinterstoisser at. al [12]	37.0	65.6	49.3
PPF-Voting [13]	30.1	57.6	41.9
Buch at. al [42]	11.2	31.7	19.9
LCHF [15]	16.2	44.3	28.3
MRCNN-StoCS [43, 19]	42.8	68.3	53.7
PoseCNN** [16]	15.0	46.9	28.7
PoseCNN + ICP** [16]	56.8	80.6	67.0
DOPE** [20]	51.0	70.6	60.8
DOPE [20]	3.5	10.5	6.5

Table 3: Pose retrieval recall rate on *densely-packed* dataset. ** [16, 20] tested on synthetic version.

Approach	o1	o2
OURS	**64.1**	**55.7**
Buch at. al [42]	63.8	44.9
PPF-Voting [13]	47.4	27.9
LCHF [15]	33.5	25.1
Tejani et. al. [44]	31.4	24.8

Table 4: Recall on *bin-picking dataset*. Results for other approaches are obtained from [42]. Several object instances were completely hidden. This leads to lower recall rates even when the algorithms could retrieve all the visible instances.

5.1 Evaluation against recent pose estimation techniques

Several state-of-the-art pose estimation techniques are evaluated on the above datasets (Table 3 and Table 4). A popular template-matching work [12] matches templates extracted from `RGB` and depth rendering of `CAD` models. It fails on several occasions as the templates are not robust to occlusion and varying lighting conditions. Approaches based on hough voting with point-pair features [13, 42] achieve high success on the *bin-picking* dataset but fail to do so on the *densely-packed* dataset. These approaches detect multiple object instances by considering the peaks in hough voting space, several of which are false positives by virtue of aligned surfaces in the packed boxes scenario. `LCHF` [15] is tailored to handle multiple instances of the same object category. Even after carefully tuning the weights of the optimization function and relaxing the criteria for the number of pose candidates to select, the recall from this approach is rather low. This can be majorly attributed to differences in pre-defined template descriptors between the scene and the object model used for matching local patches. `LCHF` [15] also includes an active vision component not considered in this evaluation.

Next, recent deep-learning based techniques for pose estimation are evaluated. StoCS [19] matches point cloud object models to stochastic output from a CNN trained for semantic segmentation with synthetic data. One way to apply StoCS for multi-instance estimation is to integrate it with an instance segmentation technique, such as Mask R-CNN [43]. Mask R-CNN was trained with synthetic data and used to extract top 10 instances of each object type according to the detection probability. Pose estimation was performed using StoCS for each of these individual segments. There are two major limitations of this combination. The first is the use of deterministic instance boundaries leading to segmentation noise that cannot be recovered during pose estimation. The second is that visibility is not considered when matching the model to the detected segment, which can lead to several incorrect poses achieving high alignment scores.

PoseCNN [16] is an end-to-end learning approach. It includes a network branch for semantic segmentation. Pixels belonging to an object class then vote for the object centroid's location. Based on the peaks in voting, the center is localized and corresponding inliers are used to find a region of interest (RoI). Features of the RoI regress in a separate branch of the network to output the object's rotation. PoseCNN was originally developed for single instances but via non-maximal suppression over the output of hough voting, it can be adapted for multiple instances. To eliminate domain gap from the scope of testing, PoseCNN was tested on a synthetic dataset with the same scenarios as the ones in real testing. PoseCNN outputs an object pose that could be used as an initialization for a depth-based ICP-like process that utilizes perturbations and local search for refinement. Overall, object symmetry and tight-packing scenarios make the simultaneous training of the multiple network branches hard to converge and the final ICP process less effective in these scenarios.

DOPE [20] is another learning-based approach that recovers 6D pose via perspective-n-point (PnP) from predictions of 3D bounding-box vertices projected on the image. It aims to bridge the simulation-to-reality gap by a combination of domain randomization and photo-realistic rendering. The open-sourced rendering engine NDDS [45] used in DOPE, however, does not provide access to photo-realistic rendering. Thus, the comparisons were made against the DOPE-DR version. The neural network output is composed of a belief map used to find the projected vertices by a local peak search as well as an *affinity map*, which indicates the direction from projected vertices to their corresponding center for assignment. Nevertheless, when multiple instances of the same object are placed next to each other, some 2D vertices significantly overlap around the border of two neighboring instances. This makes the assignment of vertices to the correct center problematic and degrade the performance of (PnP), since it requires relatively precise 2D-3D point correspondences. DOPE uses only color information without depth, which is a disadvantage when compared to other methods in Table 3. DOPE was trained from scratch with synthetic data generated by following the same pipeline as presented in [20]. Using the best tuned parameters for domain randomization and post-processing steps, the pose estimation recall was measured on a synthetic validation set and on the real test set.

5.2 Ablation study of the proposed technique

The CNN in Section 2 is trained with 20,000 scenes generated in simulation by randomly dropping objects in the bin. It is then fine-tuned with images rendered from 50 simulated scenes of tightly-packed objects. The networks were trained using the Adam optimizer with an initial learning rate of 2×10^{-3}. The weight λ_a for the GAN loss is set to 10^{-3}. To handle the class imbalance in the boundary network, the ratio of boundary to non-boundary pixels was computed in every iteration of the training and used to weight the respective loss terms.

Table 5 indicates that label-space adaptation is the most effective training strategy in this case and even more so when scenes that mimic the packing scenario were used. Training solely on synthetic data with no adaptation significantly reduces the performance. An unpaired image translation approach, Cycle-GAN [35], achieves comparable performance. But with no semantic constraints, it biases the transfer for dominant classes in the dataset (o2 in this case), and also cannot deal with background clutter. Figure 4 shows synthetic training images for different datasets and boundary predictions on real images of corresponding datasets. To evaluate the generalization capacity of the training process, the network was also trained for the Occluded-Linemod dataset [18] that contains 8 object classes and unseen background clutter. Even then, the training was able to predict boundaries of only concerned objects.

Given the predictions from CNN, the hypotheses generation process described in Appendix A finds for each object type, a set of pose candidates, which should be large enough to include the true

Training	o1	o2	All
Adapt-finetune (ours)	**79.9**	85.2	**82.1**
Adapt-nofinetune	74.5	84.6	78.8
Cyclegan [35]	75.6	**88.2**	81.0
No adaptation (rgb)	69.8	79.5	73.9

Table 5: Evaluating different training strategies.

ILP Objective function	o1	o2	All
Optimal	87.6	90.8	88.9
Learned	79.9	85.2	82.1
Manual (scene + model)	76.4	83.1	79.2
Manual (scene)	59.8	82.6	69.6

Table 6: learned vs manually-defined objective.

Method	Selected	All-Candidates
with boundary	82.1	92.9
w/o boundary	58.7	78.0

Table 7: Effect of using boundaries for hypotheses generation.

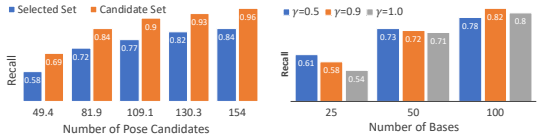

Table 8: Recall as a function of number of pose candidates and the dispersion parameter (γ)

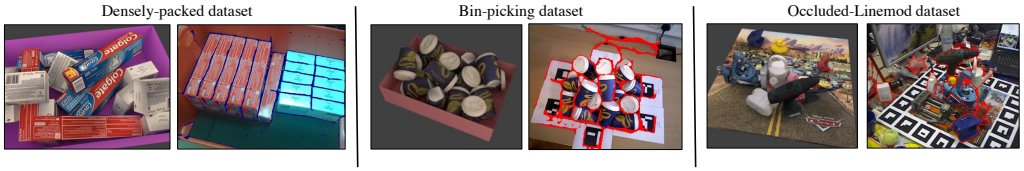

Figure 4: Examples of synthetic training data and boundary predictions on real images.

poses and small enough to reduce the computation time. The number of candidates is affected by the number of sampled bases and pose clustering. Recall rates are shown in Figure 8 separately for all generated candidates and the ones selected by ILP. It also shows the effect of dispersion parameter γ, which is by default set to 0.9. Table 7 evaluates the contribution of the boundary reasoning in the hypothesis generation and shows that it has a significant impact on the overall recall.

Table 6 compares learned objective function for ILP vs manually-defined ones. One way to combine features defined in Section 3 is to consider only scene alignment scores ($f_3 + f_4$) as in many point registration algorithms or alternatively use both scene alignment and model consistency scores ($f_1 * f_2 * f_3(f_4 + f_5)$). An upper bound of performance with the given hypotheses set is established as the optimal recall when the true ADI distance from ground-truth is used in the optimization.

The overall computation time for the current sequential implementation of the approach ranges from 10s to 15s for estimating all (15 to 19) instances in a scene. The CNN predictions, the pose hypotheses generation and the scene-level optimization along with collision checking run individually in less than a second. Broad phase collision checking is performed to speed up the process. The majority of computation time is spent on depth-buffering and local refinement for each pose candidate (130 per object category). These operations can be significantly sped up with parallel processing such as CUDA–OpenGL interoperability, which has been shown [25] to render 1000 images in 0.1s. Given the data parallelism, multiple cores can also be easily utilized to speed up the algorithm.

6 Discussion

This work focuses on hard instances of scene-level, multi-instance pose estimation, which includes highly cluttered and densely packed scenarios. The results show that the type of poses used in simulation for training the semantic and the boundary networks is important. While a simulation-to-reality domain gap exists, it can be bridged by using appropriate information, such as boundary prediction, which translates well from simulated to real images, and adversarial training strategies. Furthermore, the ILP formulation of the scene-level reasoning is able to find combinations of hypotheses that are both consistent as well as of high-quality given the learned global score function. The consistency of pose hypotheses in the current work is defined by collision constraints. It is interesting to consider a learning process for identifying compatible sets of poses that express physical constraints and which could be used in the context of the proposed ILP formulation.

Acknowledgments

This work was supported by the NSF, grant numbers IIS-1734492 and IIS-1723869.

Bibliography

[1] N. Correll, K. Bekris, D. Berenson, O. Brock, A. Causo, K. Hauser, K. Osada, A. Rodriguez, J. Romano, and P. Wurman. Analysis and Observations From the First Amazon Picking Challenge. *T-ASE*, 2016.

[2] M. Schwarz, C. Lenz, G. García, S. Koo, A. Periyasamy, M. Schreiber, and S. Behnke. Fast object learning and dual-arm coordination for cluttered stowing, picking, and packing. In *ICRA*, 2018.

[3] A. Zeng, K. Yu, S. Song, D. Suo, E. Walker, A. Rodriguez, and J. Xiao. Multiview self-supervised deep learning for 6d pose estimation in the amazon picking challenge. In *ICRA'17*.

[4] M. Gualtieri, A. Ten Pas, K. Saenko, and R. Platt. Grasp Pose Detection in Point Clouds. *IJRR*, 2017.

[5] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. Ojea, and K. Goldberg. Dex-net 2.0. In *R:SS*, 2017.

[6] D. Morrison et al. Cartman: The low-cost cartesian manipulator that won the amazon robotics challenge. In *ICRA*, 2018.

[7] A. Zeng et al. Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching. In *ICRA*, 2018.

[8] R. Shome, W. N. Tang, C. Song, C. Mitash, C. Kourtev, J. Yu, A. Boularias, and K. Bekris. Towards robust product packing with a minimalistic end-effector. In *ICRA*, 2019.

[9] W. Fan and K. Hauser. Robot Packing with Known Items and Nondeterministic Arrival Order. In *R:SS*, 2019.

[10] T. Hodan, F. Michel, E. Brachmann, W. Kehl, A. GlentBuch, D. Kraft, B. Drost, J. Vidal, S. Ihrke, X. Zabulis, et al. Bop: benchmark for 6d object pose estimation. In *ECCV*, 2018.

[11] A. Aldoma et al. Tutorial: Point cloud library: Three-dimensional object recognition and 6 dof pose estimation. *IEEE Robotics & Automation Magazine*, 2012.

[12] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab. Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes. In *ACCV*, 2012.

[13] B. Drost, M. Ulrich, N. Navab, and S. Ilic. Model Globally, Match Locally: Efficient and Robust 3D Object Recognition. In *CVPR*, 2010.

[14] J. Vidal, C.-Y. Lin, X. Lladó, and R. Martí. A Method for 6D Pose Estimation of Free-Form Rigid Objects Using Point Pair Features on Range Data. *Sensors*, 2018.

[15] A. Doumanoglou, R. Kouskouridas, S. Malassiotis, and T.-K. Kim. Recovering 6d object pose and predicting next-best-view in the crowd. In *CVPR*, 2016.

[16] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox. PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes. In *R:SS*, 2018.

[17] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab. Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In *ICCV*, 2017.

[18] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother. Learning 6d object pose estimation using 3d object coordinates. In *ECCV*, 2014.

[19] C. Mitash, A. Boularias, and K. Bekris. Robust 6d object pose estimation with stochastic congruent sets. In *BMVC*, 2018.

[20] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield. Deep Object Pose Estimation for Semantic Robotic Grasping of Household Objects. In *CoRL*, 2018.

[21] M. Sundermeyer, Z.-C. Marton, M. Durner, M. Brucker, and R. Triebel. Implicit 3D Orientation Learning for 6D Object Detection from RGB Images. In *ECCV*, 2018.

[22] A. Aldoma, F. Tombari, L. Di Stefano, and M. Vincze. A global hypotheses verification method for 3d object recognition. In *ECCV*, 2012.

[23] J. Elith, J. R. Leathwick, and T. Hastie. A working guide to boosted regression trees. *JAE'08*.

[24] V. Narayanan and M. Likhachev. Discriminatively-guided Deliberative Perceptinon for Pose Estimation of Multiple 3D Object Instances. In *R:SS*, 2016.

[25] Z. Sui, L. Xiang, O. C. Jenkins, and K. Desingh. Goal-directed robot manipulation through axiomatic scene estimation. In *IJRR*, 2017.

[26] C. Mitash, A. Boularias, and K. E. Bekris. Improving 6d pose estimation of objects in clutter via physics-aware monte carlo tree search. In *ICRA*, 2018.

[27] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[28] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *ICCV*, 2015.

[29] J. Yang, B. Price, S. Cohen, H. Lee, and M.-H. Yang. Object contour detection with a fully convolutional encoder-decoder network. In *CVPR*, 2016.

[30] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei. Fully convolutional instance-aware semantic segmentation. In *CVPR*, 2017.

[31] A. Kirillov, E. Levinkov, B. Andres, B. Savchynskyy, and C. Rother. Instancecut: from edges to instances with multicut. In *CVPR*, 2017.

[32] C. Mitash, K. Bekris, and A. Boularias. A self-supervised learning system for object detection using physics simulation and multi-view pose estimation. In *IROS*, 2017.

[33] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IROS*, 2017.

[34] S. Hinterstoisser, V. Lepetit, P. Wohlhart, and K. Konolige. On pre-trained image features and synthetic images for deep learning. In *ECCV*, 2018.

[35] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networkss. In *ICCV*, 2017.

[36] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker. Learning to adapt structured output space for semantic segmentation. In *CVPR*, 2018.

[37] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese. Densefusion: 6d object pose estimation by iterative dense fusion. In *CVPR*, 2019.

[38] N. Mellado, D. Aiger, and N. Mitra. Super4PCS: Fast Global Pointcloud Registration via Smart Indexing. In *Computer Graphics Forum*, 2014.

[39] J. Hastad. Clique is hard to approximate within $n^{1-\epsilon}$. In *Acta Mathematica*, 1996.

[40] A. Krause and D. Golovin. *Submodular Function Maximization.* Cambridge Press, 2014.

[41] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *JMLR*, 2011.

[42] A. G. Buch, L. Kiforenko, and D. Kraft. Rotational subgroup voting and pose clustering for robust 3d object recognition. In *ICCV*, 2017.

[43] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *ICCV*, 2017.

[44] A. Tejani, D. Tang, R. Kouskouridas, and T. K. Kim. Latent-class Hough Forests for 3D Object Detection and Pose Estimation. In *ECCV*, 2014.

[45] T. To, J. Tremblay, D. McKay, Y. Yamaguchi, K. Leung, A. Balanon, J. Cheng, W. Hodge, and S. Birchfield. NDDS: NVIDIA deep learning dataset synthesizer, 2018.

Appendix A: Pose Hypotheses Generation

Given the output of semantic class predictions P_l and boundary prediction P_B, this step aims to generate a set of 6D pose hypotheses $\{\{T_{ij}\}_{j=1}^{H_i}\}_{i=1}^{K}$, where H_i denotes the number of pose hypotheses for each object model M_i and is larger than the number of the object's instances in the scene, i.e, $H_i \gg N_i$. Similar to the overall objective of the problem setup, each $T_{ij} = (t_{ij}, R_{ij})$ is a candidate translation $t_{ij} \in R^3$ and rotation $R_{ij} \in SO(3)$ of some instance of object type M_i in the camera's reference frame, for each of the K object types present in the scene.

The sets of pose hypotheses should be sufficiently large to ensure they contain candidates close enough to the unknown true poses, i.e, $\forall i \in [1, K], \forall j \in [1, N_i], \exists\, T \in \{T_{ij'}\}_{j'=1}^{H_i} : \|T - T_{ij}^g\| \leq \epsilon$, where T is a hypothesis pose for object M_i and T_{ij}^g is the ground truth pose of the j^{th} instance of object M_i. Since the ground truth poses are unknown during testing, one cannot guarantee that the previous desired property will be satisfied unless the candidates are densely sampled from $SE(3)$, which would make the search computationally expensive. Therefore, it is important to sample the candidates in a manner that balances their diversity and their similarity to the ground truth given the semantic class predictions P_l and models $\{M_i\}_{i=1}^{K}$.

Pose hypotheses generation is often performed using RANSAC-like techniques [1] or via hough voting [2], or various distinctive geometric features [3, 4, 5], which ensure that multiple points can be sampled from the same object. These methods do not incorporate boundary information, which is important in cases where the surfaces of multiple object instances are aligned. The following explains the proposed pose hypotheses generation process, which utilizes both semantic segmentation and instance-boundaries.

The output of the softmax layer from P_l is used to compute $\pi_l(p_i)$, the probability of each pixel p_i belonging to an object class l. This probability is defined as the ratio of the output $P_l(p_i)$ over the sum of the outputs for the same class over all pixels p in the given test image x, i.e./ $\pi_l(p_i) = \frac{P_l(p_i)}{\sum_p P_l(p)}$.

Based on the principles of randomized alignment, candidate poses are generated for each instance of an object by sampling sets of points that belong to one instance with high probability. In the point cloud registration literature [6, 7], a set of sampled points is called a *base* and denoted by B. Once sampled, this set is matched to *congruent* sets, denoted by U, on the corresponding object model M. Each of the congruent pairs (B, U) defines a unique rigid transform as long as both B and U contain at least 3 points each. For computational reasons, the cardinality is typically limited to 4, i.e., $|B| = |U| = 4$. To maximize the probability that pose candidates for all the instances are generated, a number A_i of base sets $\{B_{ij}\}_{j=1}^{A_i}$ on the image x is sampled, for each object class $i \in \{1, \ldots, K\}$. Each base B_{ij} can be matched to several congruent bases U on the models, and thus defines a large set of candidate poses.

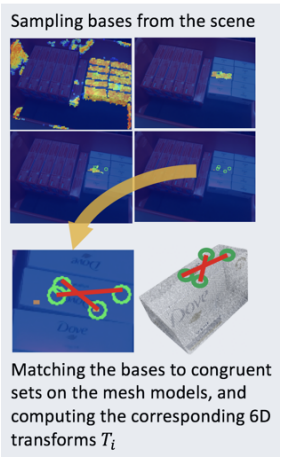

Figure 1: Hypothesis generation process via sampling a base and matching it to a congruent set from the object model.

A. Sampling a Base from a Single Object Instance: The four 3D points that form a base $B = \{b_1, b_2, b_3, b_4 \mid b_{1:4} \in S_l\}$ are sampled from a point cloud S_l, which is the set of 3D points that are labeled as a category l according to the semantic class probabilities P_l. The challenge here is to maximize the joint probability of the four points belonging to the same object instance. The joint probability distribution on the graph formed by the four points is represented by using the Hammersley-Clifford factorization and considering only unary and binary relations between the points:

$$Pr\Big(\big(c(p(b_1), x) = l\big) \wedge \big(b_{1:4} \text{ are all on the same instance}\big)\Big)$$
$$= \frac{1}{Z} \prod_{i=1}^{4} \Big(\phi_{node}^{l}(b_i) \prod_{j=1}^{j<i} \phi_{edge}^{l}(b_i, b_j)\Big),$$

where $p(b_i)$ is used to denote the pixel corresponding to the 3D point b_i and, as before, $c(p, x)$ is used to denote the object class label of pixel p in image x. The unary terms $\phi^l_{node}(b_i)$ are the individual probabilities of labeling pixels $p(b_i)$ with category label l, i.e., $\phi^l_{node}(b_i) = P_l(p(b_i))$. The binary relations are defined as,

$$\phi^l_{edge}(b_i, b_j) = \begin{cases} 1, & \text{if } \texttt{PPF}(b_i, b_j) \in \texttt{PPF}(M_k) \text{ and} \\ & \textit{length-of-shortest-path}(p(b_i), p(b_j)) < \epsilon_l \\ 0, & \text{otherwise.} \end{cases}$$

The *Point-Pair Feature* (PPF) [2] for two base points b_i, b_j with surface normals n_i, n_j is defined as:

$$\texttt{PPF}(b_i, b_j) = (||\ d\ ||_2, \angle(n_i, d), \angle(n_j, d), \angle(n_i, n_j)),$$

where $d = b_i - b_j$ is the vector from b_i to b_j and n_i, n_j are local surface normals. $\texttt{PPF}(M_k)$ is the set of point-pair features pre-computed on the object model M_k of object category k. The *length-of-shortest-path* term is the shortest distance on a graph G, where the vertices are the image pixels and an edge connects two pixels p_i and p_j if and only if: $P_B(p_i) < \delta$, $P_B(p_j) < \delta$, and p_i and p_j are adjacent pixels in the image (each pixel has at most eight adjacent pixels). In other terms, there is an edge between two adjacent pixels if both pixels have a low probability (less than a threshold δ) of being on the boundary of an object. Thus, *length-of-shortest-path*(p_i, p_j) is the length of the shortest path between the two nodes on G, and it is obtained using a breadth-first search.

B. Avoiding Oversampling and Achieving Coverage: The sampling process is dynamically adapted so as to avoid repetitively selecting the same image regions. In particular, after a base is sampled, a decay factor $\gamma \in [0, 1)$ is multiplied to the potential $\phi^l_{node}(b_i)$ of every point b_i that belongs to the same segment as b_i, i.e., $\phi^l_{node}(b_i) \leftarrow \gamma\phi^l_{node}(b_i)$. Points belonging to the same segments are those encountered during the breadth-first search. This encourages dispersion in the sampling process of the bases, so as to cover all instances of objects in the image and avoid having all the samples concentrated in the region of a single object instance. This becomes an issue when a particular, more prominent object instance has higher semantic class prediction probability than other instances.

C. Matching the Base to its Congruent Sets: Once all the base sets are sampled, a matching process is used for each one of the sampled bases $\{\{B_{ij}\}_{j=1}^{A_i}\}_{i=1}^{K}$ to compute a set $U_{ij} = \{u_1, u_2, u_3, u_4\}$ of 4-points from M_i that are congruent to the sampled bases. The basic idea of 4-point congruent sets was derived from the fact that ratios and distances on objects are invariant across any rigid transformation [8].

This matching process generates a large number of pose candidates $\{\{T_{ij}\}_{j=1}^{H_i}\}_{i=1}^{K}$, several of which are redundant due to different congruent pairs returning similar transforms and the fact that most geometries have symmetry along multiple axes. Thus, this work follows a greedy approach to provide a diverse subset of the population. The approach sorts all pose candidates based on the Largest Common Pointset score [6] and picks pose representatives that are beyond a minimum distance from already selected poses. The distance here takes into account symmetry, and there are separate thresholds for rotation and translation, since these two variables operate in different spaces. The resulting set $\{\{T_{ij}\}_{j=1}^{H_i}\}_{i=1}^{K}$ of pose representatives then undergoes local refinement using ICP.

Bibliography

[1] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother. Learning 6d object pose estimation using 3d object coordinates. In *ECCV*, 2014.

[2] B. Drost, M. Ulrich, N. Navab, and S. Ilic. Model Globally, Match Locally: Efficient and Robust 3D Object Recognition. In *CVPR*, 2010.

[3] S. Hinterstoisser, V. Lepetit, N. Rajkumar, and K. Konolige. Going further with point pair features. In *ECCV*, 2016.

[4] C. Mitash, A. Boularias, and K. Bekris. Robust 6d object pose estimation with stochastic congruent sets. In *BMVC*, 2018.

[5] F. Michel, A. Kirillov, E. Brachmann, A. Krull, S. Gumhold, B. Savchynskyy, and C. Rother. Global hypothesis generation for 6d object pose estimation. In *CVPR*, 2017.

[6] N. Mellado, D. Aiger, and N. Mitra. Super4PCS: Fast Global Pointcloud Registration via Smart Indexing. In *Computer Graphics Forum*, 2014.

[7] J. Huang, T.-H. Kwok, and C. Zhou. V4PCS: Volumetric 4PCS Algorithm for Global Registration. *Journal of Mechanical Design*, 2017.

[8] D. Aiger, N. Mitra, and D. Cohen-Or. 4-points Congruent Sets for Robust Pairwise Surface Registration. In *ACM Transactions on Graphics (TOG)*, 2008.