

Goal-directed Manipulation of Objects in Contact-based Clutter through Probabilistic Pose Estimation

Zhiqiang Sui

Zheming Zhou

Odest Chadwicke Jenkins

Laboratory for Progress (Perception Robotics and Grounded Reasoning Systems)

Department of Electrical Engineering and Computer Science

University of Michigan

Ann Arbor, MI, USA 48109-2121

Abstract

Performing robust goal-directed manipulation tasks remains a crucial challenge for autonomous robots. In the ideal case, we would like to be able to specify a high-level goal state for a robot, such as a desired scene, and have the robot reason over the actions and motions to achieve this goal. However, realizing this goal remains elusive due to the problem of perceiving the robot’s environment, which is especially challenging in the cluttered scenes. We propose a generative approach to this problem by probabilistically inferring the 6DOF object poses and then inferring the robot’s environment as a scene graph in a manner amenable to use symbolic inference for task planning and collision-free motion planning as well as execution. Considering the complexity of cluttered scenes, we also propose a GPU-optimized likelihood evaluation and a grasp pose ranking algorithm to generate feasible grasp poses for complex geometries. Results from our methods used for goal-directed manipulation in multi-object scenes by a Fetch robot.

1 Introduction

An ongoing challenge for autonomous robots is to perform robust goal-directed sequential manipulation, especially in everyday life where objects are often in the contact-based clutter. However, the perception has been the main bottleneck. In the ideal case, our aim is to allow human users to specify their desired state of the world and then realizing this goal through the robot’s autonomous reasoning and control. That is, truly autonomous robot manipulators need the ability to perceive the world, reason over manipulation actions towards the goal, and carry out these actions in terms of physical motion. There have been considerable advances in reasoning for robot decision making and purposeful robot motion, both of which are increasingly converging [Kaelbling and Lozano-Pérez, 2013; Srivastava *et al.*, 2013]. However, robots still lack the general ability to perceive the world, especially in typical human environments with contact-based clutter. This shortage affects both the robot’s ability to reliably make decisions as well as carry out manipulation actions.

Without the ability to perceive in daily and cluttered environments, methods for joint tasks and motion planning remain restricted to simulation and highly controlled environments.

Our previous work, the *Axiomatic Particle Filter (APF)* [Sui *et al.*, 2015], proposed a generative approach to scene estimation for robot manipulation, to address the problem of robot perception in cluttered scenes. The APF iteratively generates possible scene configurations (as axioms asserting object poses and relations between objects) and tests whether these hypotheses against the robot’s observations. The result of the APF is an approximate posterior probability distribution over possible scenes, where a scene estimation of maximal likelihood can be taken. Though probabilistic in nature, the result of the APF represented in axiomatic form with the intention of “closing the loop” using goal-directed symbolic planners [Fikes and Nilsson, 1972; Laird *et al.*, 1987]. Such planners reason over manipulation actions for the robot to execute towards realizing a given goal scene. Planned sequences of actions are then executed by motion planners for pick-and-place action [Ciocarlie *et al.*, 2014] or general manipulation affordances [Hart *et al.*, 2015]. However, one of the limitations of APF is it only considers three degrees of freedom in estimation of object poses(x , y and yaw) while in the real life scene, there are lots of scenarios where object poses could be in any orientations. The other limitation is the high dimensionality of sampling scene graphs.

In this paper, we explore a different way to estimate scene graph by first estimating the object poses in a generative model probabilistically and then inferring the axioms describing the current world. We propose a generative object pose estimation algorithm in contact-based cluttered scenes. For tractability, we also introduce a GPU-optimized likelihood computation incorporating both depth and normal information. For complex geometries, it is also critical to generating graspable poses in different cluttered scenes. So we also present a grasp pose ranking algorithm to generate feasible graspable poses from the resulting estimated objects. The paper organizes as follows, section 2 talks about related work and section 3 covers the methodology for the pose estimation algorithm in particle filter and the PDDL-based axioms. Section 4 presents the GPU-based likelihood evaluation and the grasp pose ranking algorithm. Then section 5 demonstrates the experiments on object pose estimation and the

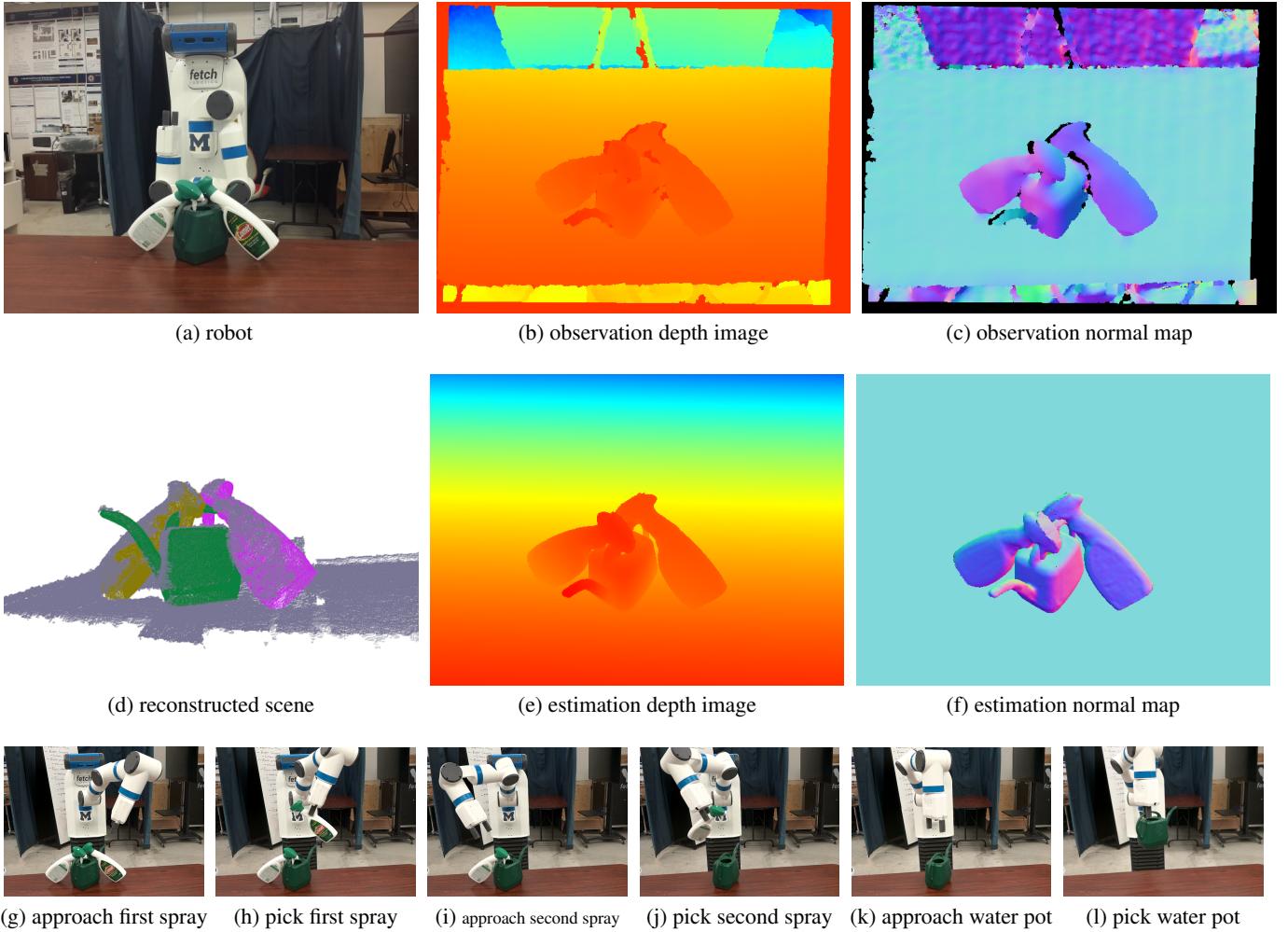


Figure 1: Robot manipulation in a cluttered scene with objects varying in dimensions and geometries. Observation and estimated depth image along with reconstructed point cloud are in the top row. Frames of robot performing pick and place actions are in the bottom row. Demo video¹

goal-directed manipulation system. Section 6 concludes the paper. Figure 1 shows our goal-directed manipulation in action for a cluttered scene.

2 Related Work

The problem addressed in this paper is to infer object-level manipulation semantics, in the form of axioms, from 3D point clouds, or 3D maps more generally. In this situation, our system can be considered as a form of semantic mapping for robots [Kuipers, 2000], which is a case of the general problem of anchoring [Coradeschi and Saffiotti, 2003] (which itself is a case of symbol grounding [Harnad, 1990]). Based on the semantic mapping work of Rusu et al. [Rusu et al., 2008], the PR2 Interactive Manipulation pipeline [Ciocarlie et al., 2014] is able to perform relatively reliable pick-and-place manipulation for non-touching objects in tabletop settings. This method assumes individual objects extrude from a

single flat support surface based on surface normal clustering.

This paper builds on the Axiomatic Particle Filter proposed by Sui et al. [Sui et al., 2015] for generative scene estimation to perform goal-directed manipulation. The methods used for the APF are similar in aim to that of belief space planning [Kaelbling and Lozano-Pérez, 2013]. While belief space planning reasons over distributions of possible plans, the APF takes an estimate of a scene for both tractability and reasoning by classical symbolic planning. From the symbolic perspective, recent work by Mohan et al. [Mohan et al., 2012; Kirk and Laird, 2013] uses the Soar cognitive architecture to perform goal-directed control of a robot arm. In the presence of occlusion, Soar is able to reason over actions for robots to play games such as tic-tac-toe and Connect-4 by constructing a scene graph which is equivalent to the axiomatic representation in this paper. Similarly, Narayanaswamy et al. [Narayanaswamy et al., 2011] perform scene estimation and goal-directed robot manipulation for cluttered scenes of Lin-

¹<https://youtu.be/MciM0cg4qEM>

coln Logs. Chao et al. [Chao *et al.*, 2011] also perform symbolic goal-directed manipulation for tasks involving robot learning from demonstrations. By taking a generative approach to the scene estimation, APF-based methods can perform similar goal-directed capabilities with less susceptibility to false positives but at the cost of greater computational complexity.

There have been a number of discriminative approaches proposed to advance robot perception for manipulation. Rosman and Ramamoorthy [Rosman and Ramamoorthy, 2011] address the problem of relational scene graph estimation by detecting contact points between objects which can be directly observed from depth. Papazov et al. [Papazov *et al.*, 2012] use a bottom-up approach of matching the 3D object geometries using RANSAC and retrieval by hashing methods. The KnowRob system of Tenorth and Beetz [Tenorth and Beetz, 2013] performs goal-directed sequential manipulation to address uncertainty at the symbolic level with knowledge from sources on the web. KnowRob relies on hard state estimates using hard-coded software components provided in the Robot Operating System (ROS) [Quigley *et al.*, 2009]. Similarly, Srivastava et al. [Srivastava *et al.*, 2013] rely on hard-coded perception systems to perform the joint task and motion planning, taking advantage of modifications in controlled environments (e.g., “green screening”, augmented reality tags). In contrast to discriminative perception, a generative approach maintains a distribution over all possible scene graphs and is not reliant upon selecting and maintaining a hard (potentially incorrect) state estimation for perception. Similar with particle filtering for robot localization, APF-based methods could be used to complement these methods for extending to broader collections of robot systems.

3 Methodology

3.1 Particle Filter

In the problem of object pose estimation, each x_t is composed of a set of 6 DOF object poses which hypothesizes a possible configuration in the real world. The inference of the state x_t from a history of robot observations $z_{1:t}$ is then modeled as a sequential Bayesian filter. This model consists of updating a prior belief from time $t - 1$ with a dynamic resampling and likelihood evaluation to form a new posterior belief at time t :

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

Although results presented are for observations of static scenes, the dynamics term $p(x_t|x_{t-1}, u_{t-1})$ in this formulation is to retain generality for tracking the scene as the robot performs an action u_{t-1} . As described by [Dellaert *et al.*, 1999], the sequential Bayesian filter in Eq. 1 is commonly approximated by a collection of N weighted particles, $\{X_t^{(j)}, w_t^{(j)}\}_{j=1}^N$, with weight $w_t^{(j)}$ for particle $X_t^{(j)}$, expressed as:

$$p(x_t|z_{1:t}) \propto p(z_t|x_t) \sum_j w_{t-1}^{(j)} p(x_t|x_{t-1}^{(j)}, u_{t-1}) \quad (2)$$

Each particle $x_t^{(j)}$ is rendered into a depth image and a normal map, represented as $\hat{z}_t^{(j)}$, through the fragment shader of a 3D graphics engine, for comparison with the robots current observation z_t :

$$p(z_t|x_t^{(j)}) = e^{-\lambda_r \cdot d(z, \hat{z}_t^{(j)})} \quad (3)$$

where λ_r is a constant scaling factor and

$$d(z, \hat{z}_t^{(j)}) = \lambda_1 d_1(D, D') + \lambda_2 d_2(N, N') \quad (4)$$

where D and D' are the depth images and N and N' are the normal maps. λ_1 and λ_2 are the normalizing factor for d_1 and d_2 , as we regulates that $\lambda_1 + \lambda_2 = 1$.

$$d_1(D, D') = \sum_{(a,b) \in z} (D(a,b) - D'(a,b))^2 \quad (5)$$

is the sum square of distance function (SSD) between depth images D and D' , where a and b are 2D image indices. And $d_2(N, N')$ is defined as:

$$d_2(N, N') = \sum_{(a,b) \in z} \text{acos}(\text{dot}(N(a,b), N'(a,b))). \quad (6)$$

For the normal vector in each pixel, we compute the acos of the dot product of two normal vectors. If the direction of two normal vectors are closer, d_2 would get a smaller value. So the likelihood of a particle is the exponential to the negative sum of d_1 and d_2 , where smaller distance results a larger likelihood.

3.2 PDDL-based Axioms

Once the posterior distribution converges about a single hypothesis, the most likely particle \hat{x}_t is taken as the current estimate and then inferred to a set of axioms for planning robot actions and motion towards a given goal state which is expressed in axiomatic form.

A particle can be inferred to a set of axiomatic assertions describing the pose and geometry of each object and relations for object interactions and affordances. The Planning Domain Definition Language (PDDL) [Mcdermott *et al.*, 1998] is used to model axiomatic state as a formal language. These axioms assert the existence of an object W_i , as (W_i object), with spatial geometry V_i , ($geom W_i V_i$), and spatial pose configuration Q_i , ($pose W_i Q_i$). Axioms also assert parent-child relationships between objects as whether an object W_i is inside another object W_j , ($in W_i W_j$), or resting on another object W_k , ($on W_i W_k$), as well as whether the object is the possession of a robot R , ($has R W_i$).

We adopt a trivial method to infer from estimated object poses to axioms. Axioms about the object’s properties can be inferred directly from the particle and the spatial relations can be inferred by ordering the object poses and then checking the dimensions of each object.

4 Implementation

4.1 Parallelized Likelihood Evaluation

Towards meeting the time-critical needs of robotics, we designed an optimized likelihood function based on the parallel

evaluation of particles. This likelihood function rapidly renders scene particle hypotheses in parallel using OpenGL to simulate depth cameras. The renderer will set up the rendering pipeline using camera extrinsic and intrinsic parameters, object geometries, and estimated object transformations. During each particle filter iteration, the OpenGL renderer will render all particles in parallel onto a single render buffer, which is then passed to CUDA kernels for computing the objective metrics of particles.

Here, a particle is a scene consisting of objects with known geometries but posed with different transformations. Each particle is specified by a draw call for its object geometries and transformations that respectively render into a sub-image of the output buffer (based on specifications to `glViewport()`). With internal GPU work scheduling, all particles are rendered in parallel and viewport specification without a reduction in parallelism.

Similar to [Choi and Christensen, 2013], we attach the output render buffer to a framebuffer object (FBO) for efficient off screen rendering. However, we use render buffer objects (RBO) instead of textures because multisampling features in textures are not useful for our purposes and only add overhead. We also attach a depth render buffer to the FBO which is required for depth enabled rendering. However, RBOs in depth format are not supported by CUDA and cannot be accessed from CUDA kernels via the OpenGL interoperation interfaces. We devise a new method for this by modifying the OpenGL fragment shader to compute the depth and normal values and output as 4 bits float color values in a color formatted `GL_RGBA32F` RBO, which can be accessed from CUDA.

Fragment shaders have access to a built-in variable `gl.FragCoord = (x, y, z, 1/w)` in which w is the extra dimension of the clip-space homogeneous coordinate of the fragment. The depth in the camera coordinate is then represented by $Z = w$. Thus, the depth values can be computed in the fragment shader with `color = 1/gl.FragCoord.w;`. The normal values can be computed in the vertex shader by using model and view matrix and passed to the fragment shader. By leveraging the fragment shader that is already part of the existing rendering pipeline, this approach obtains depth and normal values in one pass and eliminates the overhead of extra copying from a depth RBO to a color RBO.

The color RBO contains depth and normal values is then passed to CUDA kernels through memory mapping with no data transfer and minimal overhead. The CUDA kernels compute the exponential squared error objective for each pixel and rearrange the memory layout to compute the sums of errors for each particle. The sums are then normalized and used as weights in particle filter resampling.

4.2 Grasp Pose Ranking

In order to grasp every object, we propose a grasp pose ranking algorithm to find a set of feasible grasp poses.

For every object, we first generate a local point cloud with object's center locating in the coordinate origin. Then, we manually select the points in the point cloud where the object can be grasped by robot, denoting these regions as local graspable points(LGPs). Given the estimated pose of the object

that already calculated in current scene, we can determine the translation and rotation matrix which maps object from local point cloud to estimated pose which is in the robot base frame. After transforming all the objects from object space to robot base frame, all LGPs will also be available on current scene.

Then we introduce grasp pose score(GPS) to rank all the LGPs in current scene. GPS is described as:

$$GPS = S_{dsupp} + S_{dobj} + S_H \quad (7)$$

where S_{dsupp} is the score measures the distance between LGPs and support plane(like table), S_{dobj} measures the distance between LGPs in different objects, and S_H measures the absolute height of the LGPs.

S_{dsupp} takes LGPs higher in score if it is further from the support plane, but if the distance between LGPs and support plane(d_{ls}) is larger than a threshold d_{lss} , the score will not change any more. S_{dsupp} is calculated as:

$$S_{dsupp} = \begin{cases} \frac{d_{ls}}{d_{lss}} & d_{ls} < d_{lss} \\ 1 & d_{ls} \geq d_{lss} \end{cases} \quad (8)$$

S_{dobj} will be higher if the LGP is further from the LGPs in other objects. Still, S_{dobj} will saturate if it reaches a certain distance threshold d_{objs} . Denote the distance between current LGP and nearest LGP in other objects as d_{obj} , S_{dobj} is defined as:

$$S_{dobj} = \begin{cases} \frac{d_{obj}}{d_{objs}} & d_{obj} < d_{objs} \\ 1 & d_{obj} \geq d_{objs} \end{cases} \quad (9)$$

As for the S_H , it is calculated by:

$$S_H = \frac{z}{z_{hh}} \quad (10)$$

where z denotes the absolute height of current LGP, and z_{hh} denotes the height of the highest LGP in the same object.

After calculating all the LGPs' GPS, we can extract the highest LGP in that object as the current grasp pose. If there exists multiple LGPs sharing the same score, we choose the one with the highest S_H . Meanwhile, it is noteworthy that, we take $S_{dobj} = 1$ for every LGP in the last picking object since there will be no other object in the scene when it is grasped.

5 Results

Our evaluation examined the proposed methods on perception as well as manipulation in contact-based cluttered scenes. In the object pose estimation experiment, we evaluated our probabilistic pose estimation algorithm on observations from the Primesense depth sensor mounted on the head of a Fetch robot. These objects are common to households and vary in dimensions and geometries. We then closed the loop by testing the goal-directed sequential manipulation with estimation and planning on a real cluttered scene. All the experiments are tested on a Linux PC with Intel Core i7, 32 GB memory and an NVidia GeForce GTX Titan X Graphic Card with CUDA 7.5.

5.1 Object Pose Estimation

To test the proposed generative probabilistic object pose estimation algorithm, we perform the experiments on cluttered

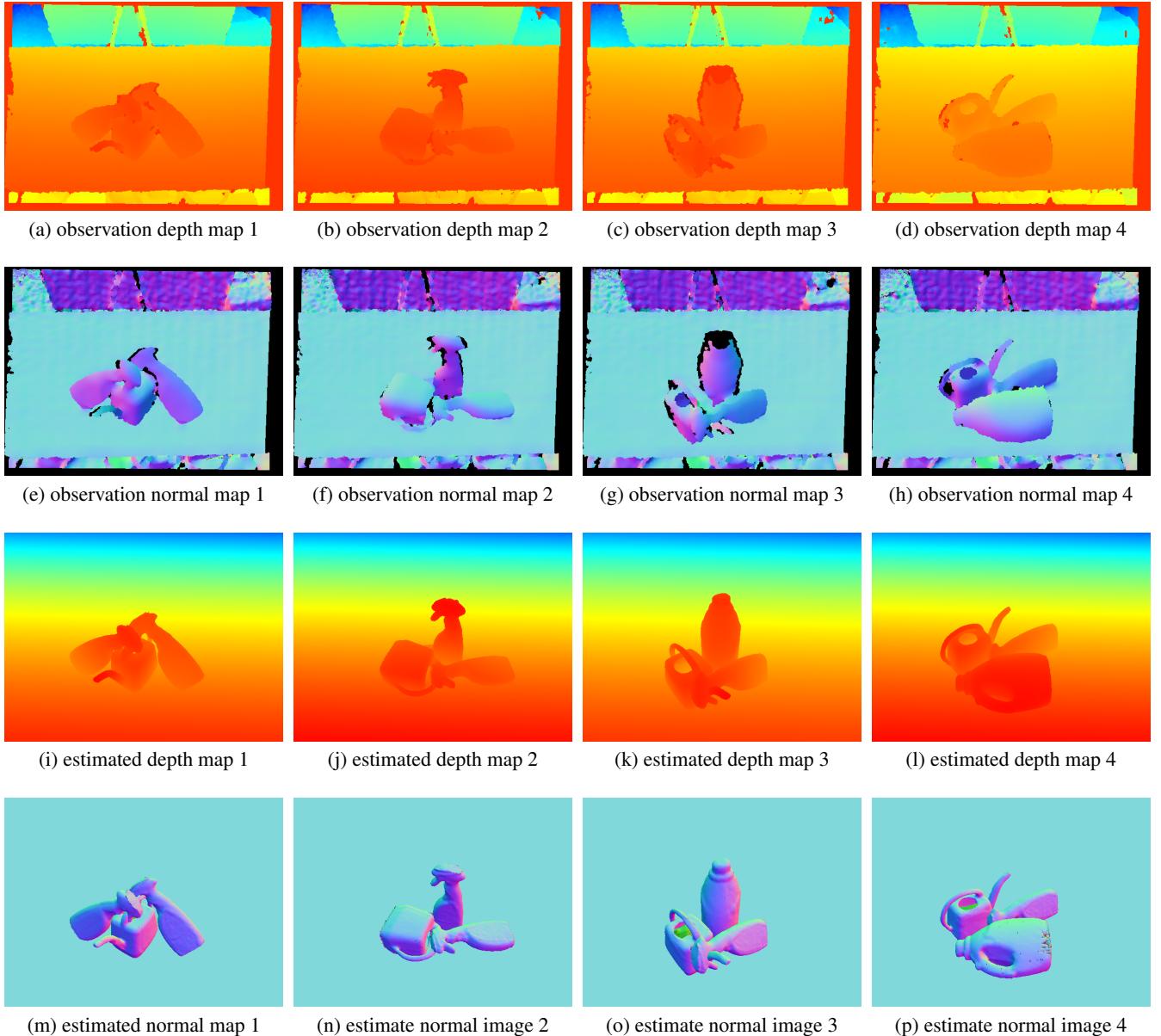


Figure 2: Object pose estimation results for four contact-based cluttered scenarios. Each column corresponds to a scene and its estimation results.

scenes. For each scene, the input are a depth map and a normal map both at a 480×640 pixel resolution. The depth map is retrieved from the Primesense depth sensor and the corresponding normal map is computed using PCL[Rusu and Cousins, 2011] 1.7 from the point cloud. The particle filter ran for a fixed 400 iterations with 625 particles. 625 is determined by the largest render buffer size (163484×16384) supported by the graphics card with respect to the size of the depth image (640×480). The initialization of the particle filter is to uniformly sample 6 DOF value of object poses within OpenGL camera viewport which simulates the real robot camera.

Figure 3 shows the estimation results for four scenes. Each column corresponds to an observation and the estimated depth and normal maps. Notice that in the depth map of an observation, there are a certain amount of null points when objects are contacted with each other. By leveraging the power of generative model, our algorithm precisely estimated the pose of each object. Accounting normal maps into the likelihood function also helps to achieve faster and more robust estimation results than our previous work [Sui *et al.*, 2015] using depth information alone.

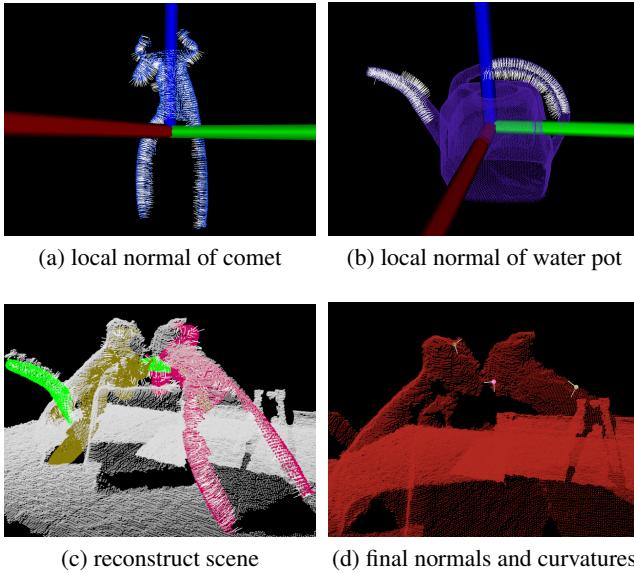


Figure 3: The top row shows the normals for manipulable parts of each object in its object space (precomputed). The bottom row shows the results for grasp pose ranking algorithm for the scene in Figure 1.

5.2 Manipulating three objects

To close the manipulation loop and validate our grasp ranking algorithm, we demonstrate three objects manipulation on the Fetch robot. The scene includes two spray bottles and one water pot, the local point clouds and LGPs with normals are generated as Figure 3(a) and Figure 3(b).

Given estimated pose of spray bottles and water pot in the scene, we transform objects from local frame to robot base frame which is shown in 3(c).

After the spray bottles and water pot are successfully fitted, we set $d_{lss} = 0.076m$ and $d_{objs} = 0.080m$ and calculate the GPS for every LGP. For every object, the algorithm chose the grasp pose based on the ranking criteria. Since the water pot is the last object for grasping according to the result from symbolic planner, the grasp pose for the water pot is the LGP with the highest position. The result of grasp poses and the corresponding normals for these objects are shown in 3(d).

The execution sequence of the Fetch robot is shown in Figure 1(g) to Figure 1(l). Figure 1(g), Figure 1(i), and Figure 1(k) shows the Fetch moves to pre-grasp pose of object while Figure 1(h), Figure 1(j), and Figure 1(l) illustrates the Fetch successfully manipulating the object and lifting to the post-grasp pose.

6 Conclusion and Future Work

In this paper, we proposed a probabilistic object pose estimation algorithm with GPU-optimized likelihood computation incorporating both depth and normal information. We demonstrate the algorithm is able to estimate object poses in contact-based cluttered scenes for manipulation. We also present a grasp pose ranking algorithm to generate feasible graspable poses from the resulting estimated objects. The

robot is then able to perform goal-directed manipulation from the action sequences generated by the PDDL planner.

As we are dealing with static scenes so far, one of our future work can be using MCMC to do optimization in such scenarios and get a better convergence. Real-time tracking is also critical to perform sequential perception and manipulation and it can make the most use of the particle filter by choosing an appropriate dynamic function.

References

- [Chao *et al.*, 2011] Crystal Chao, Maya Cakmak, and Andrea L. Thomaz. Towards grounding concepts for transfer in goal learning from demonstration. In *Proceedings of the Joint IEEE International Conference on Development and Learning and on Epigenetic Robotics (ICDL-EpiRob)*. IEEE, 2011.
- [Choi and Christensen, 2013] Changhyun Choi and Henrik I. Christensen. RGB-d object tracking: A particle filter approach on GPU. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 1084–1091, 2013.
- [Ciocarlie *et al.*, 2014] Matei Ciocarlie, Kaijen Hsiao, Edward Gil Jones, Sachin Chitta, Radu Bogdan Rusu, and Ioan A Ţucan. Towards reliable grasping and manipulation in household environments. In *Experimental Robotics*, pages 241–252. Springer Berlin Heidelberg, 2014.
- [Coradeschi and Saffiotti, 2003] Silvia Coradeschi and Alessandro Saffiotti. An introduction to the anchoring problem. *Robotics and Autonomous Systems*, 43(2):85–96, 2003.
- [Dellaert *et al.*, 1999] Frank Dellaert, Dieter Fox, Wolfram Burgard, and Sebastian Thrun. Monte carlo localization for mobile robots. In *IEEE International Conference on Robotics and Automation (ICRA) 1999*, May 1999.
- [Fikes and Nilsson, 1972] Richard E Fikes and Nils J Nilsson. Strips: A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, 2(3):189–208, 1972.
- [Harnad, 1990] Stevan Harnad. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1):335–346, 1990.
- [Hart *et al.*, 2015] S. Hart, P. Dinh, and K. Hambuchen. The affordance template ros package for robot task programming. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 6227–6234, May 2015.
- [Kaelbling and Lozano-Pérez, 2013] Leslie Pack Kaelbling and Tomás Lozano-Pérez. Integrated task and motion planning in belief space. *The International Journal of Robotics Research*, page 0278364913484072, 2013.
- [Kirk and Laird, 2013] James R. Kirk and John E. Laird. Learning task formulations through situated interactive instruction. In *Proceedings of the Second Annual Conference on Advances in Cognitive Systems*, pages 219–236, 2013.

- [Kuipers, 2000] Benjamin Kuipers. The spatial semantic hierarchy. *Artificial Intelligence*, 119(1):191–233, 2000.
- [Laird *et al.*, 1987] John E Laird, Allen Newell, and Paul S Rosenbloom. Soar: An architecture for general intelligence. *Artificial intelligence*, 33, 1987.
- [Mcdermott *et al.*, 1998] D. Mcdermott, M. Ghallab, A. Howe, C. Knoblock, A. Ram, M. Veloso, D. Weld, and D. Wilkins. Pddl - the planning domain definition language. Technical Report TR-98-003, Yale Center for Computational Vision and Control,, 1998.
- [Mohan *et al.*, 2012] Shiwali Mohan, Aaron H Mininger, James R Kirk, and John E Laird. Acquiring grounded representations of words with situated interactive instruction. In *Advances in Cognitive Systems*. Citeseer, 2012.
- [Narayanaswamy *et al.*, 2011] Siddharth Narayanaswamy, Andrei Barbu, and Jeffrey Mark 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.
- [Papazov *et al.*, 2012] Chavdar Papazov, Sami Haddadin, Sven Parusel, Kai Krieger, and Darius Burschka. Rigid 3d geometry matching for grasping of known objects in cluttered scenes. *The International Journal of Robotics Research*, page 0278364911436019, 2012.
- [Quigley *et al.*, 2009] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng. Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software*, 2009.
- [Rosman and Ramamoorthy, 2011] Benjamin Rosman and Subramanian Ramamoorthy. Learning spatial relationships between objects. *Int. J. Rob. Res.*, 30(11):1328–1342, September 2011.
- [Rusu and Cousins, 2011] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.
- [Rusu *et al.*, 2008] Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, Mihai Dolha, and Michael Beetz. Towards 3d point cloud based object maps for household environments. *Robot. Auton. Syst.*, 56(11):927–941, November 2008.
- [Srivastava *et al.*, 2013] Siddharth Srivastava, Lorenzo Riano, Stuart Russell, and Pieter Abbeel. Using classical planners for tasks with continuous operators in robotics. In *Proceedings of the ICAPS Workshop on Planning and Robotics (PlanRob)*, 2013.
- [Sui *et al.*, 2015] Zhiqiang Sui, Odest Chadwicke Jenkins, and Karthik Desingh. Axiomatic particle filtering for goal-directed robotic manipulation. In *International Conference on Intelligent Robots and Systems (IROS 2015)*, Hamburg, Germany, Oct 2015.
- [Tenorth and Beetz, 2013] Moritz Tenorth and Michael Beetz. Knowrob: A knowledge processing infrastruc-
ture for cognition-enabled robots. *Int. J. Rob. Res.*, 32(5):566–590, April 2013.