Semantic Robot Programming for Goal-Directed Manipulation in Cluttered Scenes

Zhen Zeng Zheming Zhou Zhiqiang Sui Odest Chadwicke Jenkins

***Abstract*— Service robots are becoming increasingly capable of assisting people with their daily activities. They require, however, an interface that allows the user to easily communicate the goal of a task, such as how a dinner table needs to be set up. In this work, we present a semantic robot programming paradigm where the user can directly program the goal of a task on a robot by providing a snapshot of the goal scene. The robot then parses the goal scene into object poses and inter-object relations. We propose a discriminatively informed generative scene estimation method (*DIGEST*) to estimate the initial and goal states of the world, and its effectiveness is demonstrated on public household occlusion dataset and our cluttered scene dataset. With the scene perception capability provided by *DIGEST*, even when faced with different initial states of the world, the robot is able to perform goal-directed manipulations to reach the goal. We evaluate our work on a real robot performing tray-setting tasks.**

I. Introduction

Many service robot task scenarios, such as setting up a dinner table, organizing a shelf, require the user to be able to communicate the desired world state with the robot. More specifically, how is the dinner table to be set or the shelf organized, that is, what are the objects involved in the task, what are the desired poses of those objects, and what should be the inter-object relations. For users to easily communicate the goal of such tasks with the robot, we present a semantic robot programming paradigm where the robot is able to parse a snapshot of the goal state of the world and then perform goal-directed object manipulations to reach the goal state from different initial states.

Goal-directed manipulation requires a true closing of the loop between perception and action, beyond the existing intellectual silos. Advances in object detection [13], [28] have made computers more able to filtering background noise and focus attention on objects of interest. However, it remains scientifically unclear whether such methods in computer vision and their assumptions will generically apply to robot perception suitable for goal-directed manipulation. This circumstance has given rise to new approaches to semantic mapping [20], [31], [17] to partition the a robot's environment into perceivable objects with actionable affordances.

We posit semantic mapping offers a springboard to new form of robot programming, **semantic robot programming**, where semantic maps provide a generic abstraction layer for robot programming. In our approach to this problem, we must bridge the gap of interoperation between semantic mapping and existing methods for goal-directed task planning [12], [21], grasp planning [36] and motion planning [33]. To achieve a viable level of interoperation, we propose methods for scene estimation from robot sensing that can then be expressed axiomatically for ready use with modern task, grasp, and motion planning systems. The resulting of closing this loop with a semantic abstraction layer is envisioned to enable portable robot-executable expressions accessible across a variety of modalities, including: natural language, visual programming, and put-that-there gesturing [4], [19].

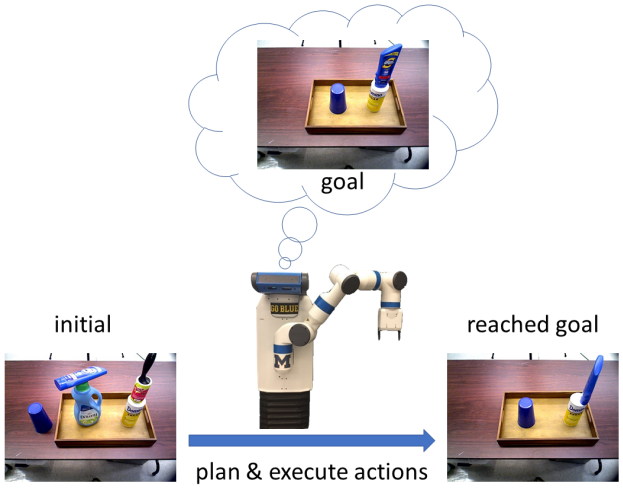

Fig. 1: A robot preparing a tray through goal-directed manipulations. Given the observation of the user desired goal state and the initial state of the tabletop workspace, the robot first perceives the axiomatic scene graph of the goal and initial state, and then plan and execute goal-directed actions to prepare the tray the way the user desires.

In our semantic robot programming paradigm, we bring the discriminative object detection and generative pose estimation to estimate the initial and goal state of the world together. After which the robot can plan and execute goal-directed manipulations to transit the world from the initial to the goal state. We argue that perceiving the state of the world as a scene graph is a critical missing piece for goal-directed manipulation. Our contributions are two-fold:

- a discriminatively informed generative method (*DIGEST*) to estimate the 6 DOF poses of objects in cluttered scenes, assuming the number of objects present in the scene is known.
- a semantic robot programming paradigm that enables robot to autonomously perform goal-directed object

Z. Zeng, Z. Zhou, Z. Sui and O.C. Jenkins are with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, USA, 48109-2121

manipulations in cluttered scenes to achieve a goal as specified by the user

We evaluate our paradigm in tray-setting tasks on the Michigan Progress Fetch robot. We benchmark the performance of *DIGEST* on a household occlusion dataset [2] and our cluttered scene dataset. We demonstrate that our paradigm is effective in understanding the goal of a task given a snapshot of the goal scene. And, the robot is able to plan and execute goal-directed manipulation actions to reach the goal from various initial states of the world. We also demonstrate that *DIGEST* outperforms the state-of-the-art method D2P [27] and other baseline methods in scene perception with fewer assumptions of prior knowledge.

II. Related Work

Programming by Demonstration (PbD) and scene perception for manipulation are closely related to our work. We share similar motivation with research work in PbD field, that is, to enable users to effectively communicate with robots on manipulation tasks. Our work deals with problems in scene perception for manipulation, with an emphasis on goal-directed manipulation.

A. Programming by Demonstrations

To improve communication of tasks from a user to a service robot, existing research has focused on learning low-level skills from users. Different approaches have been proposed in Programming by Demonstration (PbD) for low-level learning of skills, such as trajectories [26] [1] and control policy [6] [15] in robot *configuration space*. These methods are inherently limited to world states in *workspace* that are similar to the ones in the demonstrations. By representing the goal of a task in the *workspace* instead of in the *configuration space*, goal-directed manipulation can reason and plan its actions to reach the goal from arbitrary initial world states.

Other work has focused on the high-level aspects of a task. Veeraraghavan et al. [38] propose learning high level action plan for a repetitive ball collection task from demonstrations. Ekvall et al. [10] focus on learning task goals and use a task planner to reach the goal. Chao et al. [5] provide an interface for the user to teach task goals in a tabletop workspace. However, these methods wind up simplifying the scene perception problem by using planar objects, box-like objects or objects with distinguishing colors, that are far from real world scenarios. Recently, Yang et al. [39] have proposed learning action plans in real world scenario, similar to our robot programming paradigm that works with real world objects.

1) Scene Perception for Manipulation: Being able to perceive objects in real world scenarios and act on them remains a challenging. Some works are able to extract grasping point [7], [22], [35] in point cloud data, however, their methods do not provide a structural understanding of the scene, failing to support goal-directed manipulation on objects.

Although not directly targeted at scene perception for manipulation, research works on object pose estimation are highly related to our work. Feature-based object pose estimation methods suh as spin images [18], FPFH [29], OUR-CVFH [3] and VFH [30], rely on feature matching between the object model and observation, however, the problem is that the performance of feature-based methods degrades as the environment becomes more cluttered and key features are occluded. Recently, Narayanan et al. proposed D2P [27], which outperforms feature-based method OUR-CVFH on the household occlusion dataset [2]. D2P renders multiple scene hypotheses, and use A* to search for the hypothesis that best explains the observation. In our experiments, we demonstrate that our proposed scene estimation method *DIGEST* outperforms D2P on the household occlusion dataset.

To plan goal-directed manipulations, knowing the object poses is not sufficient, however. The robot must have a structural understanding of the scene, that is, the inter-object spatial relations. Given observations of the scene, our work estimates a scene graph that represent the scene structure. Liu et al. [23] also estimate a scene graph given observations, however, their approach approximates objects as oriented bounding boxes. Sui et al. proposed a generative approach (AxMC) [34] for scene graph estimation and use Markov Chain Monte Carlo (MCMC) to search for the best scene graph hypothesis that explains the observations.

Both D2P and AxMC assume that the robot knows what objects are present in the scene, and objects are standing in their upright poses, which means that these two methods can only estimate 3 DOF poses of objects (i.e., x, y, theta). However, these two assumptions are too strong in real world scenarios. Instead, our scene estimation method *DIGEST* does not rely on any of these assumptions, with the result that it can estimate 6 DOF poses of objects, as long as the number of objects in the scene is known.

III. Problem Statement

We assume that the robot knows the number of objects N_c present in the scene, 3D geometries $\mathbf{V} = \{v_1, \cdots, v_k\}$ for a set of objects. This robot is capable of performing a set of manipulation actions $\mathbf{A} = \{a_1, \cdots, a_n\}$ with known pre-conditions and post-conditions on these objects. We assume as given RGB-D observation of the goal scene o_G specified by the user at time t, and the current scene o_I at a later time $t+k$. Our objective is to infer the goal scene graph s_G and the initial scene graph s_I, respectively, then, plan a sequence of goal-directed manipulation actions $\{a_i, \cdots, a_j\}$ to rearrange objects in the world such that the inter-object relations in s_G are satisfied.

We use a list of axiomatic assertions to describe a scene as a scene graph. The scene state at time t is expressed as a scene graph $s_t = \{h_t^i(\mathbf{x})\}_{i=0}^{M}$, where $h_t^i \in \{exist, clear, on, in\}$ is an axiomatic assertion parameterized by $\mathbf{x_t} = \{w_t^j(q_t^j, v_t^j)\}_{j=0}^{N_c}$, with $w_t^j(q_t^j, v_t^j)$ representing that at time t, object w_t^j has pose q_t^j and geometry v^j, N_c being the number of objects, and M being the total number of axiomatic assertions. In our work, the assertions are limited to spatial relations that can be tested geometrically.

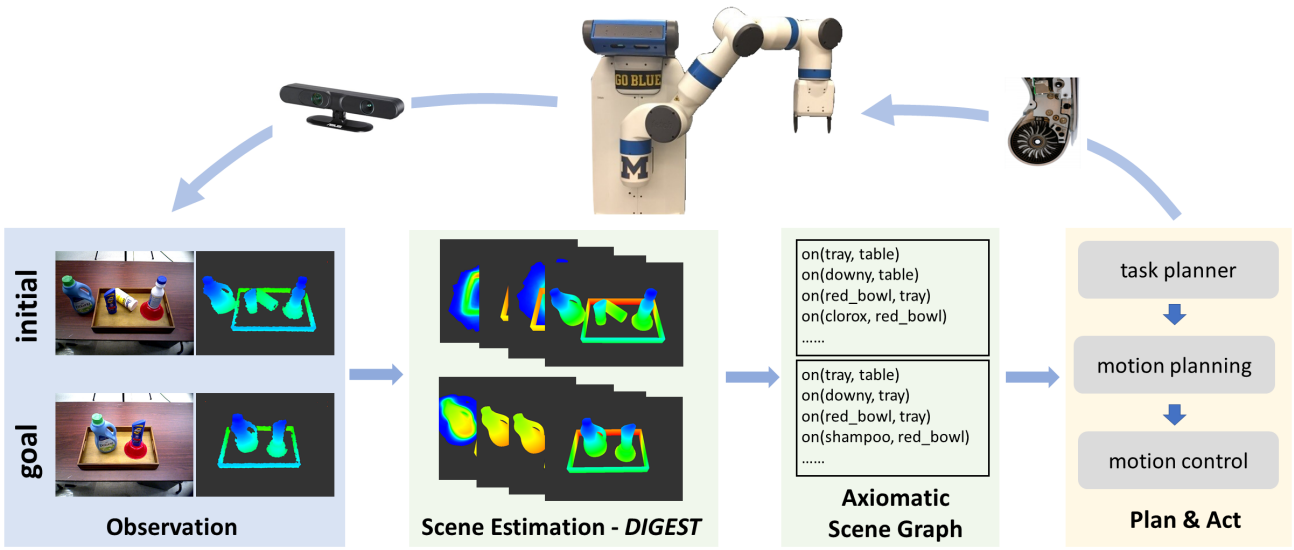

Fig. 2: Our goal-directed robot programming has three stages: 1) Given the RGB-D observation of the goal and initial scene, we use the proposed scene estimation method *DIGEST* to detect object and estimate the 6 DOF pose of objects; 2) Axiomatic scene graphs can be derived from the estimated object poses, which express the inter-object spatial relations; 3) By describing the goal and initial scene graph by PDDL, the robot uses a task planner (e.g., STRIPS) to plan and execute a sequence of goal-directed actions to reorganize the objects in the scene, reaching the same inter-object relations in the goal scene graph.

The 6 DOF pose $q_t^j = [x_t^j, y_t^j, z_t^j, \phi_t^j, \psi_t^j, \theta_t^j]$ of each object is estimated, consisting 3D position (x_t^j, y_t^j, z_t^j) and orientation $(\phi_t^j, \psi_t^j, \theta_t^j)$. The scene graph can be inferred from the estimated object poses, as explained later in Section IV-B.

IV. Methods

Our paradigm consists of the perception of goal and initial scene states, and the planning and execution stages, as shown in Figure 2. We aim to decouple probabilistic scene state estimation and action planning so that the robot takes action based on the current estimate of the scene. Given observations of a cluttered scene, the generative sampling process of object poses is informed by a discriminative object detector. A scene graph that encodes the inter-object relations is geometrically inferred from the estimated object poses, which is then used in the task planner for goal-directed manipulation.

A. Cluttered Scene Estimation

Given observation o_t as the depth image of a cluttered scene at time t, the objective is to estimate the object poses $q_t^j, j = 1, \cdots, N_c$. We utilize the discriminative power of a pre-trained object detector to first obtain a set of bounding boxes with object labels. These bounding boxes are used to guide the generative process of scene hypotheses sampling. An overview of the cluttered scene estimation is as illustrated in Figure 3.

1) Object Detection and Scene Hypotheses Generation: Given an RGB image, m bounding boxes are detected by the object detector. We use B_i $(0 \le i \le m)$ to denote the bounding box. In the output of the object detector, each B_i is associated with a list of object detection confidence $v(L_j|B_i)$, where L_j is the object class. For each B_i, we generate an object candidate C_i,

$$C_i = \{\arg\max_{L_j} v(L_j|B_i), B_i\} \tag{1}$$

which is a set including the object label with the highest confidence measure and the associated bounding box.

For m generated candidates, the number of scene hypotheses h equals to N_c chooses m, i.e.,

$$h = \begin{cases} {}^{m}C_{N_c}, & \text{if } N_c \le m \\ 1, & \text{otherwise} \end{cases} \tag{2}$$

Thus if the number of candidates is greater or equal to the number of objects in the scene, each scene hypothesis H_i contains a combination of N_c candidates selected from m candidates. If the number of candidates is less than N_c, just one scene hypothesis with m candidates will be generated.

2) Bootstrap Filtering for Pose Estimation: Each scene hypothesis H_i is modeled as a random state variable x_t (with a little abuse of notation of x_t), composed of a set of real-valued object poses. We model the inference of the state from robot observation as a Bayesian filter problem. Compared to traditional Bayesian filter problems, we have only one observation, that is, a snapshot of the scene instead of a history of observations. Thus, we apply Iterated Likelihood Weighting [25] to bootstrap the scene estimation process, where $z_1 = z_2 = \cdots = z_t$ and the state transition in the action model is replaced by a zero-mean Gaussian noise.

$$p(x_t|z_{1:t}) \propto p(z_t|x_t) \int_{x_t} p(x_t|x_{t-1}, u_{t-1}) p(x_{t-1}|z_{1:t-1}) \mathrm{d}x_{t-1}$$

We approximate the belief distribution by a collection of N particles $\{x_t^{(j)}$ weighted by $w_t^{(j)}\}_{j=1}^N$,

$$p(x_t|z_{1:t}) \propto p(z_t|x_t) \sum_j w_{t-1}^{(j)} p(x_t|x_{t-1}^{(j)}, u_{t-1}) \tag{3}$$

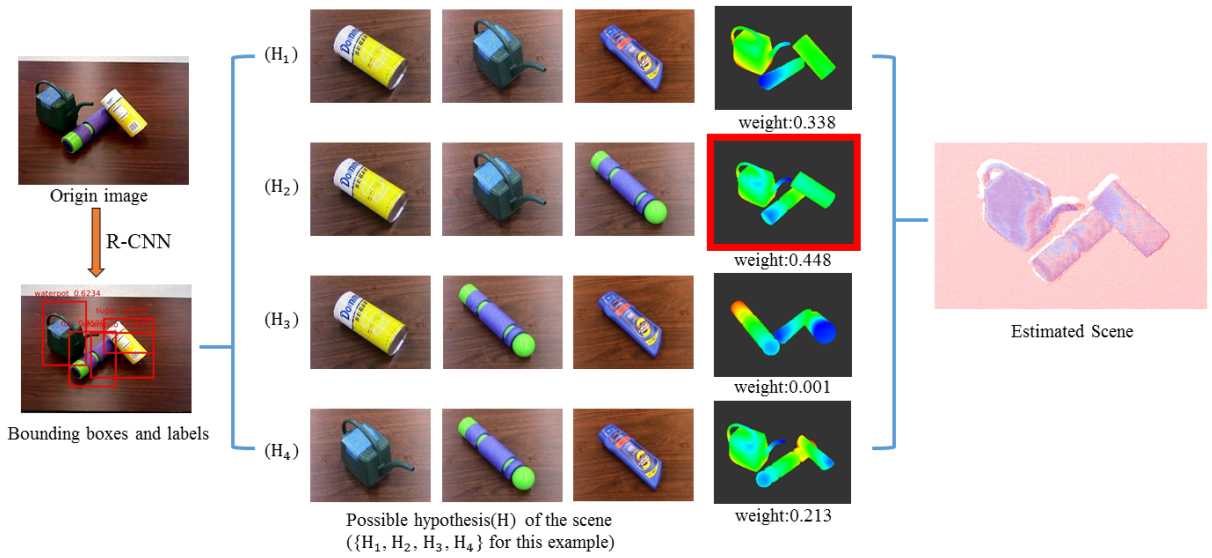

Fig. 3: The proposed *DIGEST* method for cluttered scene estimation. First, the observed RGB image is passed through a R-CNN object detector trained on our grocery object dataset. The R-CNN object detector outputs a set of bounding boxes, with associated object label and detection confidence. Knowing the number of object present in the scene, possible scene hypotheses are enumerated, e.g., ${}^4C_3 = 4$ scene hypotheses are generated in this example. For each each scene hypothesis, particle filtering is applied to converge onto object poses that best explains the observed depth image. After convergence, *DIGEST* outputs the estimated object poses for the most likely scene hypothesis.

$$x_t^{(j)} \sim \sum_j w_{t-1}^{(i)} p(x_t | x_{t-1}^{(i)}, u_{t-1}) \tag{4}$$

as described by [8]. Inference is then performed by computing the likelihood of each particle, normalizing the weights to one, and drawing N particles by importance sampling iteratively.

We render a depth image based on the object poses in each particle $x_t^{(j)}$, represented as $\hat{z}_t^{(j)}$. The rendered depth images are projected back into 3D as a point cloud $\hat{r}_t^{(j)}$ in the camera frame, given intrinsic parameters of the camera. The likelihood for each particle is evaluated as

$$p(z_t | x_t^j) = e^{-\lambda_r \cdot \mathrm{d}(z, \hat{r}_t^{(j)})} \tag{5}$$

where λ_r is a constant scaling factor and $\mathrm{d}(R,O)$ is the sum of the Euclidean distance between the points in $\hat{r}_t^{(j)}$ and the observation point cloud O,

$$\mathrm{d}(R,O) = \sum_i \|(R(i) - O(i))\| \tag{6}$$

After maximum particle filter iterations, we use the most likely particle as the scene estimate for scene hypothesis H_i:

$$x_t = \arg\max_{x_t^{(j)}} p(x_t^{(j)} | z_{1:t}) \tag{7}$$

3) Final Scene Ranking: After particle filtering for all scene hypotheses, we have a scene estimate x_t for each scene hypothesis. We then rank them based on the likelihood of each x_t as computed earlier. The most likely x_t is taken as the scene estimate and is then used to derive the scene graph.

B. Scene Graph Structure

The objects pose estimation of a cluttered scene can be turned into an axiomatic scene graph. We use following axiomatic assertions: $exist(w^j(q^j, v^j))$ for the assertion that object w^j exists in the scene with pose q^j and geometry v^j; $clear(w^i)$ for the assertion that the top of object w^i is clear and no other objects are stacked on it,; $on(w^i, w^j)$ for the assertion that object w^i is stacked on object w^j; $in(w^i, w^j)$ for the assertion that object w^i is in object w^j. An example of a scene graph is given in Figure 4.

To assert the proximity relations between two objects w^i, w^j, we add a *virtual object* $w^\gamma(q^\gamma, v^\gamma)$ into the scene graph, with v^γ being an arbitrary shape, and q^γ expressed in the frame of object w^i. Then, the proximity relation between w^i, w^j can be encoded by $\{has(w^i, w^\gamma),\ in(w^\gamma, w^j)\}$, where $has(w^i, w^\gamma)$ asserts that w^i has a *virtual object* w^γ attached to its frame. When the parent object w^i is in a new location, the robot can adapt to the new scenario by placing the child object w^j within the region of w^γ attached to the frame of w^i.

To determine the stacking relations between the objects, we use simple heuristics. In the 3D object models, the z-axis of each object is the gravitational axis when the object stands upright. The dimensions $\{h_x, h_y, h_z\}$ of the 3D box that encloses each object model are given as prior knowledge. In order to determine whether object w^i is being supported by another object, two heuristics are tested: (1) if one of the object axes (e.g., x-axis) is aligned with the gravitational axis, then the height h_i of the 3D volume occupied by the object equals to the corresponding dimension (e.g. h_x) of the provided 3D enclosing box. A simple rule $z^i - h^{table} > 0.5h^i$ is used to determine whether object w_i is being supported by another object; (2) if none of the object axes are aligned with

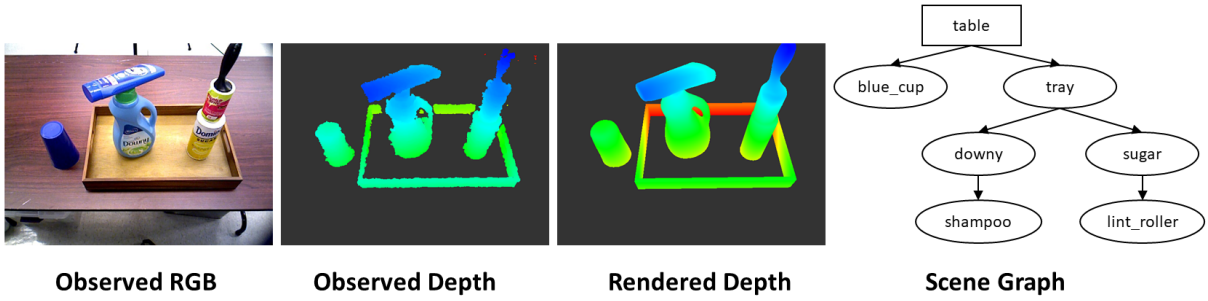

Fig. 4: An axiomatic scene graph example. In the scene graph derived from the estimated object poses, each node corresponds to an object, and each edge indicates the supporting relation between objects. *table* is by default the root node.

the gravitational axis, then w_i is being supported by another object.

The set of objects that is being supported by other objects is sorted with increasing z values of the object pose, and is denoted as O_s, the remaining objects are denoted as O_r. For each object $w^i \in O_s$, a heuristic measure is used to determine which object $w^j \in O_r$ is supporting w^j,

$$\arg\max_j m(r_b(w^i), r_t(w^j))$$

where $m(r_1, r_2)$ measures the overlapping area of two regions r_1, r_2, and $r_t(w), r_b(w)$ represent the projected region on the table of the top and bottom surface of object w, respectively. Once the supporting object for $w^i \in O_s$ is identified, w^i is moved from set O_s to O_r. With the supporting relation between a pair of objects w^i, w^j identified, the corresponding axiomatic assertion is expressed as either $on(w^i, w^j)$ or $in(w^i, w^j)$, depending on the geometry type of the supporting object w^j being convex or concave.

V. Implementation

A. RCNN object detector

We employ R-CNN [14] as our discriminative object detector for *DIGEST*. R-CNN first generates object bounding boxes given an image, then for each bounding box, it outputs the confidence measure through a deep convolutional neural network. For the sake of efficiency and performance, we replace the original selective search [37] with EdgeBox [40] for object proposal generation. We train an R-CNN object detector on our object dataset that includes 15 grocery objects. The dataset contains 8366 ground truth images (~557 average ground truth images for one object) and 60563 background images. We fine tuned our object detector on a pre-trained model on ImageNet [9].

B. Particle Filtering and parallelization

The implementation of the bootstrap filtering pose estimation method consists of three modules: *measurement*, *resampling* and *diffusion*. Each object in each particle $x_t^{(j)}$ is initialized by candidate C_i in the scene hypothesis, the object label l_i determines which 3D object model to use, and the initial pose is uniformly sampled inside the bounding box B_i. A parallel graphics engine rapidly renders depth images given all particles. CUDA is used to compute the likehood of all particles in parallel. Through our experiment, we fix particle filter iteration to 400 and use 625 particles.

In the particle filtering process, the pose of each object is estimated sequentially. For example, if there are four hypothesized objects and 400 iterations for particle filtering, the pose of the object with the maximum detection confidence is estimated in the first 100 iterations. Then the pose of the object with the second largest detection confidence is estimated in the next 100 iterations, with the first object fixed at the most likely pose. We iterate the same estimation process for the remaining objects.

C. Plan and Execution

Given the observation of the goal state of the world, the robot estimates the goal scene graph, that is, the object poses and desired inter-object relations, and stores the goal scene graph by PDDL [24]. Similarly, the robot estimates and stores the initial scene graph by PDDL. With sets of PDDL that describe the initial and goal state, the robot uses a task planner to plan a series of goal-directed actions to rearrange objects in the initial scene, such that the same inter-object relations in the goal scene graph are satisfied. We use breadth first search STRIPS[11] as our task planner.

To execute the planned actions, the robot uses Moveit! [32] to generate collision-free arm trajectories. In our experiment, the object manipulation actions are essentially a sequence of pick-and-place actions. Affordance templates [16] can be incorporated to extend our pipeline to deal with more complex manipulation behavior.

VI. Experiments

In our experiments, we first evaluate our scene estimation method on a public household occlusion dataset and our cluttered scene dataset, and then evaluate our overall semantic robot programming paradigm in tray setting tasks. *DIGEST* outperforms the state of the art method D2P on the household occlusion dataset, and outperforms FPFH on our cluttered scene dataset. We demonstrate the effectiveness of our system for programming a robot to complete various tray-setting tasks through goal-directed manipulations. We run all experiments on a Ubuntu 14.04 system with an Titan X Graphics card and CUDA 7.5.

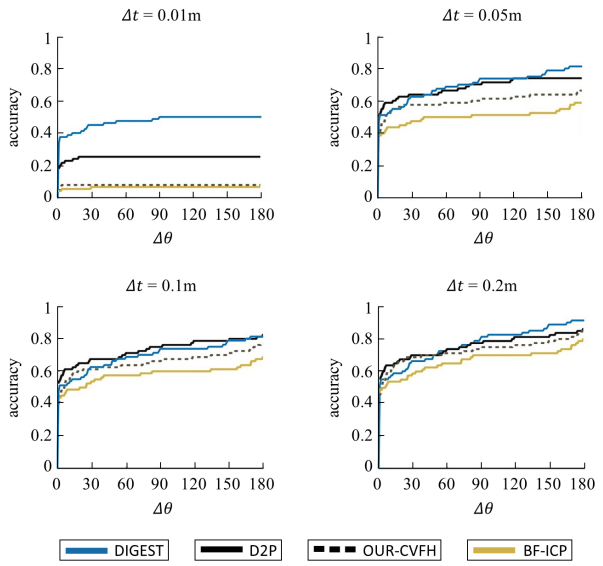

Fig. 5: Object pose estimation benchmark of *DIGEST* on public household object dataset [2], compared with three baseline methods: D2P, OUR-CVFH and BF-ICP for different correctness criteria Δt, $\Delta\theta$. *DIGEST* outperforms D2P for strict correctness criteria, and performs on par with D2P for relaxed correctness criteria.

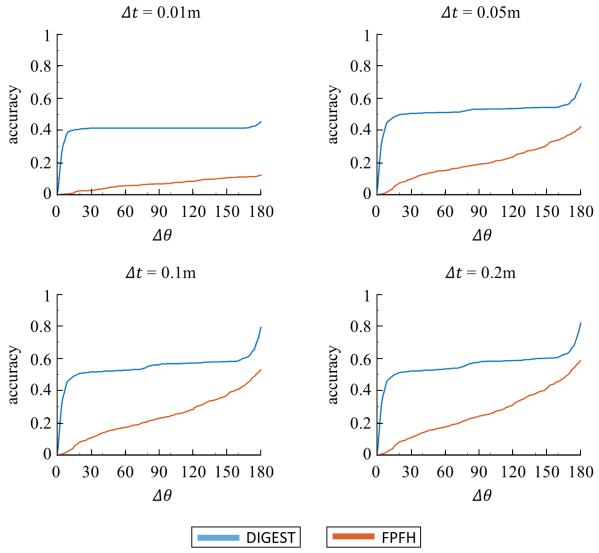

Fig. 6: Object pose estimation benchmark of *DIGEST* on our cluttered scene dataset, compared with baseline method FPFH under different correctness criteria Δt, $\Delta\theta$. *DIGEST* outperforms FPFH with large margin.

A. *DIGEST: Cluttered Scene Estimation*

To evaluate *DIGEST* on object pose estimation, we benchmarked the performance of *DIGEST* on two different datasets: household occlusion dataset [2], and cluttered scene dataset[1] collected by us. The household occlusion dataset contains objects standing up right, thus this dataset only affords benchmarking on 3 DOF object pose estimation. In our cluttered scene dataset, objects can be in arbitrary pose, thus we use this dataset for benchmarking on 6 DOF object pose estimation.

Object pose estimation accuracy is calculated as the percentage of correctly localized objects over the total number of objects in the dataset. An object is correctly localized if the pose error falls within certain position error threshold Δt and rotation error threshold $\Delta\theta$. The position error is the Euclidean distance between the estimated and ground truth object position; the orientation error is the absolute angle error between the estimated and ground truth object orientation. For rotationally symmetric objects, the rotation error w.r.t the symmetric axis is ignored.

1) Household Occlusion Dataset – 3 DOF Object Poses: The household occlusion dataset contains 22 test scenes with 80 objects in total. We compare *DIGEST* against three baseline methods as described in [27], that is, D2P, OUR-CVFH [3], and Brute Force ICP (BF-ICP). D2P also uses an R-CNN object detector as part of their pose estimation process, but it is not clear what hyper parameters they choose during the training phase of the object detector. In order to avoid bias in the training of the object detector, we use their object detector on the household occlusion dataset.

When only little error is allowed for an estimated pose to be counted as correct, as shown in the left upper plot in Figure 5, the accuracy of *DIGEST* is nearly twice the accuracy of D2P. As we relax the tolerance on the pose estimation error, as shown in the other three plots in Figure 5, *DIGEST* performs on par with D2P. Overall, *DIGEST* outperforms D2P since (1) *DIGEST* explores the state space a lot more than D2P, as we do not discretize the state space, and (2) *DIGEST* does not use ICP for local search, which D2P employs for their pose estimation step.

In terms of run time, *DIGEST* takes around 30 seconds (varying with the number of objects and the size of object geometries), which is faster than the 139.74 seconds reported in D2P.

2) Cluttered Scene Dataset – 6 DOF Object Poses: We collect a cluttered scene dataset with 16 different sceness, and 72 objects in total. The number of objects in each scene ranges from 3 to 7. This dataset is much more challenging than the household object dataset, as the objects can have random 6 DOF poses. We compare the performance of *DIGEST* with FPFH [29], as shown in Figure 6.

B. *Semantic Robot Programming: Tray Setting*

We design our experiments around a service robot scenario, as illustrated in Figure 1. The robot needs to prepare a tray as specified by the user int the goal scene. We tested our system on scenes of four to six objects including the tray, with different inter-object spatial relations, such as objects stacked and objects placed next to each other. The robot is able to perceive the initial and goal state of the world as scene graphs, then plan and execute goal-directed actions to satisfy the inter-object relations in the goal scene graph.

As shown in Figure 7, the goal and start scenes are well estimated as a collection of 6DOF poses of objects. Based on the scene graph inferred from the object pose estimates, the robot generates a sequence of goal-directed actions to transit

[1] https://www.dropbox.com/sh/5ub39urzxsf87m1/AADAtsjM138tKbbyBxCoxmVla?dl=0

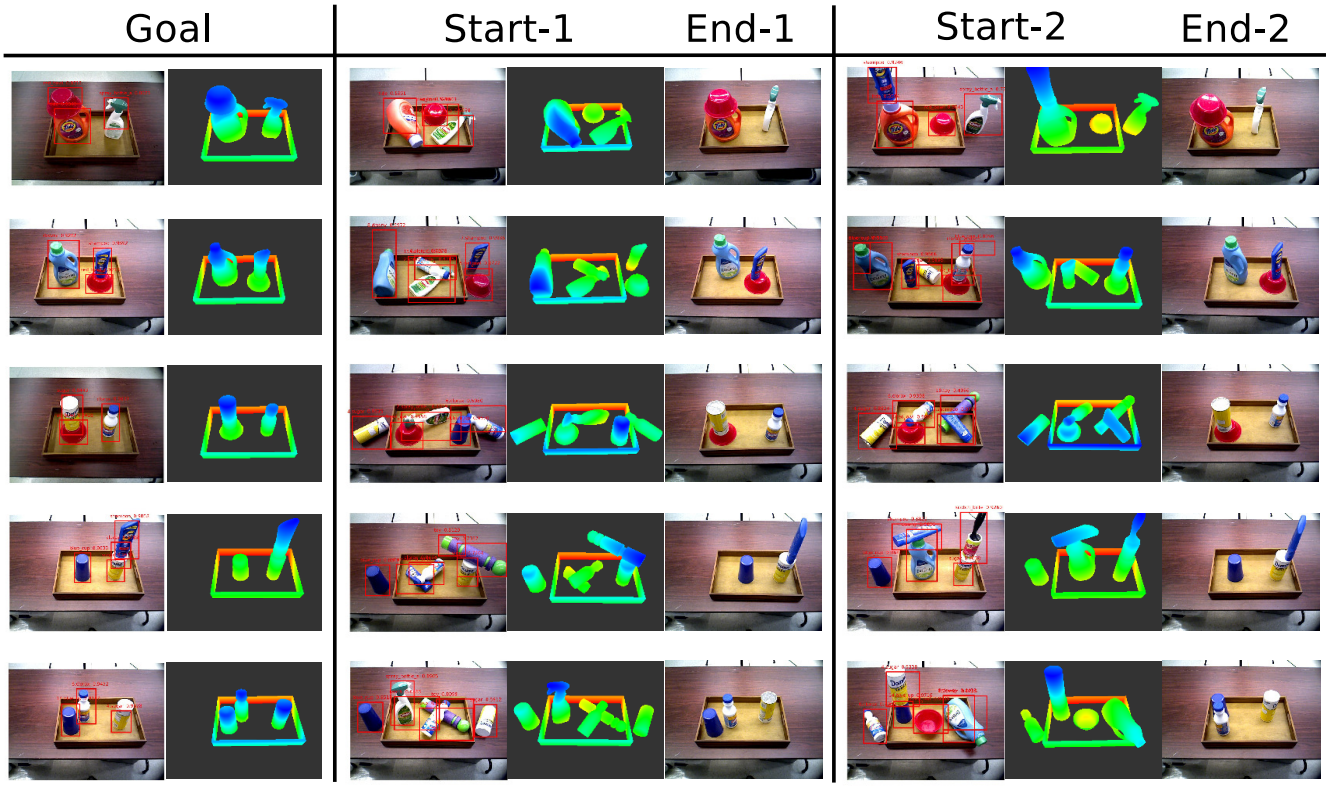

Fig. 7: We tested our goal-directed robot programming paradigm for 5 different tray preparation tasks. The left column shows how the user desires the tray to be prepared. For each goal, the robot starts from two different initial states and successfully performs goal-directed manipulations to prepare the tray. Depth images are rendered based on 6 DOF object poses output by *DIGEST*.

the state of the world from the initial to the goal state. The robot successfully sets up a tray as the user desired in 10 out of 10 different tray setting scenarios. An example of the robot manipulation action sequence is shown in Figure 8.

VII. Discussion

As the number of objects in the scene increases, MCMC sampling methods can be incorporated into our paradigm to search for scene hypotheses more efficiently. For grasp actions as part of the goal-directed manipulation, we tested using the existing grasp point localization method [35] by integrating it into our system. Although this method helps select good grasp poses given the point cloud of an object, the selected grasp pose is usually not appropriate for a later placement action. Research on grasp planning for a later sequence of actions is beyond the scope of this paper. Instead, in our experiments, appropriate grasp poses for later placement actions are selected by the provided affordance associated with the object. In the future, we would like to investigate grasp planning for later sequence of actions. To monitor the manipulation status and make the system more robust to scene graph estimation errors, we would also like to explore real-time object and robot-arm tracking during manipulation actions.

VIII. Conclusion

We provide a semantic robot programming paradigm for users to easily program the goal of a task for a robot, so that the robot can autonomously plan actions to reach the goal from the initial state of the world. The state of world is estimated and represented as a scene graph. Our system is evaluated in a common service robot scenario—tray-setting tasks. We demonstrate the effectiveness of the proposed *DIGEST* method on both house occlusion dataset and our cluttered scene dataset. By combining a discriminative object detector and a generative object pose estimation method, our system is able to estimate 6 DOF object poses and a scene graph, given only the number of objects in the scene. In the future, we will incorporate affordance templates into the system for complex manipulation actions beyond pick and place.

References

[1] B. Akgun, M. Cakmak, K. Jiang, and A. L. Thomaz. Keyframe-based learning from demonstration. *International Journal of Social Robotics*, 4(4):343–355, 2012.

[2] A. Aldoma, Z.-C. Marton, F. Tombari, W. Wohlkinger, C. Potthast, B. Zeisl, R. B. Rusu, S. Gedikli, and M. Vincze. Point cloud library. *IEEE Robotics & Automation Magazine*, 1070(9932/12), 2012.

[3] A. Aldoma, F. Tombari, R. B. Rusu, and M. Vincze. Our-cvfh–oriented, unique and repeatable clustered viewpoint feature histogram for object recognition and 6dof pose estimation. In *Joint DAGM (German Association for Pattern Recognition) and OAGM Symposium*, pages 113–122. Springer, 2012.

[4] D. J. Cannon. Point-and-direct telerobotics: Object level strategic supervisory control in unstructured interactive human-machine system environments. 1992.

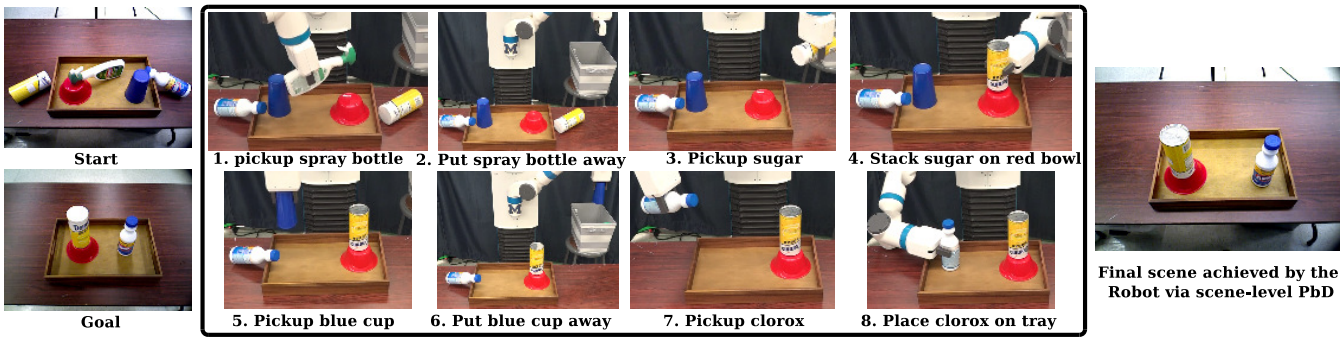

Fig. 8: A robot performing goal-directed manipulations to prepare the tray as the user desires.

[5] C. Chao, M. Cakmak, and A. L. Thomaz. Towards grounding concepts for transfer in goal learning from demonstration. In *2011 IEEE International Conference on Development and Learning (ICDL)*, volume 2, pages 1–6. IEEE, 2011.

[6] S. Chernova and M. Veloso. Interactive policy learning through confidence-based autonomy. *Journal of Artificial Intelligence Research*, 34(1):1, 2009.

[7] M. Ciocarlie, K. Hsiao, E. G. Jones, S. Chitta, R. B. Rusu, and I. A. Şucan. Towards reliable grasping and manipulation in household environments. In *Experimental Robotics*, pages 241–252. Springer, 2014.

[8] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte carlo localization for mobile robots. In *IEEE International Conference on Robotics and Automation (ICRA 1999)*, May 1999.

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[10] S. Ekvall and D. Kragic. Robot learning from demonstration: a task-level planning approach. *International Journal of Advanced Robotic Systems*, 5(3):33, 2008.

[11] R. E. Fikes and N. J. Nilsson. Strips: A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, 2(3-4):189–208, 1971.

[12] R. E. Fikes and N. J. Nilsson. Strips: A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, 2(3):189–208, 1972.

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition*, 2014.

[15] D. H. Grollman and O. C. Jenkins. Incremental learning of subtasks from unsegmented demonstration. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 261–266. IEEE, 2010.

[16] S. Hart, P. Dinh, and K. Hambuchen. The affordance template ROS package for robot task programming. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6227–6234. IEEE, 2015.

[17] E. Herbst, P. Henry, and D. Fox. Toward online 3-d object segmentation and mapping. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 3193–3200. IEEE, 2014.

[18] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on pattern analysis and machine intelligence*, 21(5):433–449, 1999.

[19] C. C. Kemp, C. D. Anderson, H. Nguyen, A. J. Trevor, and Z. Xu. A point-and-click interface for the real world: laser designation of objects for mobile manipulation. In *Human-Robot Interaction (HRI), 2008 3rd ACM/IEEE International Conference on*, pages 241–248. IEEE, 2008.

[20] B. Kuipers. The spatial semantic hierarchy. *Artificial intelligence*, 119(1-2):191–233, 2000.

[21] J. E. Laird, A. Newell, and P. S. Rosenbloom. Soar: An architecture for general intelligence. *Artificial intelligence*, 33, 1987.

[22] I. Lenz, H. Lee, and A. Saxena. Deep learning for detecting robotic grasps. *The International Journal of Robotics Research*, 34(4-5):705–724, 2015.

[23] Z. Liu, D. Chen, K. M. Wurm, and G. von Wichert. Table-top scene analysis using knowledge-supervised mcmc. *Robotics and Computer-Integrated Manufacturing*, 33:110–123, 2015.

[24] D. McDermott, M. Ghallab, A. Howe, C. Knoblock, A. Ram, M. Veloso, D. Weld, and D. Wilkins. PDDL-the planning domain definition language. 1998.

[25] S. J. McKenna and H. Nait-Charif. Tracking human motion using auxiliary particle filters and iterated likelihood weighting. *Image and Vision Computing*, 25(6):852–862, 2007.

[26] J. Nakanishi, J. Morimoto, G. Endo, G. Cheng, S. Schaal, and M. Kawato. Learning from demonstration and adaptation of biped locomotion. *Robotics and Autonomous Systems*, 47(2):79–91, 2004.

[27] V. Narayanan and M. Likhachev. Discriminatively-guided deliberative perception for pose estimation of multiple 3d object instances. In *Proceedings of Robotics: Science and Systems*, AnnArbor, Michigan, June 2016.

[28] 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*, pages 91–99, 2015.

[29] R. B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (fpfh) for 3d registration. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 3212–3217. IEEE, 2009.

[30] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu. Fast 3d recognition and pose using the viewpoint feature histogram. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 2155–2162. IEEE, 2010.

[31] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz. Towards 3d point cloud based object maps for household environments. *Robotics and Autonomous Systems*, 56(11):927–941, 2008.

[32] I. A. Sucan and S. Chitta. Moveit! *Online Available: http://moveit. ros. org*, 2013.

[33] I. A. Şucan, M. Moll, and L. E. Kavraki. The Open Motion Planning Library. *IEEE Robotics & Automation Magazine*, 19(4):72–82, December 2012. `http://ompl.kavrakilab.org`.

[34] Z. Sui, L. Xiang, O. C. Jenkins, and K. Desingh. Goal-directed robot manipulation through axiomatic scene estimation. *The International Journal of Robotics Research*, 36(1):86–104, 2017.

[35] A. ten Pas and R. Platt. Using geometry to detect grasp poses in 3d point clouds. In *Intl Symp. on Robotics Research*, 2015.

[36] A. Ten Pas and R. Platt. Localizing handle-like grasp affordances in 3d point clouds. In *Experimental Robotics*, pages 623–638. Springer, 2016.

[37] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[38] H. Veeraraghavan and M. Veloso. Teaching sequential tasks with repetition through demonstration. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 3*, pages 1357–1360. International Foundation for Autonomous Agents and Multiagent Systems, 2008.

[39] Y. Yang, Y. Li, C. Fermüller, and Y. Aloimonos. Robot learning manipulation action plans by" watching" unconstrained videos from the world wide web. In *AAAI*, pages 3686–3693, 2015.

[40] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *European Conference on Computer Vision*, pages 391–405. Springer, 2014.