Improving 6D Pose Estimation of Objects in Clutter via Physics-aware Monte Carlo Tree Search

Chaitanya Mitash, Abdeslam Boularias and Kostas E. Bekris

***Abstract*— This work proposes a process for efficiently searching over combinations of individual object 6D pose hypotheses in cluttered scenes, especially in cases involving occlusions and objects resting on each other. The initial set of candidate object poses is generated from state-of-the-art object detection and global point cloud registration techniques. The best scored pose per object by using these techniques may not be accurate due to overlaps and occlusions. Nevertheless, experimental indications provided in this work show that object poses with lower ranks may be closer to the real poses than ones with high ranks according to registration techniques. This motivates a global optimization process for improving these poses by taking into account scene-level physical interactions between objects. It also implies that the Cartesian product of candidate poses for interacting objects must be searched so as to identify the best scene-level hypothesis. To perform the search efficiently, the candidate poses for each object are clustered so as to reduce their number but still keep a sufficient diversity. Then, searching over the combinations of candidate object poses is performed through a Monte Carlo Tree Search (MCTS) process that uses the similarity between the observed depth image of the scene and a rendering of the scene given the hypothesized pose as a score that guides the search procedure. MCTS handles in a principled way the tradeoff between fine-tuning the most promising poses and exploring new ones, by using the Upper Confidence Bound (UCB) technique. Experimental results indicate that this process is able to quickly identify in cluttered scenes physically-consistent object poses that are significantly closer to ground truth compared to poses found by point cloud registration methods.**

I. Introduction

Robot manipulation systems frequently depend on a perception pipeline that can accurately perform object recognition and six degrees-of-freedom (6-DOF) pose estimation. This is becoming increasingly important as robots are deployed in less structured environments than traditional automation setups. An example application domain corresponds to warehouse automation and logistics as highlighted by the Amazon Robotics Challenge (ARC) [1]. In such tasks, robots have to deal with a large variety of objects potentially placed in complex arrangements, including cluttered scenes where objects are resting on top of each other and are often only partially visible, as shown in Fig. 1.

This work considers a similar setup, where a perception system has access to RGB-D images of objects in clutter, as well as 3D CAD models of the objects, and must provide accurate pose estimation for the entire scene. In this domain, solutions have been developed that use a Convolutional Neural Network (CNN) for object segmentation [2], [3] followed by a 3D model alignment step using point cloud registration techniques for pose estimation [4], [5]. The focus of the current paper is to improve this last step and increase the accuracy of pose estimation by reasoning at a scene-level about the physical interactions between objects.

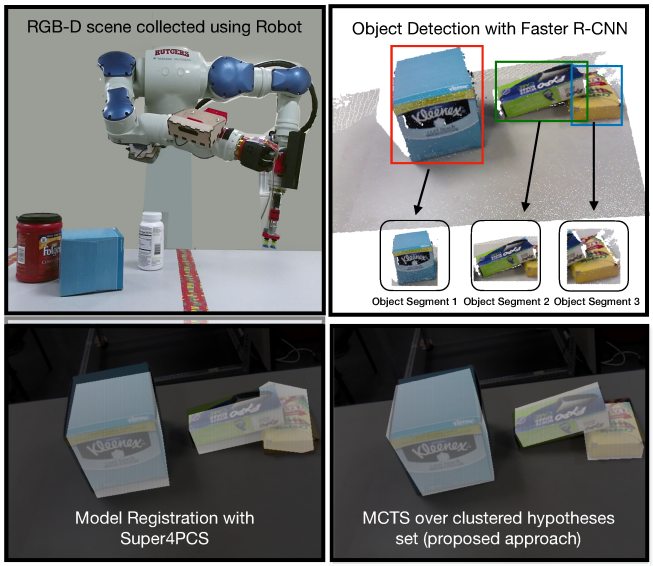

Fig. 1. (top left) A Motoman robot using Intel RealSense to capture RGB-D image of the scene. (top-right) An example scene with 3 objects and resulting segmentation achieved with state-of-the-art method [6]. (bottom left) The pose estimation according to the best hypothesis from Super4PCS [4]. Notice the penetration between the two objects on the right side of the scene. (bottom right) Improved pose estimation via the proposed physics-aware search process, which considers the ensemble of hypotheses returned by Super4PCS.

In particular, existing object-level reasoning for pose estimation can fail on several instances. One reason can be imperfect object detection, which might include parts of other objects, thus guiding the model registration process to optimize for maximum overlap with a mis-segmented object point cloud. Another challenge is the ambiguity that arises when the object is only partially visible, due to occlusion. This results in multiple model placements with similar alignment scores, and thus a failure in producing a unique, accurate pose estimate. These issues were the primary motivation behind hypothesis verification (HV) methods [7], [8]. These techniques follow a pipeline where: (a) they first generate multiple candidate poses per object given feature matching against a model, (b) they create a set of scene-level hypotheses by considering the Cartesian

The authors are with the Computer Science Department of Rutgers University in Piscataway, New Jersey, 08854, USA. Email: {cm1074,kb572,ab1544}@rutgers.edu.

This work is supported by NSF awards IIS-1734492, IIS-1723869 and IIS-1451737. Any opinions or findings expressed in this paper do not necessarily reflect the views of the sponsors.

product of individual object pose candidates, and find the optimal hypothesis with respect to a score defined in terms of similarity with the input scene and geometric constraints.

It has been argued, however, that existing HV methods suffer from critical limitations for pose estimation [9], [10]. The argument is that the optimization in the HV process, may not work because the true poses of the objects may not be included in the set of generated candidates, due to errors in the process for generating these hypotheses. Errors may arise from the fact that the training for detection typically takes place for individual objects and is not able to handle occlusions in the case of cluttered scenes.

This motivated the development of a search method [9], [10] for the best explanation of the observed scene by performing an exhaustive but informed search through rendering all possible scene configurations given a discretization over 3-DOFs, namely (x, y, yaw). The search was formulated as a tree, where nodes corresponded to a subset of objects placed at certain configurations and edges corresponded to adding one more object in the scene. The edge cost was computed based on the similarity of the input image and the rendered scene for the the additional object. An order of object placements over the tree depth was implicitly defined by constraining the child state to never occlude any object in the parent state. This ensured an additive cost as the search progressed, which allows the use of heuristic search.

The current work adapts this idea of tree search to achieve better scalability and increased accuracy, while performing a comprehensive hypothesis verification process by addressing the limitations of such HV techniques. In particular, instead of imposing a discretization, which is difficult to scale to 6-DOF problems, the proposed search is performed over scene hypotheses. In order to address the issue of potentially conflicting candidate object poses, the scene hypotheses are dynamically constructed by introducing a constrained local optimization step over candidate object poses returned by Super4PCS, a fast global model matching method [4]. To limit detection errors that arise in cluttered scene, the proposed method builds on top of a previous contribution [11], which performs clutter-specific autonomous training to get object segments. This paper provides experimental indications that the set of candidate object poses returned by Super4PCS given the clutter-aware training contains object poses that are close enough to the ground truth, however, these might not be the ones that receive the best matching score according to Super4PCS. This is why it becomes necessary to search over the set of candidate poses. Searching over all possible hypotheses returned by Super4PCS, however, is impractical. Thus, this work introduces a clustering approach that identifies a small set of candidate pose representatives that is also diverse enough to express the spread of guesses in the matching process.

The search operates by picking candidate object poses from the set of cluster representatives given a specific order of object placement. This order is defined by considering the dependencies among objects, such as, when an object is stacked on top of another or is occluded by another object. At every expansion of a new node, the method uses the previously placed objects to re-segment the object point cloud. It then performs local point cloud registration as well as physics simulation to place the object in physically consistent poses with respect to the already placed objects and the resting surface. As the ordering considers the physical dependency between objects, the rendering cost is no longer additive and cannot be defined for intermediate nodes of the search tree. For this reason, a Monte Carlo Tree Search (MCTS) [12] approach is used to heuristically guide the search based on evaluation of the rendering cost for the complete assignment of object poses. Specifically, the UCT search method is used with the Upper Confidence Bound (UCB) to trade-off exploration and exploitation in the search tree.

The experimental evaluation of the proposed framework indicates that searching over the space of scene hypotheses in this manner can quickly identify physically realistic poses that are close to ground truth and can significantly improve the quality of the output of global model matching methods.

II. Related Work

This section covers the different approaches that have been applied to object recognition and pose estimation using range data and their relationship to the current work.

A. Feature Matching

A traditional way of approaching this problem has been computing local shape based descriptors in the observed scene and on the 3D CAD object models [13]. Given feature correspondence between the real scene and the model, rigid body transformations can be computed for the models that maximize the correspondences between these descriptors. Some examples of such local descriptors correspond to the SHOT [14], and PFH [15]. There has also been significant use of viewpoint-specific global shape features like CVFH [16]. In this case, features are computed offline for object models viewed from several sampled viewpoints and are matched to the feature corresponding to the object segment retrieved in the observed data to directly get the pose estimate. These methods have gained popularity because of their speed of execution and minimal training requirement. However, the local descriptors are prone to failure in case of occlusion when the keypoints are not visible and the global descriptors are highly dependent on clean object segmentation, which is often difficult to obtain.

B. Progress in Deep Learning

There has been recent success in using deep learning techniques for learning local geometric descriptors [17] and local RGB-D patches [18] for object recognition and pose estimation. Even though this is promising as data driven approaches could help close the gap between 3d geometric models and noisy observed data, it needs access to a large model aligned 3d training dataset [19], which may be difficult to collect. Another technique often used is to perform object segmentation using CNNs trained specifically for the setup [11], [2], [20] and perform point cloud registration methods

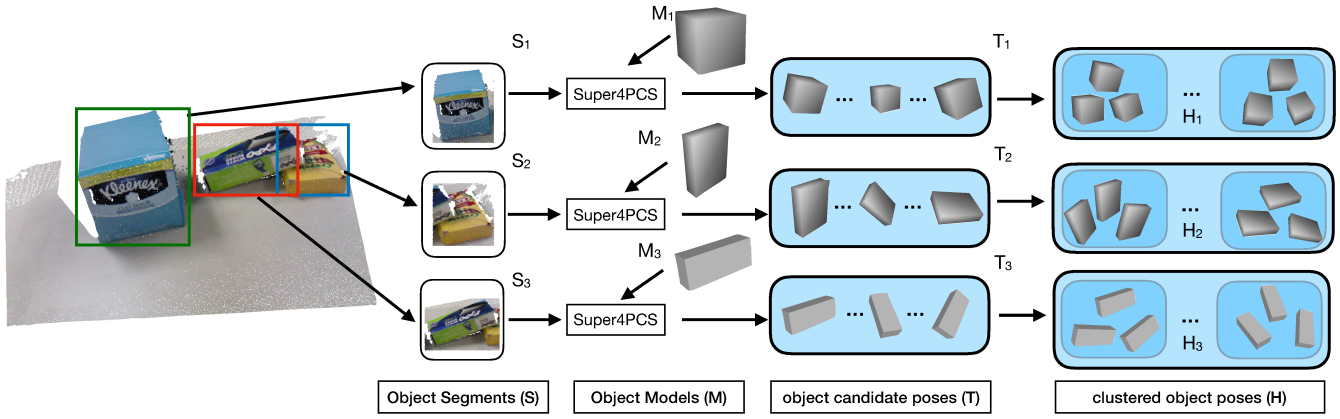

Fig. 2. The image describes the process of hypotheses generation for objects present in the scene. The process starts with extracting object segments $S_{1:3}$ using Faster-RCNN [6], followed by using a global point cloud registration technique [4] to compute a set of possible model transformations ($T_{1:3}$) that corresponds to the respective segments. These transformations are then clustered to produce object specific hypotheses sets ($H_{1:3}$).

[4], [5] to align 3d models to the object segments. This method may fail in the case of over-segmentation or when enough surfaces are not visible.

C. Global Scene-Level Reasoning

A popular approach to resolve conflicts arising from local reasoning is to generate object pose candidates, and perform a Hypothesis Verification (HV) step [21], [8], [22]. The hypotheses generation in most cases occurs using a variant of RANSAC [23], [4]. One of this method's drawbacks is that the generated hypotheses might already be conflicted due to errors in object segmentation and thus performing an optimization over this might not be very useful. Recently, proposed method reasons globally about hypotheses generation process [24]. Nevertheless, this requires explicit training for pixel-wise prediction. Another approach to counter these drawbacks corresponds to an exhaustive but informed search to find the best scene hypotheses over a discrete set of object placement configurations [9], [10]. A tree search formulation as described above was defined to effectively search in 3-DOF space. It is not easy, however, to apply the method for 6-DOF pose estimation due to scalability issues.

This work shows that by training an object detector with an autonomous clutter-aware process [11], it is possible to generate a set of object candidate poses by a fast global point cloud registration method [4], which only has local geometric conflicts. Generating candidate poses in this manner and then applying a search process, which constrains each object expansion to other object placements leads to significant improvements in the final pose estimation results.

III. Approach

The problem involves estimating 6-DOF poses of objects placed in a clutter, the inputs to which are:

- RGB and depth images captured by a sensor;
- a set of 3D CAD models M, one for each object.
- a known 6-DOF pose of the camera: T_c.

The proposed method approaches the problem by (1) generating a set of pose hypotheses for each object present in the scene, and (2) searching efficiently over the set of joint hypotheses for the most globally consistent solution. Global consistency is quantitatively evaluated by a score function. The score function measures the similarity between the actual observed depth image and a rendering of the objects in simulation using their hypothesized poses. The hypothesized poses are adapted during the search process, so as to correspond to poses where the objects are placed in a physically realistic and stable configuration according to a physics engine that simulates rigid object dynamics.

A. Hypothesis Generation

Some of the desired properties for a set of 6D pose hypotheses are the following:

- informed and diverse enough such that the optimal solution is either already contained in the set or a close enough hypothesis exists so that a local optimization process can fine-tune it and return a good result;
- limited in size, as evaluating the dependencies among the hypotheses set for different objects can lead to a combinatorial explosion of possible joint poses and significantly impact the computational efficiency;
- does not require extensive training.

This work considers all of these properties while generating the hypothesis set. The pseudocode for hypothesis generation is presented in Algorithm 1.

Algorithm 1: GEN_HYPOTHESIS(RGB, depth, M, T_c)

$H \leftarrow \{h_O = \emptyset, \forall O \in M\}$;
foreach *object O in the scene* **do**
  $bbox_O \leftarrow$ RCNN_DETECT(RGB, O);
  $P_O \leftarrow$ GET_3DPOINTS(bbox_O, depth, T_c);
  $T_O \leftarrow$ SUPER4PCS(M_O, P_O);
  $\{cluster_{tr}, center_{tr}\} \leftarrow \text{KMEANS}_{tr}(T_O)$;
  foreach *cluster C in* $cluster_{tr}$ **do**
    $\{cluster_{rot}, center_{rot}\} \leftarrow \text{K-KMEANS}_{tr}(C)$;
    $h_O \leftarrow h_O \cup (center_{tr}, center_{rot})$;
  $H \leftarrow H \cup h_O$;
return H;

The algorithm first employs an object detector on the RGB image based on Faster-RCNN [6]. This detector, which uses a VGG16 network architecture [25], was trained with an autonomous training process proposed in prior work [11]. The training involves a large number of realistic synthetic data used to bootstrap a self-learning process to get a high performing detector. For each object O, the corresponding bounding-box ($\texttt{bbox}_O$) returned by the object detector is used to get a segment P_O of the 3D point cloud. Segment P_O is a subset of the point cloud of the scene and contains points from the visible part of object O. Segment P_O frequently contains some points from nearby objects because the bounding box does not perfectly match the shape of the object.

The received point set P_O is then matched to the object model M_O by using the *Super4PCS* algorithm [4]. Typically, the *Super4PCS* algorithm is used to find sets of congruent 4-points in the two point clouds related by a rigid transformation and returns the transformation which results in the best alignment according to the LCP (Largest Common Pointset) measure. Nevertheless, this returned transformation is not necessarily the optimal pose of the object as the point cloud segment extracted via the detection process could include parts of other objects or due to lack of visible surface might not be informative enough to compute the correct solution.

This motivates the consideration of other possible transformations for the objects, which can be evaluated in terms of scene-level consistency. This is the focus of the following section. Thus, the subroutine SUPER4PCS in Algorithm 1 is used to generate a set of possible transformations T_O computed using congruent 4-point sets within a time budget t_o. It is interesting to consider the quality of the hypotheses set returned by the above process by measuring the error between the returned pose hypotheses and the ground truth. For this purpose, a dataset containing 90 object poses was used. Specifically, in each hypotheses set, the pose hypothesis that has the minimum error in terms of rotation is selected as well as the one with the minimum translation error. The mean errors for these candidates over the dataset are shown in Table. I. The results positively indicate the presence of hypotheses close to the true solution. Specifically, the candidate with the minimum rotation error seems almost perfect in rotation and not very far even with respect to translation. Nevertheless, this hypotheses set contained approximately 20,000 elements. It is intractable to evaluate scene level dependencies for that many hypotheses per object as the combined hypotheses set over multiple objects grows exponentially in size.

B. Clustering of Hypotheses

To reduce the cardinality of the hypotheses sets returned by the subroutine SUPER4PCS in Algorithm 1, this work proposes to cluster the 6D poses in each set T_O, given a distance metric. Computing distances between object poses, which are defined in SE(3), in a computationally efficient manner is not trivial [26]. This challenge is further complicated if one would like to consider the symmetry of the geometric models, so that two different poses that result in the same occupied volume given the object's symmetry would get a distance of zero.

To address this issue, a two-level hierarchical clustering approach is followed. The first level involves computing clusters of the pose set in the space of translations (i.e., the clustering occurs in $\mathbb{R}^3$ by using the Euclidean distance and ignoring the object orientations) using a K-Means process [27] to get a smaller set of cluster representatives $\texttt{cluster}_{tr}$. In the second level, the poses that are assigned to the same clusters are further clustered based on a distance computed in the SO(3) space that is specific to the object model, i.e., by considering only the orientation of the corresponding pose. The second clustering step uses a *kernel* K-Means approach [28], where the cluster representative is found by minimizing the sum of kernel distances to every other point in the cluster. This process can be computationally expensive but returns cluster centers that nicely represent the accuracy of the hypotheses set. By using this clustering method, the size of the hypotheses set can be reduced down from 20,000 rigid transforms in T_O to 25 object pose hypotheses in h_O for each object in the scene. The two bottom rows of Table I evaluate the quality of the cluster representatives in the hypotheses set. This evaluation indicates that the clustering process returns hypotheses as cluster representatives that are still close to the true solution. In this way, it provides an effective way of reducing the size of the hypotheses set without sacrificing its diversity.

C. Search

Once the hypotheses set is built for each object in the scene, the task reduces to finding the object poses which lie in the physically consistent neighborhood of the pose candidates that best explain the overall observed scene. In particular, given:

- the observed depth image,
- the number of objects in the scene N,
- a set of 3D mesh models for these objects $\mathrm{M}_{1:N}$,
- and the sets of 6D transformation hypotheses for the objects $\mathrm{h}_{1:N}$ (output of Algorithm 1),

the problem is to search in the hypotheses sets for an N-tuple of poses $\mathrm{T}_{1:N} \mid T_i \in f(h_i)$, i.e. one pose per object. The set $T_{1:N}$ should maximize a global score computed by comparing the observed depth image with the rendered image $\mathrm{R}(\mathrm{T}_{1:N})$ of object models placed at the corresponding poses $\mathrm{T}_{1:N}$. Here, f is the constrained local optimization of the object pose h_i based on physical consistency with respect to the other objects in the scene and also the fact that same points in the scene point cloud cannot be explained by multiple objects simultaneously. Then, the global optimization score is defined as:

$$C(depth, T_{1:N}) = \sum_{p \in P} S(R(T_{1:N})[p], depth[p])$$

where p is a pixel (i, j) of a depth image, $R(T_{1:N})[p]$ is the depth of pixel p in the rendered depth image $R(T_{1:N})$,

Metric for selection	Mean Rotation error	Mean Translation error
[All hypotheses] max. LCP score	11.16°	1.5cm
[All hypotheses] min. rotation error from ground truth	2.11°	2.2cm
[All hypotheses] min. translation error from ground truth	16.33°	0.4cm
[Clustered hypotheses] min. rotation error from ground truth	5.67°	2.5cm
[Clustered hypotheses] min. translation error from ground truth	20.95°	1.7cm

TABLE I

EVALUATING THE QUALITY OF THE HYPOTHESES SET RETURNED BY SUPER4CPS [4] WITH RESPECT TO DIFFERENT METRICS.

$depth[p]$ is the depth of pixel p in the observed depth image `depth`, $P = \{p \mid R(T_{1:N})[p] \neq 0 \text{ or } D[p] \neq 0)\}$ and

$$S(R(T_{1:N})[p], D[p]) = \begin{cases} 1, \text{ if } \mid R(T)[p] - D[p] \mid < \epsilon \\ 0, \text{otherwise} \end{cases}$$

for a predefined precision threshold ϵ. Therefore, score `C` counts the number of non-zero pixels p that have a similar depth in the observed image `D` and in the rendered image `R` within an ϵ threshold. So, overall the objective is to find:

$$T^*_{1:N} = \arg \max_{T_{1:N} \in f(h_{1:N})} C(D, R(T_{1:N})).$$

At this point a combinatorial optimization problem arises so as to identify $T^*_{1:N}$, which is approached with a tree search process. A state in the search-tree corresponds to a subset of objects in the scene and their corresponding poses. The root state s_0 is a null assignment of poses. A state s_d at depth `d` is a placement of d objects at specific poses selected from the hypotheses sets, i.e., $\mathrm{s}_d = \{(M_i, T_i), i = 1 : d\}$ where T_i is the pose chosen for object M_i, which is assigned to depth i. The goal of the tree search is to find a state at depth `N`, which contains a pose assignment for all objects in the scene and maximizes the above mentioned rendering score. Alg. 2 describes the expansion of a state in the tree search process towards this objective.

Algorithm 2: EXPAND(s_d, T_{d+1}, P_{d+1})

1 **if** $d = N$ **then**
2 | return NULL;
3 **foreach** *object* $O \in s_d$ **do**
4 | $P_{d+1} \leftarrow P_{d+1}$ - points_explained(P_{d+1}, M_O, T_O);
5 $T_{d+1} \leftarrow$ TrICP(T_{d+1}, P_{d+1});
6 $T_{d+1} \leftarrow$ PHYSIM$(T_{d+1}, T_{1:d})$;
7 $s_{d+1} \leftarrow s_d \cup T_{d+1}$;
8 return s_{d+1};

The EXPAND routine takes as input, the state s_d at depth d in the tree, the point cloud segment corresponding to the next object to be placed P_{d+1} and the pose hypothesis for the next object T_{d+1}. Lines 3-4 of the algorithm iterate over all the objects already placed in state s_d and remove points already explained by these object placements from the point cloud segment of the next object to be placed. This step helps in achieving much better segmentation, which is utilized by the local optimization step of Trimmed ICP [29] in line 5. The poses of objects in state s_d physically constrain the pose of the new object to be placed. For this reason, a rigid body physics simulation is performed in line 6. The physics simulation is initialized by inserting the new object into the scene at pose T_{d+1}, while the previously inserted objects in the current search branch are placed in the poses $T_{1:d}$. A physics engine is used to ensure that the newly placed object attains a physically realistic configuration (stable and no penetration) with respect to other objects and the table under the effect of gravity. After a fixed number of simulation steps, the new pose T_{d+1} of the object is appended to the previous state to get the successor state s_{d+1}.

The above primitive is used to search over the tree of possible object poses. The objective is to exploit the contextual ordering of object placements given information from physics and occlusion. This does not allow to define an additive rendering score over the search depth as in previous work [9], which demands the object placement to not occlude any part of the already placed objects. Instead, this work proposes to use a heuristic search approach based on Monte Carlo Tree Search utilizing Upper Confidence Bound formulation [12] to trade off exploration and exploitation in the expansion process. The pseudocode for the search is presented in Alg. 3.

To effectively utilize the constrained expansion of states, an order of object placements needs to be considered. This information is encoded in a *dependency graph*, which is a directed acyclic graph that provides a partial ordering of object placements but also encodes the interdependency of objects. The vertices of the *dependency graph* correspond to the objects in the observed scene. Simple rules are established to compute this graph based on the detected segments $\mathtt{P}_{1:N}$ for objects $\mathrm{O}_{1:N}$.

- A directed edge connects object O_i to object O_j if the the x-y projection of P_i in the world frame intersects with the x-y projection of P_j and the z-coordinate (negative gravity direction) of the centroid for P_j is greater than that of P_i.
- A directed edge connects object O_i to object O_j if the detected bounding-box of O_i intersects with that of O_j and the z-coordinate of the centroid of P_j in camera frame (normal to the camera) is greater than that of P_i.

The information regarding independency of objects helps to significantly speed up the search as the independent objects are then evaluated in different search trees and prevents exponential growth of the tree. This results in `K` ordered list of objects, $\mathtt{L}_{1:K}$ from the step GET_DEPENDENCY each of which are used to run independent tree searches for pose computation. The `MCTS` proceeds by selecting the first unexpanded node starting from the root state.

Algorithm 3: SEARCH

Function *MCTS* ($M_{1:N}, P_{1:N}, h_{1:N}$)
  T $\leftarrow \emptyset$;
  $L_{1:K} \leftarrow$ GET_DEPENDENCY($P_{1:N}$);
  foreach $L \in L_{1:K}$ **do**
    $s_0 \leftarrow \emptyset$;
    $best\_render\_score \leftarrow 0$;
    $best\_state \leftarrow s_0$;
    while $search\_time < t_{th}$ **do**
      $s_i \leftarrow$ SELECT($s_0, P_{1:N}$);
      $\{s_R, R\} \leftarrow$ RANDOM_POLICY($s_i, P_{1:N}$);
      if $R > best\_render\_score$ **then**
        $best\_render\_score \leftarrow R$;
        $best\_state \leftarrow s_R$;
      BACKUP_REWARD(s_i, R);
    $T \leftarrow T \cup best\_state$;
  return T;

Function *SELECT* ($s, P_{1:N}$)
  while $depth(s) < N$ **do**
    if *s has unexpanded child* **then**
      level $\leftarrow depth(s) + 1$;
      $T_{level} \leftarrow$ NEXT_POSE_HYPOTHESIS(h_{level});
      return EXPAND($s, T_{level}, depth(s), P_{level}$);
    else
      s $\leftarrow$ GET_BEST_CHILD(s)
  return s;

Function *GET_BEST_CHILD* (*s*)
  return $\arg\max_{s' \in succ(s)} \frac{h(s')}{n(s')} + \alpha\sqrt{\frac{2log(n(s))}{n(s')}}$;

Function *RANDOM_POLICY* ($s, P_{1:N}$)
  while $depth(s) < N$ **do**
    $level \leftarrow depth(s) + 1$;
    $T_{level} \leftarrow$ GET_RANDOM_HYPOTHESIS(H_{level});
    $s \leftarrow$ EXPAND(s, T_{level}, depth(s), P_{level});
  return {s, render(s)};

Function *BACKUP_REWARD* (s, R)
  while $s \neq$ *NULL* **do**
    n(s) $\leftarrow$ n(s) + 1;
    h(s) $\leftarrow$ h(s) + R;
    s $\leftarrow$ parent(s);

The selection takes place based on a reward associated with each state. The reward is the mean of the rendering score received at any leaf node in its subtree along with a penalty based on the number of times this subtree has been expanded relative to its parent. This is indicated in the subroutine GET_BEST_CHILD, where h(s) is the maintained score and n(s) stores the number of times the subtree corresponding to the state s has been expanded. The selected state is then expanded by using a RANDOM_POLICY, which in this case is picking a random object pose hypothesis for each of the succeeding objects while performing the constrained local optimization at each step. The output of this policy is the final rendering score of the generated scene hypotheses. This reward is then backpropogated in the step BACKUP_REWARD to the preceeding nodes. Thus, the search is guided to the part of the tree, which gets a good rendering score but also explores other portions, which have not been expanded enough (controlled by the parameter α).

IV. EVALUATION

This section discusses the dataset and metrics used for evaluating the approach, and an analysis of intermediate results explaining the choice of the system's components.

A. Dataset and Evaluation Metric

A dataset of RGB-D images was collected and ground truth `6-DOF` poses were labeled for each object in the image. The dataset contains table-top scenes with 11 objects from the 2016 Amazon Picking Challenge [30] with the objects representing different object geometries. Each scene contains multiple objects and the object placement is a mix of independent object placements, objects with physical dependencies such as one stacked on/or supporting the other object and occlusions. The dataset was collected using an Intel RealSense sensor mounted over a Motoman robotic manipulator. The manual labeling was achieved by aligning 3D CAD models to the point cloud extracted from the sensor. The captured scene expresses three different levels of interaction between objects, namely, independent object placement where an object is physically independent of the rest of objects, two-object dependencies where an object depends on another, and three object dependencies where an object depends on two other objects.

The evaluation is performed by computing the error in translation, which is the Euclidean distance of an object's center compared to its ground truth center (in centimeters). The error in rotation is computed by first transforming the computed rotation to the frame attached to the object at ground truth (in degrees). The rotation error is the average of the roll, yaw and pitch angles of the transformation between the returned rotation and the ground truth one, while taking into account the object's symmetries, which may allow multiple ground truth rotations. The results provide the mean of the errors of all the objects in the dataset.

B. Pose Estimation without Search

Evaluation was first performed over methods that do not perform any scene level or global reasoning. These approaches trust the segments returned by the object segmentation module and perform model matching followed by local refinement to compute object poses. The results of performing pose estimation over the collected dataset with some of these techniques are presented in Table II. The `APC-Vision-Toolbox` [2] is the system developed by Team MIT-Princeton for Amazon Picking Challenge 2016. The system uses a Fully Convolutional Network (`FCN`) [31] to get pixel level segmentation of objects in the scene, then uses Principal Component Analysis (`PCA`) for pose initialization, followed by `ICP` [5] to get the final object pose. This system was designed for shelf and tote environments

Method	One-object Dependencies		Two-objects Dependencies		Three-objects Dependencies		All	
	Rotation Error	Translation Error	Rotation Error	Translation Error	Rotation Error	Translation Error	Rotation Error	Translation Error
APC-Vision-Toolbox [2]	15.5°	3.4 cm	26.3°	5.5 cm	17.5°	5.0 cm	21.2°	4.8 cm
faster-RCNN + PCA + ICP [11]	8.4°	1.3 cm	13.1°	2.0 cm	12.3°	1.8 cm	11.6°	1.7 cm
faster-RCNN + Super4PCS + ICP [3], [11]	2.4°	0.8 cm	14.8°	1.7 cm	12.1°	2.1 cm	10.5°	1.5 cm
PHYSIM-Heuristic (depth + LCP)	2.8°	1.1 cm	5.8°	1.4 cm	12.5°	3.1 cm	6.3°	1.7 cm
PHYSIM-MCTS (proposed approach)	2.3°	1.1 cm	5.8°	1.2 cm	5.0°	1.8 cm	4.6°	1.3 cm

TABLE II

COMPARING OUR APPROACH WITH DIFFERENT POSE ESTIMATION TECHNIQUES

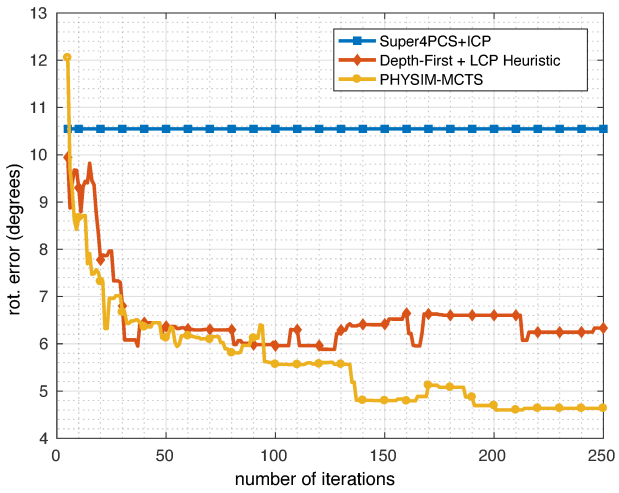

Fig. 3. Rotation error in degrees as a function of the number of iterations.

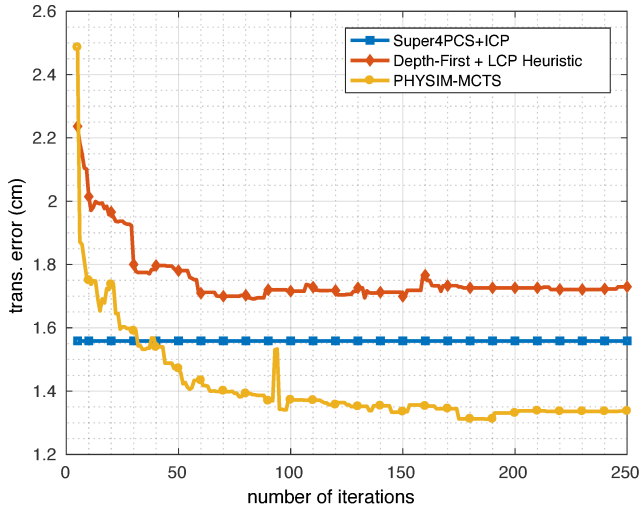

Fig. 4. Translation error in cm as a function of the number of iterations.

and often relies on multiple views of the scene. Thus, the high error in pose estimates could be attributed to the low recall percentage in retrieving object segment achieved by the semantic segmentation method, which in turn resulted in the segment not having enough information to compute a unique pose estimate. The second system tested uses a `Faster-RCNN`-based object detector trained with a setup-specific and autonomously generated dataset [11]. The point cloud segments extracted from the bounding box detections were used to perform pose estimation using two different approaches: i) `PCA` followed by `ICP` and ii) `Super4PCS` followed by `ICP` [5]. Even though the detector succeeded in providing a high recall object segment on most occasions, in the best case the mean rotation error using local approaches was still high (10.5°). This was sometimes due to bounding boxes containing parts of other object segments, or due to occlusions. Reasoning only at a local object-level does not resolve these issues.

C. Pose Estimation with the proposed approach

The proposed search framework was used to perform pose estimation on the dataset. In each scene, the dependency graph structure was used to get the order of object placement and initialize the independent search trees. Then, the object detection was performed using `Faster-RCNN` and `Super4PCS` was used to generate pose candidates, which were clustered to get 25 representatives per object. The search is performed over the combined set of object candidates and the output of the search is an anytime pose estimate based on the best rendering score. The stopping criterion for the searches was defined by a maximum number of node expansions in the tree, set to 250, where each expansion corresponds to a physics simulation with *Bullet* and a rendering with *OpenGL*, with a mean expansion time of ~ 0.2 secs per node. The search was initially performed using a depth-first heuristic combined with the LCP score returned by the Super4PCS for the pose candidates. The results from this approach, `PHYSIM-Heuristic (depth + LCP)`, are shown in Table II, which indicates that it might be useful to use these heuristics if the tree depth is low (one and two object dependencies). As the number of object dependencies grow, however, one needs to perform more exploration. For three-object dependencies, when using 250 expansions, this heuristic search provided poor performance. The UCT Monte Carlo Tree Search was used to perform the search, with upper confidence bounds to trade off exploration and exploitation. The exploration parameter was set to a high value ($\alpha = 5000$), to allow the search to initially look into more branches while still preferring the ones which give a high rendering score. This helped in speeding up the search process significantly, and a much better solution could be reached within the same time. The plots in Fig. 3 and Fig. 4 captures the anytime results from the two heuristic search approaches.

D. Limitations

One of the limitations of global reasoning as is in this approach is the large amount of time required for computing and searching over an extensive hypotheses set. Particularly, due to the hierarchical clustering approach, which was adapted to consider object specific distances, the hypotheses generation time for an object can be in the order of multiple seconds. The search process, which seemed to converge to good solutions with 150 expansions for three-object depen-

dencies, takes approximately 30 seconds. Nevertheless, both of these processes are highly parallelizable. Future work can perform the hypotheses generation and the search with parallel computing. Even though the use of bounding boxes to compute 3D point segments gives a high recall in terms of obtaining the object point segment, and the proposed system also addresses any imprecision which might occur in these segments by performing a constrained segmentation, sometimes an error which occurs in the initial part of object placement could lead to failures that need to be addressed.

V. Discussion

This work provides a novel way of performing pose estimation for objects placed in clutter by efficiently searching for the best scene explanation over the space of physically consistent scene configurations. It also provides a method to construct these sets of scene configurations by using state-of-the-art object detection and model registration techniques which by themselves are not sufficient to give a desirable pose estimate for objects. The evaluations indicate significant performance improvement in the estimated pose using the proposed search, when compared to systems which do not reason globally. Moreover, the use of Monte Carlo Tree search with the scene rendering cost evaluated over physically simulated scenes makes the search tractable for handling multi-object dependencies.

References

[1] N. Correll, K. E. 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," *IEEE Trans. on Automation Science and Engineering (T-ASE)*, 2016.

[2] A. Zeng, K.-T. Yu, S. Song, D. Suo, E. Walker Jr, A. Rodriguez, and J. Xiao, "Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[3] C. Hernandez, M. Bharatheesha, W. Ko, H. Gaiser, J. Tan, K. van Deurzen, M. de Vries, B. Van Mil, J. van Egmond, R. Burger, *et al.*, "Team delft's robot winner of the amazon picking challenge 2016," *arXiv preprint arXiv:1610.05514*, 2016.

[4] N. Mellado, D. Aiger, and N. K. Mitra, "Super 4pcs fast global pointcloud registration via smart indexing," *Computer Graphics Forum. Vol. 33. No. 5*, 2014.

[5] P. J. Besl and N. D. McKay, "Method for registration of 3D shapes," *International Society for Optics and Photonics*, 1992.

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 91–99.

[7] A. Aldoma, F. Tombari, L. Di Stefano, and M. Vincze, "A global hypothesis verification framework for 3d object recognition in clutter," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 7, pp. 1383–1396, 2016.

[8] A. Aldoma, F. Tombari, J. Prankl, A. Richtsfeld, L. Di Stefano, and M. Vincze, "Multimodal cue integration through hypotheses verification for rgb-d object recognition and 6dof pose estimation," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 2104–2111.

[9] V. Narayanan and M. Likhachev, "Perch: Perception via search for multi-object recognition and localization," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5052–5059.

[10] ——, "Discriminatively-guided deliberative perception for pose estimation of multiple 3d object instances." in *Robotics: Science and Systems*, 2016.

[11] C. Mitash, K. E. Bekris, and A. Boularias, "A Self-supervised Learning System for Object Detection using Physics Simulation and Multi-view Pose Estimation," in *IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, 2017.

[12] L. Kocsis and C. Szepesvári, "Bandit based monte-carlo planning," in *ECML*, vol. 6. Springer, 2006, pp. 282–293.

[13] A. Aldoma, Z.-C. Marton, F. Tombari, W. Wohlkinger, C. Potthast, B. Zeisl, R. B. Rusu, S. Gedikli, and M. Vincze, "Tutorial: Point cloud library: Three-dimensional object recognition and 6 dof pose estimation," *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 80–91, 2012.

[14] S. Salti, F. Tombari, and L. Di Stefano, "Shot: Unique signatures of histograms for surface and texture description," *Computer Vision and Image Understanding*, vol. 125, pp. 251–264, 2014.

[15] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz, "Aligning point cloud views using persistent feature histograms," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. IEEE, 2008, pp. 3384–3391.

[16] A. Aldoma, M. Vincze, N. Blodow, D. Gossow, S. Gedikli, R. B. Rusu, and G. Bradski, "Cad-model recognition and 6dof pose estimation using 3d cues," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*. IEEE, 2011, pp. 585–592.

[17] A. Zeng, S. Song, M. Niessner, M. Fisher, J. Xiao, and T. Funkhouser, "3dmatch: Learning local geometric descriptors from rgb-d reconstructions," *arXiv preprint arXiv:1603.08182*, 2016.

[18] W. Kehl, F. Milletari, F. Tombari, S. Ilic, and N. Navab, "Deep learning of local rgb-d patches for 3d object detection and 6d pose estimation," in *European Conference on Computer Vision*. Springer, 2016, pp. 205–220.

[19] C. Rennie, R. Shome, K. E. Bekris, and A. F. De Souza, "A dataset for improved rgbd-based object detection and pose estimation for warehouse pick-and-place," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 1179 – 1185, 2016.

[20] M. Schwarz, A. Milan, A. S. Periyasamy, and S. Behnke, "Rgb-d object detection and semantic segmentation for autonomous manipulation in clutter," *The International Journal of Robotics Research*, p. 0278364917713117, 2016.

[21] A. Aldoma, F. Tombari, L. Di Stefano, and M. Vincze, "A global hypotheses verification method for 3d object recognition," in *European Conference on Computer Vision*. Springer, 2012.

[22] S. Akizuki and M. Hashimoto, "Physical reasoning for 3d object recognition using global hypothesis verification," in *Computer Vision–ECCV 2016 Workshops*. Springer, 2016, pp. 595–605.

[23] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, 1981.

[24] F. Michel, A. Kirillov, E. Brachmann, A. Krull, S. Gumhold, B. Savchynskyy, and C. Rother, "Global hypothesis generation for 6d object pose estimation," *arXiv preprint arXiv:1612.02287*, 2016.

[25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[26] L. Zhang, Y. J. Kim, and D. Manocha, "C-dist: efficient distance computation for rigid and articulated models in configuration space," in *Proceedings of the 2007 ACM symposium on Solid and physical modeling*. ACM, 2007, pp. 159–169.

[27] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.

[28] I. S. Dhillon, Y. Guan, and B. Kulis, "Kernel k-means: spectral clustering and normalized cuts," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004, pp. 551–556.

[29] D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek, "The trimmed iterative closest point algorithm," in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, vol. 3. IEEE, 2002, pp. 545–548.

[30] (2016) Official website of amazon picking challenge. [Online]. Available: http://amazonpickingchallenge.org

[31] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.