Semantic Robot Programming for Goal-Directed Manipulation in Cluttered Scenes

Zhen Zeng Zheming Zhou Zhiqiang Sui Odest Chadwicke Jenkins

***Abstract*— We present the Semantic Robot Programming (*SRP*) paradigm as a convergence of robot programming by demonstration and semantic mapping. In *SRP*, a user can directly program a robot manipulator by demonstrating a snapshot of their intended goal scene in workspace. The robot then parses this goal as a scene graph comprised of object poses and inter-object relations, assuming known object geometries. Task and motion planning is then used to realize the user's goal from an arbitrary initial scene configuration. Even when faced with different initial scene configurations, *SRP* enables the robot to seamlessly adapt to reach the user's demonstrated goal. For scene perception, we propose the Discriminatively-Informed Generative Estimation of Scenes and Transforms (*DIGEST*) method to infer the initial and goal states of the world from RGBD images. The efficacy of *SRP* with *DIGEST* perception is demonstrated for the task of tray-setting with a Michigan Progress Fetch robot. Scene perception and task execution are evaluated with a public household occlusion dataset and our cluttered scene dataset.**

I. Introduction

Many service robot scenarios, such as setting up a dinner table or organizing a shelf, require a computational representation of a user's desired world state. For example, how is the dinner table to be set, or how is the shelf to be organized. More specifically, what are the objects involved in the task, what are the desired poses of those objects, and what are the important spatial relationships between objects. Towards natural and intuitive modes of human-robot communication, we present the **Semantic Robot Programming (*SRP*)** paradigm for declarative robot programming over user demonstrated scenes. In *SRP*, we assume a robot is capable of goal-directed manipulation [35] for realizing an arbitrary scene state in the world. A user can program such goal-directed robots by demonstrating their desired goal scene. *SRP* assumes such scenes can be perceived from partial RGBD observations, which has proven a challenging problem in itself.

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] from appearance has improved greatly in filtering of background noise and focused attention to objects of interest. However, the applicability of such vision-based methods robot perception remains unclear, especially for the purposes of goal-directed manipulation. This circumstance has given rise to new approaches to semantic mapping [20], [31], [17] to computationally model a robot's environment into perceivable objects with robot-actionable affordances.

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 [zengzhen|zhezhou|zsui|ocj]@umich.edu

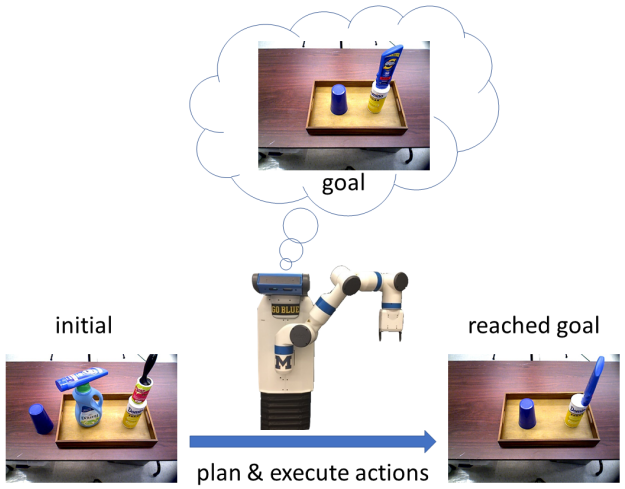

Fig. 1: A robot preparing a tray through goal-directed manipulations. Given the observation of the user desired goal state and the initial state of the tabletop workspace, the robot first perceives the axiomatic scene graph of the goal and initial state, and then plan and execute goal-directed actions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

We posit semantic mapping offers a springboard to new forms of robot programming, such as 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 [37] and motion planning [33]. We have previously proposed methods for scene estimation [34], [35] from robot RGBD sensing that used scene graphs expressed axiomatically as a semantic mapping abstraction. This abstraction allowed 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]. However, the computational cost of inference over scenes is asymptotically intractable as the number of objects grows.

In this paper, we propose the paradigm of Semantic Robot Programming for robot manipulators with a complementary method for more tractable scene perception. *SRP* is a declarative approach to programming robots through demonstration, where users only need to demonstrate their desired state

of the world. *SRP* is general across methods of perception, given the perceived scene is represented axiomatically. For scene perception, we present the Discriminatively-Informed Generative Estimation of Scenes and Transforms (*DIGEST*) method to infer the initial and goal scene states for *SRP* from RGBD images. *DIGEST* brings together discriminative object detection and generative pose estimation for inference of 6 DOF object poses in cluttered scenes, assuming the number of objects is known. Given perceived initial and goal scenes, the robot can plan and execute goal-directed manipulation to autonomously transit the world from the initial to the goal state.

We evaluate the *SRP* paradigm in tray-setting task scenario with the Michigan Progress Fetch robot (Figure 1). We benchmark the performance of *DIGEST* on a household occlusion dataset [2] and our cluttered scene dataset. We demonstrate that *SRP* is effective in understanding the goal of a task given a demonstrated 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 additionally found *DIGEST* performs favorably in comparison with state-of-the-art methods for scene perception, such as D2P [27], with fewer assumptions of prior knowledge.

II. Related Work

SRP builds on much existing work in robot Programming by Demonstration (PbD) and scene perception for manipulation. Similar to robot PbD, *SRP* aims to enable users to effectively communicate their objectives to robots for performing manipulation tasks. We posit advances in scene perception for manipulation offers new avenues for extending the ease and intuitiveness of robot PbD.

A. Programming by Demonstration

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. [39] 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. [40] have proposed learning action plans in real world scenario, similar to our robot programming paradigm that works with real world objects.

B. Scene Perception for Manipulation

Being able to perceive objects in real world scenarios and act on them remains a challenge. Some works are able to extract grasping point [7], [22], [36] 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, work 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) [35] 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, thus both methods can only estimate 3 DOF poses of objects (i.e., x, y, θ). However, these assumptions are too strong in real world scenarios. Instead, our scene estimation method *DIGEST* does not rely on any of these assumptions, and it can estimate 6 DOF poses of objects, as long as the number of objects in the scene is known.

III. Problem Statement

SRP with *DIGEST* assumes the number of objects N_c present in the scene, 3D mesh models $\mathbf{M} = \{m_1, \cdots, m_l\}$ for a set of objects. The robot is assumed 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 $t+T$. The objective of *SRP* is to 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

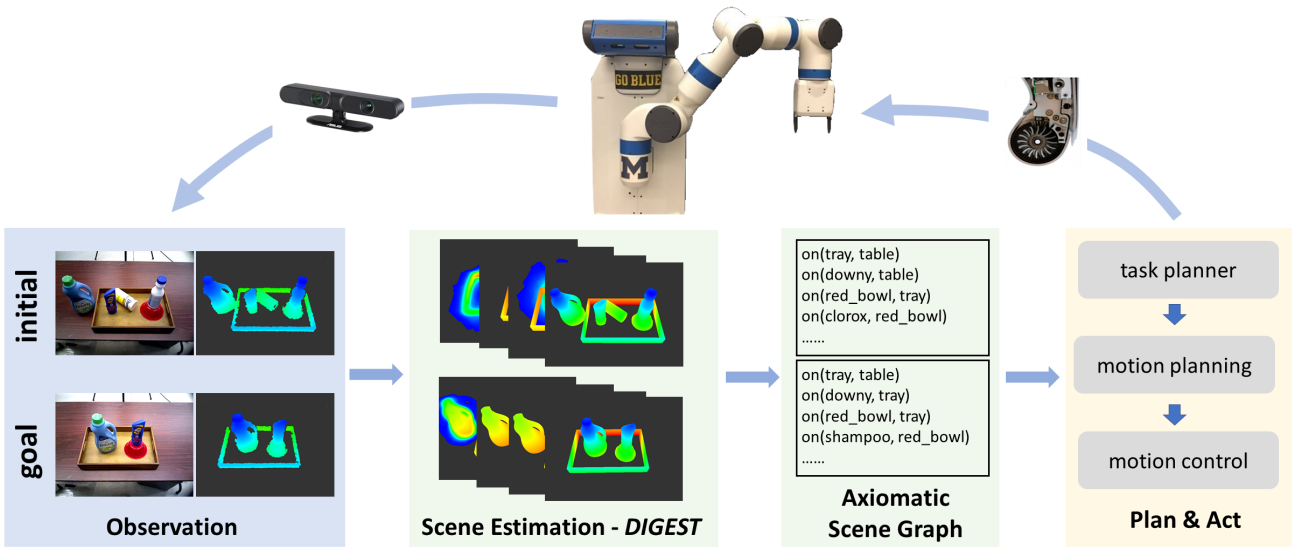

Fig. 2: Our goal-directed robot programming has three stages: 1) Given the RGB-D observation of the goal and initial scene, we use the proposed scene estimation method *DIGEST* to detect object and estimate the 6 DOF pose of objects; 2) Axiomatic scene graphs can be derived from the estimated object 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.

are satisfied; where *DIGEST* infers the goal scene graph s_G and the initial scene graph s_I, respectively.

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 $s_t = \{v^i(x_t)\}_{i=0}^{K}$, where $v^i \in \{exist, clear, on, in\}$ is an axiomatic assertion parameterized by $x_t = \{q_t^j\}_{j=0}^{N_c}$, with q_t^j denoting the pose of jth object at time t, N_c being the number of objects, and K being the total number of axiomatic assertions. In our work, the assertions are limited to spatial relations that can be tested geometrically. 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

The *SRP* paradigm consists of the perception of goal and initial scene states, and the planning and execution stages, as shown in Figure 2. Given observations of a cluttered scene, the generative sampling inference process over object poses is informed by detections from a discriminative object detector. A scene graph encoding inter-object relations is geometrically inferred from an estimate of inferred object poses. The resulting scene graph is then expressed axiomatically for use in task planning and execution.

A. DIGEST Cluttered Scene Estimation

Given observed RGB-D image pair 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, composed of a set of real-valued object poses. Object poses are assumed to be statistically independent. 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: a snapshot of the scene instead of a history of observations. Thus, we apply Iterated Likelihood Weighting [25] to bootstrap the scene estimation

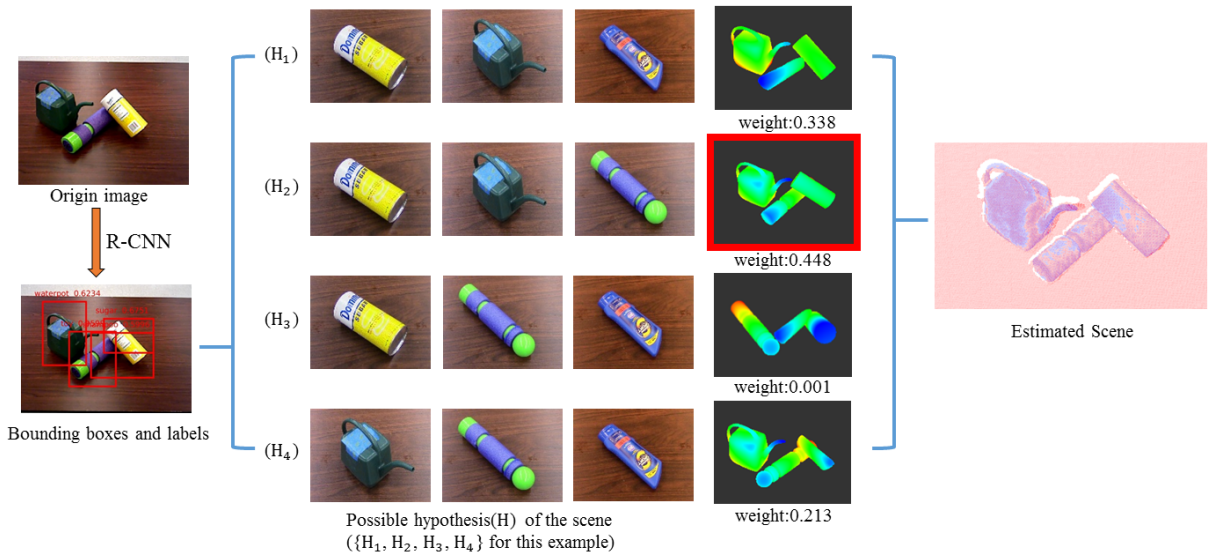

Fig. 3: The proposed *DIGEST* method for cluttered scene estimation. First, the observed RGB image is passed through a R-CNN object detector trained on our grocery object dataset. The R-CNN object detector outputs a set of bounding boxes, with associated object label and detection confidence. Knowing 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 scene hypothesis, particle filtering is applied to estimate object poses that best explains the observed depth. After convergence, *DIGEST* outputs the estimated object poses for the most likely scene hypothesis.

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

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

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

as described by [8]. To evaluate the weight $w_t^{(j)}$ for particle $x_t^{(j)}$, we render a depth image based on the object poses in $x_t^{(j)}$, and compare it against the observed depth image $\hat{z}_t^{(j)}$,

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

where λ_r is a constant scaling factor. $\mathrm{d}(z, \hat{r}_t^{(j)})$ is the sum of the Euclidean distance between the 3D points projected back from depth images z, $\hat{r}_t^{(j)}$, using the intrinsic parameters of the camera. Pose estimation is performed over successive iterations that: 1) compute the weight of each particle, 2) normalize the weights to one, 3) draw N particles by importance sampling, and 4) diffuse each sampled particle by a zero-mean Gaussian noise. After maximum number of iterations, the most likely particle as the scene estimate for scene hypothesis H_i:

$$x_t = \arg\max_{x_t^{(j)}} p(x_t^{(j)}|z_{1:t}) \tag{6}$$

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: $exist(q^j)$ for the assertion that object j exists in the scene with pose q^j; $clear(q^i)$ for the assertion that the top of object i is clear and no other objects are stacked on it; $on(q^i, q^j)$ for the assertion that object i is stacked on object j; $in(q^i, q^j)$ for the assertion that object i is in object j. An example of a scene graph is given in Figure 4.

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

To determine the stacking relations between the objects, we use simple heuristics. In the 3D mesh 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 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 i is being supported by another object; (2) if none of the object axes are aligned

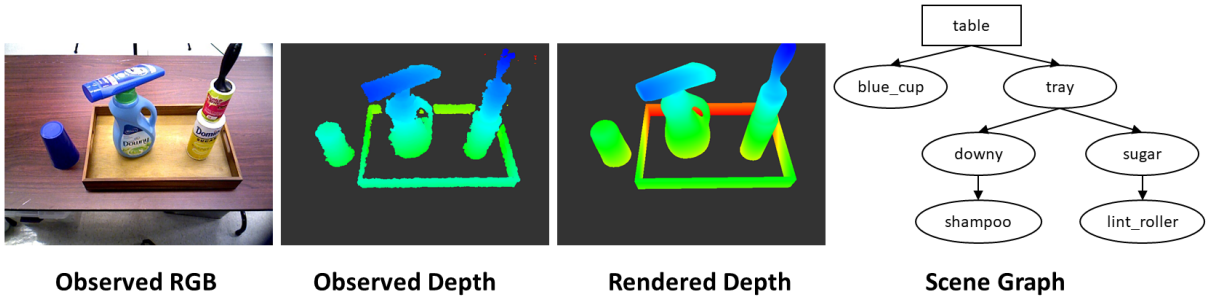

Fig. 4: An axiomatic scene graph example. In the scene graph derived from the estimated object poses, each node corresponds to an object, and each edge indicates the supporting relation between objects. *table* is by default the root node.

with the gravitational axis, then object 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 $i \in O_{\mathrm{s}}$, a heuristic measure is used to determine which object $j \in O_{\mathrm{r}}$ is supporting i,

$$\arg\max_j f(r_b(q^i), r_t(q^j))$$

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

V. Implementation

A. RCNN object detector

We employ R-CNN [14] as our discriminative object detector as described in section IV-A.1. 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 [38] with EdgeBox [41] 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

During bootstrap filtering for pose estimation as described in section IV-A.2. 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 mesh 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 weights 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 particle filter iterations, 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 2nd largest detection confidence is estimated in the next 100 iterations, with the first object fixed at the most likely pose. We carry on the estimation process iteratively for the remaining objects.

C. Planning and Execution

Given the observation of the goal state of the world, the robot estimates the goal scene graph, and stores the desired inter-object relations by PDDL [24]. Similarly, the robot estimates and stores the initial inter-object relations 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. Note that the robot does not need to rearrange the objects with the exact same poses as in the goal scene, as long as the same inter-object relations are achieved, similarly to how human would arrange a set of daily objects based on simple instructions.

The task planner gives a sequence of high-level pick-and-place actions. To pick an object, the robot is given a set of pre-computed grasp poses of the object using [36], and uses Moveit! [32] to check which grasp pose it can generate a collision-free trajectory for, and use that for grasping. To place an object, the robot sample place poses in the empty space that satisfies the desired inter-object relations, and again use the place pose it can generate a collision-free trajectory for.

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

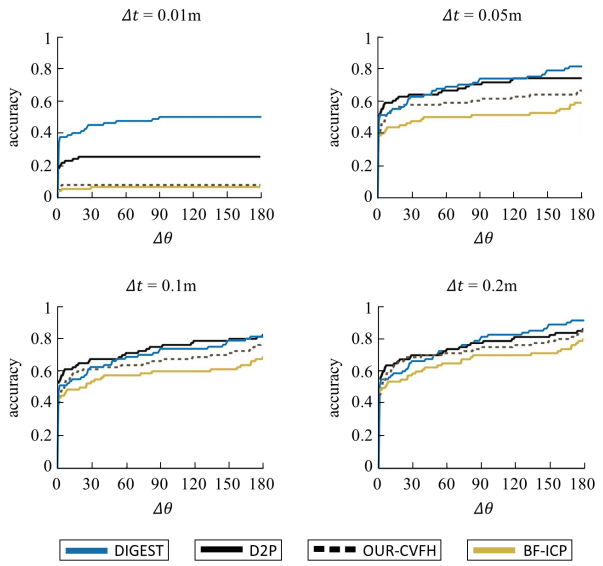

Fig. 5: Object pose estimation benchmark of *DIGEST* on public household object dataset [2], compared with three baseline methods: D2P, OUR-CVFH and BF-ICP for different correctness criteria Δt, $\Delta\theta$. *DIGEST* outperforms D2P for strict correctness criteria, and performs on par with D2P for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

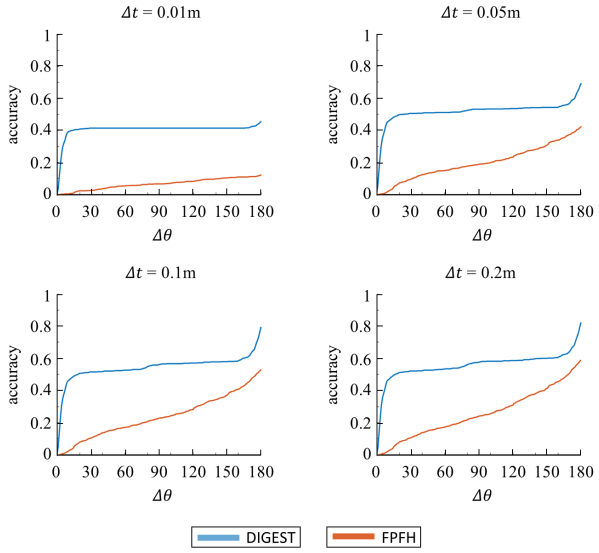

Fig. 6: Object pose estimation benchmark of *DIGEST* on our cluttered scene dataset, compared with baseline method FPFH under different correctness criteria Δt, $\Delta\theta$. *DIGEST* outperforms FPFH with large margin.

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 computer with an Titan X Graphics card and CUDA 7.5.

A. *DIGEST: Cluttered Scene Estimation*

To evaluate *DIGEST* on pose estimation, we benchmarked the performance of *DIGEST* on two different datasets: household occlusion dataset [2], and our cluttered scene dataset. The household occlusion dataset contains objects standing up right, thus it only affords benchmarking on 3 DOF object pose estimation. In our cluttered scene dataset, objects can be in arbitrary pose, and we use it 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 error in translation; the rotation error is the absolute angle error in orientation. For rotationally symmetric objects, the rotation error about 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. The test scenes include objects such as milk bottles, laundry items, mugs and etc; 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 pose estimation. In terms of run time, *DIGEST* takes around 30 seconds (varying with the number of objects and the size of object mesh), which is faster than 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. This dataset includes laundry items, kitchen items and toy with non-trivial geometry. 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 designed 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 4 to 6 objects including the tray, with different inter-object relations, such as stacking and proximity relations. The robot is able to perceive the initial and goal state, then plan and execute goal-directed actions to satisfy the inter-object relations in the goal scene graph.

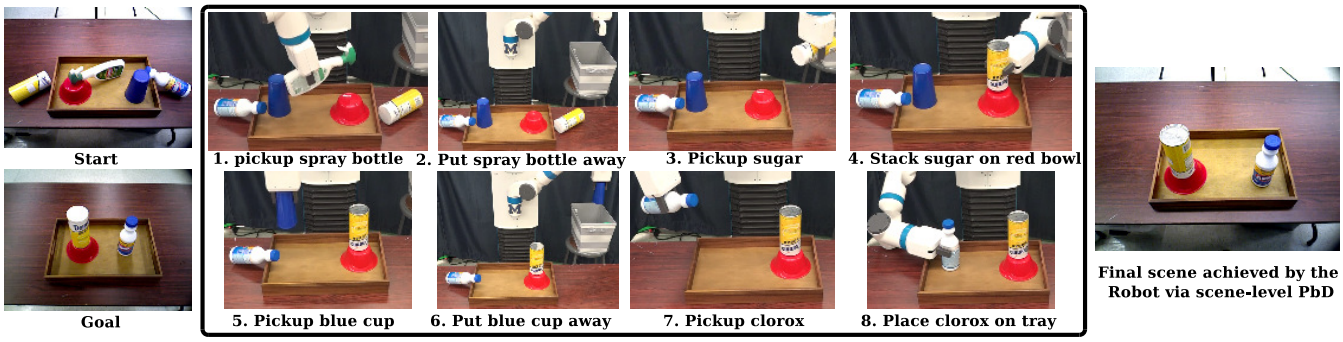

Fig. 7: Our robot performing goal-directed manipulation (middle columns) to prepare a tray (right) satisfying the user-demonstrated goal (left bottom).

An example of *SRP* for goal-directed manipulation is shown in Figure 7. Based on the scene graph inferred from the object pose estimates, the robot generates a sequence of goal-directed actions to achieve the goal state. Our tray setting experiments are shown in Figure 8, and more detail in the accompanying video[1]. The goal and start scenes are well estimated as a collection of 6DOF poses of objects. The robot successfully sets up a tray as the user desired in 10 out of 10 different tray setting experiments.

VII. Conclusion

We have presented Semantic Robot Programming as a paradigm for users to easily program robots in a declarative goal-directed manner. We demonstrate the effectiveness of *SRP* using the proposed *DIGEST* scene perception method on two datasets of objects in occlusion and clutter: both house occlusion dataset and our cluttered scene dataset. Through our approach to generative-discriminative perception, *SRP* with *DIGEST* is able to perceive, reason, and act to realize an arbitrary user-demonstrated goal in cluttered scenes.

There are many future directions to pursue, such as motion planning over sequences of general manipulation actions. Currently, grasp point localization [36] is used to select good grasp poses for object picking. However, such selected grasp poses are not necessarily appropriate for a later placement actions. Visual inspection on selected grasps is done before robot execution. Ideally, appropriate grasp poses would be provided by a manipulation affordance mechanism, such as Affordance Templates [16] associating robot action with an object. Such affordance mechanisms would allow for investigation of more flexible task and motion planning over sequences of actions. We further posit scene perception can be made to run in interactive-time through a thoughtful parallelized implementation, enabling potentially interactive planning and manipulation execution.

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.

[1] `http://progress.eecs.umich.edu/projects/srp`

[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.

[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

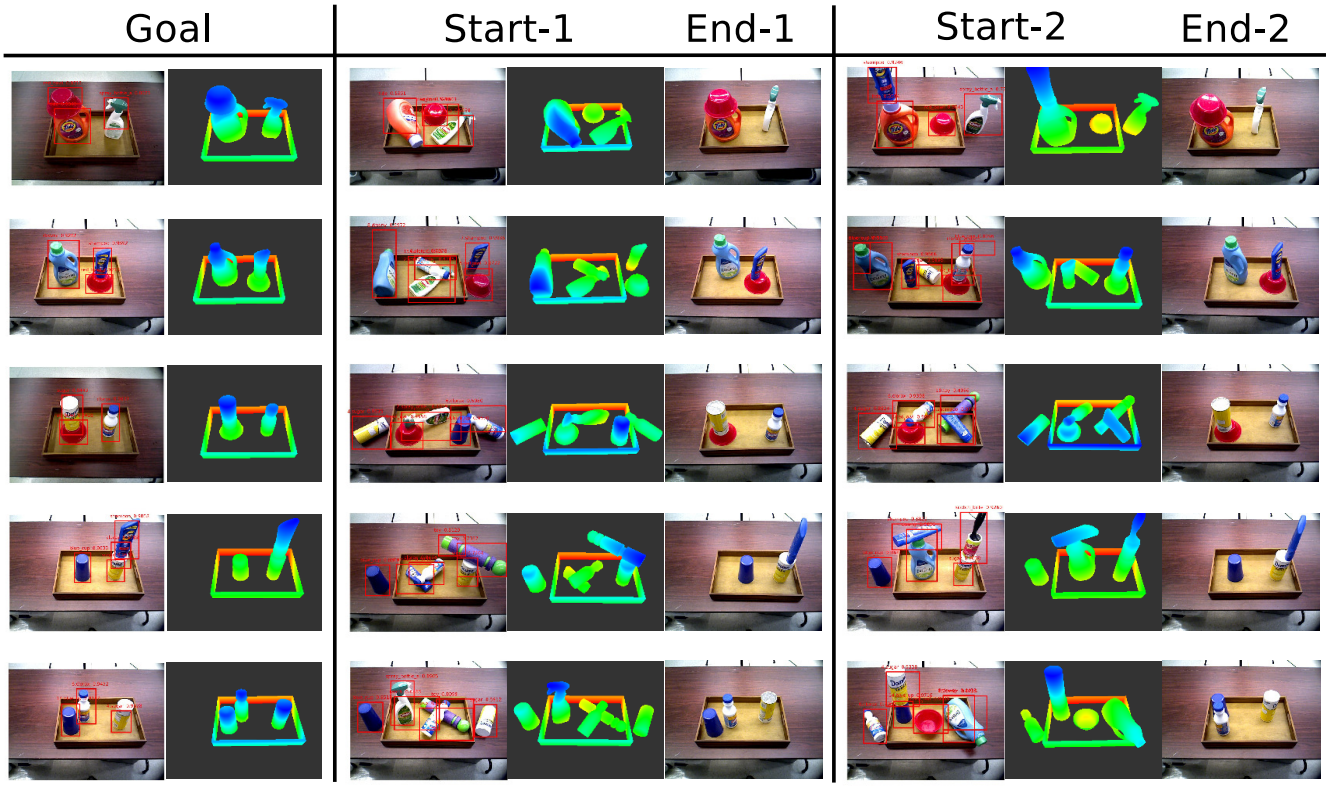

Fig. 8: Our *SRP* system tested for 5 different tray preparation tasks. The left column shows the goal scene. For each goal, the robot starts from two different initial states and successfully performs goal-directed manipulations to prepare the tray. Depth images are rendered based on 6 DOF object poses output by *DIGEST*.

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! 2013. [Online] Available: `http://moveit.ros.org`.

[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, O. C. Jenkins, and K. Desingh. Axiomatic particle filtering for goal-directed robotic manipulation. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 4429–4436. IEEE, 2015.

[35] 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.

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

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

[38] 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.

[39] 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.

[40] 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.

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