SUM: Sequential Scene Understanding and Manipulation

Zhiqiang Sui   Zheming Zhou   Zhen Zeng   Odest Chadwicke Jenkins

Abstract— In order to perform autonomous sequential manipulation tasks, perception in cluttered scenes remains a critical challenge for robots. In this paper, we propose a probabilistic approach for robust sequential scene estimation and manipulation - Sequential Scene Understanding and Manipulation (*SUM*). *SUM* considers uncertainty due to discriminative object detection and recognition in the generative estimation of the most likely object poses maintained over time to achieve a robust estimation of the scene under heavy occlusions and unstructured environment. Our method utilizes candidates from discriminative object detector and recognizer to guide the generative process of sampling scene hypothesis, and each scene hypotheses is evaluated against the observations. Also *SUM* maintains beliefs of scene hypothesis over robot physical actions for better estimation and against noisy detections. We conduct extensive experiments to show that our approach is able to perform robust estimation and manipulation.

I. INTRODUCTION

Perception is a critical capability to enable purposeful goal-directed manipulation for autonomous robots, particularly in cluttered environments. Truly autonomous robot manipulators need to be able to perceive the world, reason over manipulation actions towards the goal, and carry out these actions in terms of physical motion. Closing this loop for autonomous scene-level manipulation is now within reach given the current advances in capable mobile manipulation platforms (such as the Fetch platform shown in Fig. 1 and tractable task and motion planning. Without the ability to perceive in common unstructured environments, however, autonomous manipulation will remain restricted to simulation and highly controlled environments.

Previously, we addressed the problem of perception for goal-directed manipulation as *axiomatic scene estimation* [32], [33], sharing similar aims to existing work in scene estimation for manipulation of rigid objects [22], [23], [19], [17], [6]. These methods take a generative multi-hypothesis approach to robustly inferring a tree-structured scene graph, as object poses and directed inter-object relations, from cluttered scenes observed as 3D point clouds. The inferred scene graph estimate can be expressed as parameterized axioms that allow for interoperable symbolic task and motion planning towards goals expressed as desired scenes in world space. We posit this pattern of symbolic task-level reasoning using estimates from probabilistic perception will be as applicable to scenes for autonomous manipulation as it has been for autonomous navigation.

Z. Sui, Z. Zhou, Z. Zeng and O.C. Jenkins are with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, USA, 48109-2121 [zsui|zhezhou|zengzhen|ocj]@umich.edu

Fig. 1: (Top) a robot using *SUM* for scene perception in the sorting of a cluttered set of objects on a table into cleaning (right bin) and non-cleaning categories (left bin). *SUM* performs perception by using (bottom left) RGB object recognition to inform (bottom right) sequential pose estimation from 3D point cloud observations and the feasible grasp poses on the manipulated object.

Existing formulations of axiomatic scene estimation impose several limiting assumptions that must be relaxed for viable autonomous manipulation. First, existing axiomatic scene estimators assume the identification of objects observed in a scene are given or provided by an idealized object detection and recognition system. Object detection and recognition [15], [12], [11] has greatly improved towards feasible general use, due in part to the renaissance in convolutional neural networks. However, such recognition methods remain subject to substantial and inherent errors in discriminating false positive and negative detections. As such, our robots will need to handle uncertainty due to such recognition errors in both its scene estimation and task execution.

Second, scenes have been assumed to be static, where scene state at each moment in time is effectively decoupled sequentially from its past state. This assumption can be prohibitively costly in terms of computation, as the dimensionality of scene state space grows exponentially with the number of scene objects. We posit that robots can manage this complexity by maintaining a belief of scene state over time informed by past beliefs, a manipulation process model, and current object detections, as well as incorporation of

physical and contextual constraints [10].

Lastly, existing scene estimation often assumes some scene structure, such as a canonical object orientation, a large flat surface support, or "stacking" as a single support surface per object. By maintaining belief sequentially and managing computational burden, our robots will be able to perform tractable inference in cluttered scenes with full six degree-of-freedom object poses and an arbitrary number inter-object contacts and supports.

In this paper, we propose Sequential Scene Understanding and Manipulation (*SUM*) as a model for scene perception from RGBD sensing in sequential manipulation tasks. *SUM* considers uncertainty due to discriminative object detection and recognition in the generative estimation of the most likely object poses maintained over time. Towards this end, *SUM* utilizes the output of modern convolutional neural networks [12] for object detection and recognition to guide the generative process of sampling scene hypotheses within a pose estimation process. Pose estimation is modeled as a recursive Bayesian filter [8], [36] to maintain a belief over object poses across a sequence of robot actions. We demonstrate the effectiveness and robustness of *SUM* with respect to both perception and manipulation errors in a cluttered tabletop scenario for an object sorting task with a Fetch mobile manipulator.

II. RELATED WORK

A. Perception for Manipulation

We summarize a relevant subset of existing work with respect to perception for manipulation. The PR2 interactive manipulation pipeline [5] segments objects from a flat tabletop surface through clustering of surface normals. This pipeline can perform relatively robust pick-and-place manipulation for isolated, non-occluded, and non-touching objects. For cluttered scenes, Narayanaswamy et al. [24] perform pose and structure estimation of toy parts for flexible assembly of structures. MOPED [6] is a framework for objects detection and pose estimation using clustering of features from multi-views. Joho et al. [17] use a probabilistic generative model to model the spatial arrangement of objects on a flat surface in the context of a table setting task.

Narayanan et al. [22], [23] are similar to our work on estimation where they integrate exhaustive global reasoning with discriminatively-trained algorithm to perform scene estimation. However, their work assume the identification of objects are given or provided by an idealized object detection and recognition system in order to perform A* search.

In terms of sequential manipulation, the KnowRob system [35] performs task-directed manipulation at the scale of entire buildings by integrating different knowledge sources from a perception system, an internal knowledge base and Internet repositories. Srivastava et al. [29] perform jointly task and motion planning for goal-directed manipulation relying upon a hard-coded perception system. Cogsun et al. [7] performs sequential manipulation for placing objects in a scene where the objects are separating on a clear table. Similar to *SUM* , Papasov et al. [25] perform sequential scene

estimation and manipulation through matching of known 3D geometries with an observed point cloud. However, this method takes a bottom-up approach using a RANSAC algorithm with Iterative Closest Point registration that neither requires nor uses a model of uncertainty.

B. Object Recognition and Pose Estimation

We consider two categories of traditional methods for model-based object recognition and pose estimation into two categories. First, feature-based methods, also known as descriptor-based methods, aim to match key features in the models to the scenes. Key features can be comprised of local or global descriptors [1]. Using local features [16], [28], [26], the pose of the object is estimated by first matching a set of extracted features from object model to the scene. Then, every matching pair will go through the filtering process to generate the final transformation. In contrast, methods using global features [2], [20], [27] attempt to match features with high resistance to the variance of the object pose. The object pose can be estimated by comparing those pose-preserving features from the training phase with the features computed from test scene. A limitation of feature-based methods is that the estimation quality will degrade as the number of objects (and clutter) in the scene increases due to occlusion of key features.

Alternatively, generative methods (or *analysis by synthesis*) attempt to find the state estimate that best describes the observed sensor input through iterating over comparisons with state hypotheses rendered into sensor readings. The *Render-Match-Refine* paradigm of Stevens and Beveridge [30] applied iterative optimization method to find a rendering that best explain the input. Their early work demonstrated Render-Match-Refine for 2D images, relying upon low-level feature extraction. Recent work uses convolutional neural networks (CNNs) to compare rendered and observed images. Among this work, Krull et al. [18] cast the CNN as a probabilistic model to output energy value, where as work by Gupta et al. [13] directly output a coarse object pose and use ICP to refine it later. However, these methods do not address multi-object pose estimation such as in cluttered scenes.

Similarly building on the renaissance in CNNs, object detection and recognition has greatly improved towards feasible general use. The Region-based Convolutional Neural Network method (R-CNN) [12] is a two-stage object detection system that integrates a high capacity CNN with bottom-up region proposal methods, which has demonstrated excellent detection accuracy.

III. PROBLEM FORMULATION

Given past RGBD observations $(z_0, \ldots, z_t)$ and robot manipulation actions $(u_0, \ldots, u_t)$, our aim is to estimate the scene as a collection of k objects, with object labels o, 2D image-space bounding boxes b_t, and six DoF object poses q_t. Note, k is the number of objects *recognized* in the scene. Object labels are strings containing a semantic identifier assumed to be a human-intuitive reference. In our

Fig. 2: An overview of our *SUM* framework. Given an observation of the scene, pre-trained R-CNN object detector and recognizer output bounding boxes and object labels along with their confidence. Assuming that every object is independent with each other, we estimate the state of the scene by estimating the state of each object individually. The numbers in the figure denotes the value of each term that composes to the posterior probability of a hypothesized object state. Multiple object pose estimator can originate from the same bounding box, for example, both the "shampoo" and "clorox" pose estimator originates from the same bounding box, and clorox is selected as the correct estimate since it has higher $p(x_t^i)$. More detail is explained in Section III and Equation 3.

experiments, the manipulation actions u_t are pick-and-place actions, which will invoke a motion planning process. However, our formulation is general such that u_t also applies to low level motor commands represented by joint torques, such as in scenarios for object tracking. The state of an individual object i in the scene is represented as $x_t^i = \{q_t^i, b_t^i, o^i\}$. We assume that every object is independent of all other objects, which implies there will be only one object with a given label in the eventual inferred scene estimate. Independence between objects allows us to state this scene estimation problem as:

$$p(x_t^1, \cdots, x_t^k | z_{0:t}, u_{1:t}) = \prod_{i=1}^{k} p(x_t^i | z_{0:t}, u_{1:t}) \quad (1)$$

where, for each object, the posterior probability is

$$
\begin{aligned}
&p(x_t^i | z_{0:t}, u_{1:t}) \\
&= p(q_t^i, b_t^i, o^i | z_{0:t}, u_{1:t}) \\
&= p(b_t^i | z_{0:t}, u_{1:t}) p(o^i | b_t^i, z_{0:t}, u_{1:t}) \cdot \\
&\quad p(q_t^i | b_t^i, o^i, z_{0:t}, u_{1:t}) \quad (2) \\
&= \underbrace{p(b_t^i | z_t)}_{detection} \underbrace{p(o^i | b_t^i, z_t)}_{recognition} \underbrace{p(q_t^i | b_t^i, o^i, z_{0:t}, u_{1:t})}_{Bel(q_t^i)} \quad (3)
\end{aligned}
$$

using the statistical chain rule and independence assumptions to yield Equations 2 and 3, respectively. Equation 3 represents the factoring of the scene estimation problem into object detection, object recognition, and belief over

object pose. The object detection factor $p(b_t^i | z_t)$ denotes the probability of object i being observed within the bounding box b_t^i given observation z_t. The object recognition factor $p(o^i | b_t^i, z_t)$ denotes the probability of this object having label o^i given the observation z_t inside the bounding box b_t^i. These distributions are generated as the output of a pre-trained discriminative object detector and recognizer that evaluates all possibilities. The implementation of these detectors and recognizers is as explained in Section IV. The pose belief factor for a particular object o^i is modeled over time by a recursive Bayesian filter, as illustrated in Figure 3. The belief over the object pose q_t^i at time t is estimated as:

$$
\begin{aligned}
&Bel(q_t^i) \propto \\
&\underbrace{p(z_t | q_t^i, b_t^i, o^i)}_{observation\ model} \int_{q_{t-1}^i} \underbrace{p(q_t^i | q_{t-1}^i, u_t, b_t^i, o^i)}_{action\ model} Bel(q_{t-1}^i) dq_{t-1}^i \quad (4)
\end{aligned}
$$

Further explanations of the observation likelihood function and the action model are in section IV.

A. Data Association

Across all objects, this Bayesian filtering framework also requires a data association process to correspond previous object estimators with the current detection and recognition. Data association for *SUM* maintains independent filters for each possible object, which are spawned or terminated based on object detection and recognition. At the initial instance

Fig. 3: Graphical model for estimating pose of a particular object o^i given observations, actions. The bounding box and object label hypothesis at each frame is based on object detection, recognition and data association as explained in Section IV.

of time $t = 0$, the number of objects k is estimated by thresholding on the detection and recognition results for the initial observation z_0. For each recognized object o^i along with its bounding box b_0^i, we will assign an object pose estimator to localize the object within the region defined by b_0^i. When the robot manipulates the objects in the scene for the next action u_1, the objects poses change as a result. To decide within which region that each object pose estimator should continue to localize the object o^i, there is a data association stage where it is associated to a bounding box b_1^i detected at time $t = 1$ after the robot action. Thus after every robot action u_t, the robot receives a new observation z_t, a data association stage takes care of associating each object estimator T^i with a bounding box b_t detected in z_t. More details on data association and when to terminate or add an object estimator is discussed in section IV.

IV. METHODOLOGY

A. Object Detection and Recognition

The *SUM* model above is agnostic to the specific algorithms for objects detection and recognition as long as distributions of possible object bounding boxes b and labels o can be generated. Specifically, each proposed detection of an object will have an bounding box in the image space with a probability of belonging to one of N object labels in the training database. For each generated bounding box at time t, we filter out an object proposal o_l, $1 \leq l \leq N$, if its confidence is smaller than a certain ratio σ_c of the maximum confidence in this bounding box:

$$p(o_l|z_0, b) < \max_{o_l} p(o_l|z_0, b) \cdot \sigma_c \quad (5)$$

where $\sigma_c \in (0 \ldots 1)$. The number of recognized objects k is determined by the above thresholding procedure. An object pose estimator T^i is associated to each recognized object i and its corresponding bounding box b_0^i and object label o^i pair.

B. Particle Filtering

Particle filtering is employed with each object estimator T^i to infer the pose q_t^i of object i. A particle filter is a means of inference for the sequential Bayesian filter in Eq. 4 through an approximation consisting of n weighted particles, $\{q_t^{ij}, w_t^{(j)}\}_{j=1}^n$. Weight $w_t^{(j)}$ for particle q_t^{ij} is expressed as:

$$Bel(q_t^i) \propto$$
$$p(z_t|q_t^i, b_t^i, o^i) \sum_j p(q_t^{ij}|q_{t-1}^{ij}, u_t, b_t^i, o^i) Bel(q_{t-1}^i) \quad (6)$$

as described by Dellaert et al. [8]. The initial belief of object pose is uniform. At each time instance, the weight of each hypothesis is computed, normalized to one, and resampled based on importance into an updated set of n particles:

$$q_t^i \sim \sum_j w_{t-1}^{(j)} p(q_t^{ij}|q_{t-1}^{ij}, u_t) \quad (7)$$

Before each robot action, we apply iterated likelihood weighting [21] to estimate the distribution of the object pose given the bounding box and the object label. This serves as a *bootstrap filter*, where the state transition in action model is replaced by a zero-mean Gaussian noise.

Our observation likelihood function measures how well a particle's rendered point cloud explains the observation point cloud. The observation model of this particle filter uses the z-buffer of a 3D graphics engine to render each particle q_t^{ij} into a depth image for comparison with the observation. This depth image, represented as $\hat{z}_t^{(j)}$, is backprojected into a point cloud $\hat{r}_t^{(j)}$ in the camera frame to simulate the camera model. The observation likelihood for each particle hypothesis with respect to the point cloud r_t associated with the observation z_t is then expressed as:

$$p(z_t|q_t^{ij}, b_t^i, o^i) = \frac{\sum_{a,b \in \hat{r}_t^{(j)}} \text{INLIERS}(r_t(a,b), \hat{r}_t^{(j)}(a,b))}{N_{z_t}} \quad (8)$$

where a and b are 2D indices in the rendered point cloud $\hat{r}_t^{(j)}$, N_{z_t} is the total number of points in the observation point cloud and

$$\text{INLIERS}(p, p') = \begin{cases} 1, & \text{if } \|p - p'\|_2 < \varepsilon \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

Thus, if the Euclidean distance between an observed point and a rendered point are within a certain sensor resolution ε, total number of inliers will increment by 1.

A robot manipulation action is represented by $u(j, \phi_{pick}, \phi_{place})$. This pick-and-place action is parametrized by the target object index j, object pick and place pose $\phi_{pick}, \phi_{place}$. For a particular object o^i, we use Gaussian components to model how the object pose q_t^i will change from q_{t-1}^i after a robot action $u_t(j, \phi_{pick}, \phi_{place})$,

$$p(q_t^i|q_{t-1}^i, u_t(j, \phi_{pick}, \phi_{place}), b_t^i, o^i)$$
$$\propto \begin{cases} w_1 \mathcal{N}(\phi_{place}, \sigma_1^2) + w_2 \mathcal{N}(q_{t-1}^i, \sigma_2^2), & \text{if } i = j \\ \mathcal{N}(q_{t-1}^i, \sigma_3^2), & \text{if } i \neq j \end{cases} \quad (10)$$

If the action u_t is targeted on object o^i, then either the action succeeds and the object is moved to the place pose ϕ_{place} with uncertainty characterized by σ_1^2, or the action fails and the object stays at its previous pose q_{t-1} with uncertainty characterized by σ_2^2. If the action u_t is not targeted on object o^i, then we assume that the object stays at its previous pose q_{t-1}^i with uncertainty characterized by σ_3^2. In cases where the action fails due the manipulated object being accidentally mishandled, the new pose q_t far from its previous pose q_{t-1} or its expected pose from manipulation success. This possibility is currently not modeled. Instead, data association will handle this object through the spawning of a new estimator.

C. Data Association

Data association is needed to associate each currently detected object bounding box with at most one object estimator at each moment in time. We use a greedy algorithm for our data association problem, which yields similar results at lower computational cost compared to the Hungarian algorithm (as reported by Breitenstein et al. [4]). First, a matching score matrix S of every pair (T^i, b_t^l) of object estimator and bounding box is calculated, with the matching score defined as

$$s(T^i, b_t^l) = IoU(b_{t-1}^i, b_t^l) p(b_t^l|z_t) p(o^i|b_t^l, z_t) \qquad (11)$$

which consists of three factors: the overlap between b_t^l and b_{t-1}^i by Intersection over Union (IoU), the likelihood of an object to be in bounding box b_t^l, the likelihood of object o^i inside the bounding box b_t^l. The pair (T^{i*}, b_t^{l*}) with the maximum score in S is selected as an established association. The rows and columns belonging to the object estimator T^{i*} and the bounding box b_t^{l*} are removed from S. This process is repeated until no further pairing is possible. In the end, we only keep the established associations with a matching score above a chosen threshold. A new object estimator is spawned for a bounding box b_t^l not associated with any existing object estimators. An object estimator will be terminated if it is not associated with any bounding boxes for K consecutive frames.

V. Implementation

Our implementation of *SUM* uses R-CNN in a manner that divides object detection problem into two stages: a) image proposal generation, and b) proposal classification. R-CNN was chosen because of its suitability for small datasets and relatively high accuracy. The baseline image proposals generation method used by R-CNN, selective search [37], was replaced for Edge Boxes [38] due to its computational efficiency and recall [14]. The Edge Box method is an edge-based proposal generation method which applies a score function to evaluate the completeness of contours that contain in a certain bounding box. By implementing structured decision tree, the evaluation process for a huge number of candidate boxes can be performed in a second. A Softmax layer was used for the final label output layer rather than a separate SVM.

Fig. 4: The *SUM* dataset objects (a). Eight of these objects in a cluttered scene (b) viewed as an observed depth image (c) and as ground truth (d).

For particle filtering, we employed the CUDA-OpenGL interoperation to render all particles in a single render buffer on the GPU and can be accessed by the CUDA kernels to compute the weights for particles. The major computation is operated on directly on GPU and there is very few data transfer between GPU and CPU memory. This provides a tractable solution for us to employ more particles to sample hypothesized objects.

Manipulation actions used TRAC-IK [3] to generate the joint states of the arm given the pose of the end-effector and MoveIt![31] to perform motion planning afterwards. Based on methods proposed by ten Pas and Platt [34], a custom grasp planning pipeline was developed to evaluate all possible grasp candidates based on the Darboux frame (surface normal and principal curvature axes) of each object vertex.

VI. Results

We first examined *SUM* on single scenes where estimates static images without robot actions. We compare *SUM* with a local descriptor, Fast Point Feature Histograms (FPFH) [27], on 10 test scenes of cluttered unstructured environment. For sequential manipulation, eight experiments for sorting objects into two groups were performed with a Fetch mobile manipulation robot.

SUM was run on a Ubuntu 14.04 system with an Titan X Graphics card and CUDA 7.5with 625 particles and 25 resampling iterations for all trials. σ_c is set to 0.1. Sensor resolution ε is set to 0.008 in meters. σ_1, σ_2 and σ_3 are set to 0.04, 0.02, 0.01 respectively. A custom dataset of 15 household objects (Figure 4) was used for testing, as well as 3D model generation. For CNN training, 8-10 streams of each object in the dataset was captured in a variety of different poses with different backgrounds. The whole training dataset contains 8366 ground truth images and 60563

Fig. 5: The plots compare the performance between our method *SUM* and FPFH with respect to the accuracy of correct poses. In each plot, there is a fixed translation error bound (1cm, 5cm, 10cm and 20cm), the x-axis is the changing rotation error bound and the y-axis is the percentage of the correct poses. Each point in the plot shows the accuracy of correctly localized objects with a fixed translation and rotation error bound.

background images. The Caffenet model [15] was used for network fine-tuning, which was trained on ImageNet [9].

A. Single Scene Estimation

SUM was evaluated and compared with FPFH on 10 single scenes, with 10 trials each, with respect to the accuracy of estimated object poses. We compute the accuracy of correct poses over all the test scenes and all the runs. Accuracy is defined as the number of correctly localized objects over the total number of detection true positives from the RCNN object detector, where a true positive has IoU greater than 0.5 between estimated and ground truth bounding boxes. We deem an object as correctly localized if its translation error and rotation error fall with chosen error bounds. The translation error is the Euclidean distance between estimated object position (x, y, z) and ground truth pose (x_{gt}, y_{gt}, z_{gt}) and the orientation error is the shortest angle error between estimate object $(roll, pitch, yaw)$ and ground truth $(roll_{gt}, pitch_{gt}, yaw_{gt})$.

The four plots in Fig. 5 depict the comparison between *SUM* and the baseline method. In each plot, there is a fixed translation error bound (1cm, 5cm, 10cm and 20cm) and the x-axis is the changing rotation error bound. The y-axis is the percentage of the correct poses. We can see in Fig. 5a, 5b and 5c, our method performs better than FPFH in the small error bounds (translation error smaller than 1cm, 5cm 10cm). *SUM* can also reject false positives from detection results. Fig 6 shows two examples of how *SUM* corrected false positives from detection results. We also calculated the mean ratio of rejected detection false positives. The mean ratio of rejected detection false positives over all the test scenes is 0.84 and the standard error over 10 runs is 0.0126.

B. Estimation and Manipulation on Sequential Scenes

In the manipulation evaluation, the robot must sort object on a cluttered tabletop into cleaning and non-cleaning categories by picking and placing the object into the right or left bin. In order to make a natural unstructured scene, we avoid manually placing objects in the scene by indiscriminately pouring the objects onto the cluttered table. After scene estimation by *SUM* , the object with the most likely estimate is selected to be grasped and sorted into the appropriate bin. No matter whether the robot succeeds or not, *SUM* updates the pose hypotheses by the action model, associates the object estimators with current detection results and estimates the scene iteratively.

The manipulation results are shown in Table I. Each scene contains five recognizable objects. We evaluate the method by the completion ratio of each sequence shown in the last row of the table. The completion ratio is how much the recognizable objects on the table are successfully sorted by the robot. The robot successfully completed six out of eight sequences. In sequence(a), failure occured when robot was trying to pick up "downy", it swept "sugar can" onto the ground. In sequence(f), the robot failed to pick up "waterpot" as the feasible grasp poses are out of joint limits of the arm. As shown in the second row, a manipulation action error occurs about once on average per trial. Despite such errors, *SUM* performs robustly to not only detection uncertainties but also manipulation failures. As shown in Figure 7, there is a manipulation error in the fourth action of the sequence, where the "spray bottle" slipped from the gripper. *SUM* subsequently estimated this object again and the robot picked it up successfully.

VII. CONCLUSION

In this paper, we propose *SUM* as a combined generative and discriminative approach to robust sequential scene estimation and manipulation. *SUM* utilizes output from discriminative object detector and recognizer to guide the generative process of sampling scene hypothesis for 6DOF pose estimation. By maintaining a belief over object poses over a sequence robot actions, *SUM* is able to perform robust estimation and manipulation in a cluttered and unstructured tabletop scenario.

REFERENCES

[1] 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.

[2] 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.

[3] P. Beeson and B. Ames. Trac-ik: An open-source library for improved solving of generic inverse kinematics. In *IEEE-RAS International Conference on Humanoid Robots*, 2015.

[4] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Robust tracking-by-detection using a detector confidence particle filter. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1515–1522. IEEE, 2009.

Fig. 6: (a)(d) The detection results from EgdeBox and R-CNN object detector . (b)(e) Intermediate results from *SUM* which contain false positive estimation (c) The scene estimate result of (a) after thresholding, correct the false positives: two "spray bottle", a "shampoo" and a "sugar". (f) The scene estimate result of (d), correct the false positives: "ranch", "waterpot", "spray bottle" and "tide".

	Sequence(a)	Sequence(b)	Sequence(c)	Sequence(d)	Sequence(e)	Sequence(f)	Sequence(g)	Sequence(h)
Number of total objects	5	5	5	5	5	5	5	5
Number of Manipulation Errors	1	1	2	0	0	1	1	0
Number of Manipulation Trials	4	6	7	5	5	5	6	5
Completion Ratio	0.80	1.0	1.0	1.0	1.0	0.8	1.0	1.0

TABLE I: The tables shows results of manipulation experiments for 8 sequences. The first row shows the number of object in each scene. Row two and three show the the count for the manipulation errors and trials for the sequence. The last row shows the ratio of how much the recognizable objects on the table are successfully sorted by the robot.

[5] 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 Berlin Heidelberg, 2014.

[6] A. Collet, M. Martinez, and S. S. Srinivasa. The moped framework: Object recognition and pose estimation for manipulation. *Int. J. Rob. Res.*, 30(10):1284–1306, Sept. 2011.

[7] A. Cosgun, T. Hermans, V. Emeli, and M. Stilman. Push planning for object placement on cluttered table surfaces. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4627–4632, 2011.

[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] K. Desingh, O. C. Jenkins, L. Reveret, and Z. Sui. Physically plausible scene estimation for manipulation in clutter. In *IEEE-RAS International Conference on Humanoid Robots (Humanoids 2016)*, 2016.

[11] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010.

[12] 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.

[13] S. Gupta, P. Arbeláez, R. Girshick, and J. Malik. Inferring 3d object pose in rgb-d images. *arXiv preprint arXiv:1502.04652*, 2015.

[14] J. Hosang, R. Benenson, and B. Schiele. How good are detection proposals, really? *arXiv preprint arXiv:1406.6962*, 2014.

[15] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.

[16] 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.

[17] D. Joho, G. D. Tipaldi, N. Engelhard, C. Stachniss, and W. Burgard. Nonparametric bayesian models for unsupervised scene analysis and reconstruction. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2012.

[18] A. Krull, E. Brachmann, F. Michel, M. Ying Yang, S. Gumhold, and C. Rother. Learning analysis-by-synthesis for 6d pose estimation in rgb-d images. In *Proceedings of the IEEE International Conference*

Fig. 7: Sequential manipulation by a robot to sort 5 objects on a cluttered tabletop into two bins: cleaning (right bin) and non-cleaning (left bin). From the left to right is the detection results from RCNN object detector, most likely object from *SUM* , computed collision-free grasp poses and robot manipulating the object in action. Our system estimated and manipulated "tide", "scotch brite", "spray bottle", "sugar" and "toy" sequentially.

on Computer Vision, pages 954–962, 2015.

[19] 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. Special Issue on Knowledge Driven Robotics and Manufacturing.

[20] Z.-C. Marton, D. Pangercic, N. Blodow, and M. Beetz. Combined 2d–3d categorization and classification for multimodal perception systems. *The International Journal of Robotics Research*, 30(11):1378–1402, 2011.

[21] 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.

[22] 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.

[23] 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*, pages 5052–5059. IEEE, 2016.

[24] S. Narayanaswamy, A. Barbu, and J. M. Siskind. A visual language model for estimating object pose and structure in a generative visual

domain. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 4854–4860. IEEE, 2011.

[25] C. Papazov, S. Haddadin, S. Parusel, K. Krieger, and D. Burschka. Rigid 3d geometry matching for grasping of known objects in cluttered scenes. *The International Journal of Robotics Research*, page 0278364911436019, 2012.

[26] 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.

[27] 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.

[28] R. B. Rusu, Z. C. Marton, N. Blodow, and M. Beetz. Persistent point feature histograms for 3d point clouds. In *Proc 10th Int Conf Intel Autonomous Syst (IAS-10), Baden-Baden, Germany*, pages 119–128, 2008.

[29] S. Srivastava, L. Riano, S. Russell, and P. Abbeel. Using classical planners for tasks with continuous operators in robotics. In *Proceedings of the ICAPS Workshop on Planning and Robotics (PlanRob)*, 2013.

[30] M. R. Stevens and J. R. Beveridge. Localized scene interpretation from 3d models, range, and optical data. *Computer Vision and Image Understanding*, 80(2):111–129, 2000.

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

[32] Z. Sui, O. C. Jenkins, and K. Desingh. Axiomatic particle filtering for goal-directed robotic manipulation. In *International Conference on Intelligent Robots and Systems (IROS 2015)*, Hamburg, Germany, Oct 2015.

[33] 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.

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

[35] M. Tenorth and M. Beetz. Knowrob: A knowledge processing infrastructure for cognition-enabled robots. *Int. J. Rob. Res.*, 32(5):566–590, Apr. 2013.

[36] S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. MIT press, 2005.

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