

Human-in-the-loop Affordance Registration via Pose Estimation

Zhefan Ye¹, Odest Chadwicke Jenkins¹, Zhiqiang Sui¹, Stephen Hart²

Abstract—Successful remote operation requires a careful balance of human operation and robot autonomy, commonly referred to as the problem of shared autonomy, which remains an open challenge for robotics where the perception of manipulation affordances for shared autonomy is a critical bottleneck. Through the use of *spectrum of autonomy*, we introduce a human-in-the-loop pose estimator and demonstrate its use in three scenes, followed by manipulation task execution using affordance template framework.

I. INTRODUCTION

Reliable operation of remote autonomous mobile manipulators remains an open challenge for robotics where the perception of manipulation affordances for shared autonomy is a critical bottleneck. Within the well-known sense-plan-act paradigm, truly autonomous robot manipulators need the ability to perceive the world, reason over manipulation actions afforded by objects towards a given goal, and carry out these actions in terms of physical motion. As the complexity of robotic systems and relevant tasks increases, full autonomy and learning for dexterous robot manipulation is beyond the foreseeable state-of-the-art, especially for tasks in remote, unstructured environments. Conversely, direct teleoperation is also not feasible in these scenarios, as deployed systems are often highly complex (with multiple degrees of freedom and high-resolution sensors) and are often controlled over sub-optimal communication channels. As such, successful remote operation requires a careful balance of human operation and robot autonomy, commonly referred to as the problem of shared autonomy.

With shared autonomy in mind, Hart et al. developed *affordance template framework* [11] to enable mobile manipulation and humanoid robot control, such as for robots in the DARPA Robotics Challenge [16] and on the International Space Station [1]. An affordance template is an adjustable pairing of 3D object geometries and sequence of robot actions represented in object-centric coordinates. However, robot operation with affordance templates currently requires manual placement/registration onto a 3D map (point cloud) and the selection of appropriate command sequence (or strategy) relevant to meet the demands of the context.

As a replacement for this manual registration, we present an automated method to fit/register object geometries associated with an affordance onto a 3D geometric map as a “human-in-the-loop” estimator. Since manual registration is

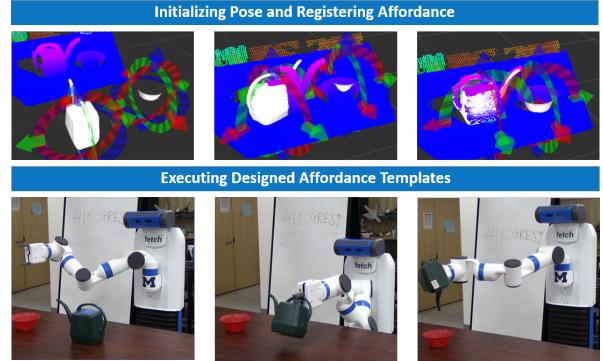


Fig. 1: Top row: user gives a rough initial pose estimation, after which a pose estimation method refines the pose to register the affordance template. Bottom row: affordance template framework performs a series of designed waypoints to perform a task.

often labor-intensive, a human-in-the-loop estimator can alleviate cumbersome fine-tuning by the user. Instead, the user can perform a coarse initialization first using the interactive tools and let perception system take over and perform fine registration. Once an object associated with an affordance is registered, the action of the affordance can then be executed by a robot as demonstrated in Figure 1.

Our contributions are three-fold: 1) we introduce our *spectrum of autonomy* used for human-in-the-loop affordance registration and demonstrate its use in three scenes for *using a spray, pouring, and door opening*; 2) we leverage spectrum of autonomy with generative pose estimator to achieve shared autonomy; 3) we compare three different pose estimation methods and shed insight on their registration qualities.

II. RELATED WORK

With the introduction of accurate and low-cost 3D sensors in recent years, the pose estimation of objects in depth images has been a topic of interest in the perception and grasping community. There have been a number of discriminative approaches proposed to advance robot perception for manipulation. Collet et al. presented a discriminative approach, MOPED, to detect object and estimate object pose using iterative clustering-estimation (ICE) using multiple cameras [4]. Papazov et al. [20] used a bottom-up approach of matching the 3D object geometries using RANSAC and retrieval by hashing methods. 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

¹Zhefan Ye, Odest Chadwicke Jenkins, and Zhiqiang Sui are with Department of Computer Science and Engineering, University of Michigan, Bob and Betty Beyster Building, 2260 Hayward Street, Ann Arbor, MI, 48109-2121 zhefanye|ocj|z sui@umich.edu

²Stephen Hart is with TRACLabs Inc., 16969 N Texas AVE. Suite 300, Webster, TX 77598 swhart@traclabs.com

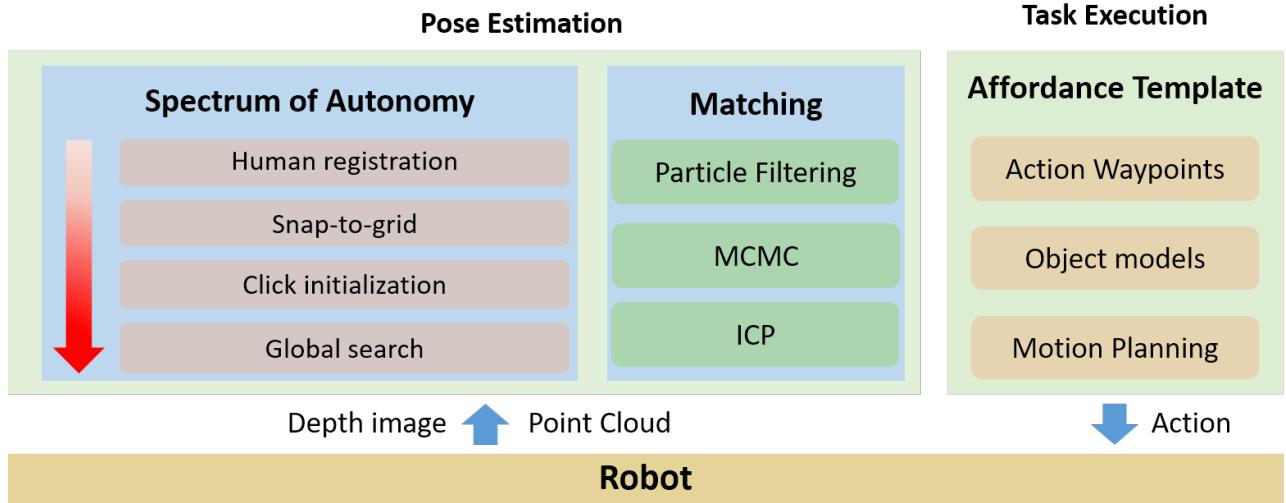


Fig. 2: The top left block illustrates our spectrum of autonomy and various pose estimators. The red arrow indicates the level of autonomy (human registration being no autonomy, whereas global search being full autonomy). The top right block shows the components of affordance template framework for executing object-centric manipulation tasks.

estimate for perception. Similar to particle filtering for robot localization, particle filtering-based methods could be used to complement these methods for extending to broader collections of robot systems. This paper builds on the Axiomatic Particle Filter proposed by Sui et al. [26] 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 [17].

Shared autonomy is a popular method to robotic operation due to its human-in-the-loop approach. Witzig et al. presented a method to grasp planning in which the user provides contextual information that the robot cannot perceive [27]. Shared autonomy grasping has been demonstrated with RViz interactive markers by Gossow et al. [10], whereas Pitzer et al. demonstrated a shared autonomy system that allows a user to help a robot manipulate its environment by performing object detection in the robots environment [22]. Parasuraman et al. first introduced the concept of levels of automation of decision and action selection in computer hardware and software system [21]. Coppin and Legras further developed a graphical representation for autonomy spectrum in swarm robots[5].

J.J. Gibson first proposed the theory of affordances, stating that organisms perceive their environment in terms of their ability to interact with it [15]. The application of affordances to robotics was introduced by Stoytchev to address how tools could be used to manipulate objects [25]. Modayil and Kuipers applied the concept of affordances to enable a mobile robot to navigate through its environment [19]. Fitzpatrick et al. demonstrated that a robot can learn affordances for pushing and grasping objects through its own exploration [9].

Sahin et al. demonstrated how affordances could be modeled probabilistically in terms of the likely effect of a

robot's behavior on its perceived sensor signals, and how those probabilities could be used to plan extended sequences of manipulation tasks [24]. Hart and Grupen showed how manipulation affordances could be acquired through intrinsically motivated reinforcement learning to form a basis of skills that could be assembled into hierarchically complex behavior [12], [13].

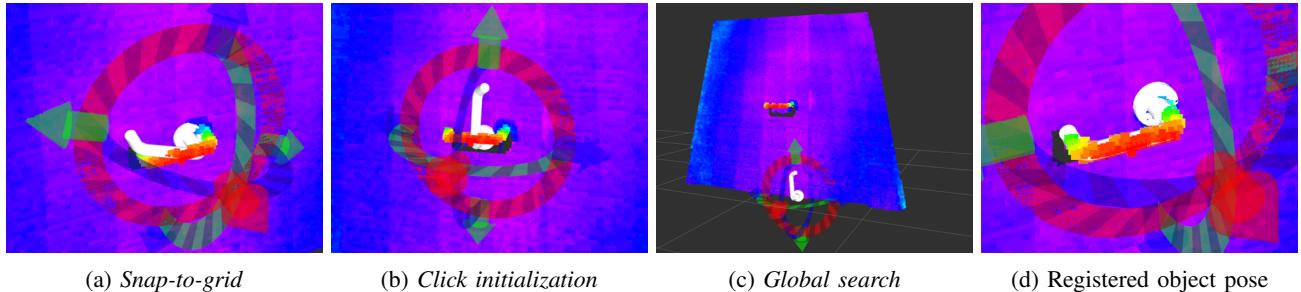
Affordance-based representations for shared autonomy have recently been used to guide humanoid robots over poor communication channels where direct teleoperation is infeasible [8], [18], [6]. Grounding task descriptions in terms of perceived objects, and the actions they afford, represents an efficient way of communicating goals to a robot, and is therefore applicable in such situations. Our approach uses the affordance template framework [11], [16] that defines tasks in terms of adjustable object geometries and end effector sequences expressed in the coordinate systems of those objects. The AT framework describes tasks in such a way that they can easily be transferred to different environmental contexts and different robot platforms.

III. APPROACH

In this section, we first describe the affordance template framework and how it is registered to point cloud observations as a pose estimation problem. Using different matching algorithms for pose estimation, we further delineate *spectrum of autonomy* that registers affordance templates utilizing different modes of initialization. The overview of our approach is illustrated in Figure 2.

A. Affordance template framework

An affordance template, $A = \{V, x, W\}$, consists of an object with geometry V in a frame defined by a 6 degree-of-freedom (DOF) pose $x \in SE(3)$, where the robot can perform action W on an object. As an implementation

Fig. 3: Using *spectrum of autonomy* to initialize or match object pose in the door handle scene

choice, action W is often an ordered sequences of end-effector waypoints W_{ee} for that serve as goals for motion planning.

To represent an affordance template in the robot frame, we express an object as v_i , which has a 3D geometry model (Figure 3) and a Cartesian pose $x = \{p_i, R_i\}$, where p expresses the position of the object and R represents the orientation of the object in $SE(3)$, with respect to a fixed frame in robot frame. For each waypoint $w_f \in W_{ee}$ at frame f , there must exist a parent object $v_i \in V_{obj}$. Likewise, w_f also has a Cartesian pose $({}^{v_i}p_{w_f}, {}^{v_i}R_{w_f})$ that express the end-effector's pose. Between w_f and w_{f+1} , a motion planner is needed to find a trajectory. Each waypoint also consists of an end-effector configuration (opened, closed, etc.) in addition to a spatial position.

While end-effector waypoints W_{ee} and objects V_{obj} are defined by a human user in RViz, affordance template strategies define only coarse spatial goals to the robot. It is dependent on the robot motion planning system to compute the trajectories necessary to guide the robot through those goals at run-time once the template has been instantiated in the environment. In order to perform such actions, the user first has to register the object with the true pose of that object in the scene (p_{g_i}, R_{g_i}) as perceived in the robot's 3D sensor data using the RViz interactive controls (Figure 3). Hence, the user has to estimate goal pose (\hat{p}_g, \hat{R}_g) and the transformation T_i^g between the goal pose and the object's current pose, which can be a labor-intensive task.

B. Spectrum of autonomy

Spectrum of autonomy is the representation of level of human involvement in an operation, such as performing object manipulation task. The following scenarios represent the increasing order of autonomy on the spectrum.

Human registration: this scenario indicates no autonomy on the spectrum. The entire registration procedure is done by a user. Through a user interface widget, such as a ROS interactive marker control, the user can move the object model to the goal pose, (\hat{p}_g, \hat{R}_g) , in the point cloud (Figure 3d).

Snap-to-grid: to alleviate the need for manual registration, the user will provide a coarse pose instead of full registration. Using the interactive tool in RViz, the user can move the object model to an intermediate pose $x_m =$

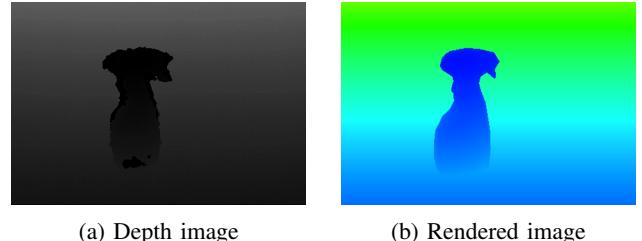


Fig. 4: Depth image and rendered image of a spray bottle.

(p_m, R_m) , which does not have to be accurate as shown in Figure 3a). Finally, a matching algorithm will perform pose estimation using the provided pose (p_m, R_m) and produce the goal pose estimation (\hat{p}_g, \hat{R}_g) (see Figure 3d).

Click initialization: to further increase the level of autonomy, the user will click on the point cloud to retrieve a position p_m . Coupled with a random orientation R_m , a matching algorithm will perform registration using the provided pose x_m (Figure 3b).

Global search: in this scenario, the entire registration process is done without any human input. Thus, given a random initial pose $x_{rand} = (p_{rand}, R_{rand})$ (Figure 3c), the goal pose (\hat{p}_g, \hat{R}_g) will be solely determined by a matching algorithm. This process indicates full autonomy.

C. Pose Estimation

We use three methods for pose estimation: particle filtering, Markov chain Monte Carlo (MCMC) and iterative closest point (ICP).

1) Particle filtering: particle filtering is a generative method that we employ to estimate the object's goal pose (\hat{p}_g, \hat{R}_g) , which generates a distribution over the possible scenes from the point of view of robot's depth camera using rendering engine and produces the most likely state estimate of an objects goal pose.

The pose at time t , $x_t = (p_t, R_t)$, is inferred from the observed states $z_{1:t}$ as a sequential Bayesian filter. This model needs to update a prior belief from time $t - 1$ using dynamic re-sampling and assess likelihood to generate a posterior belief at time t :

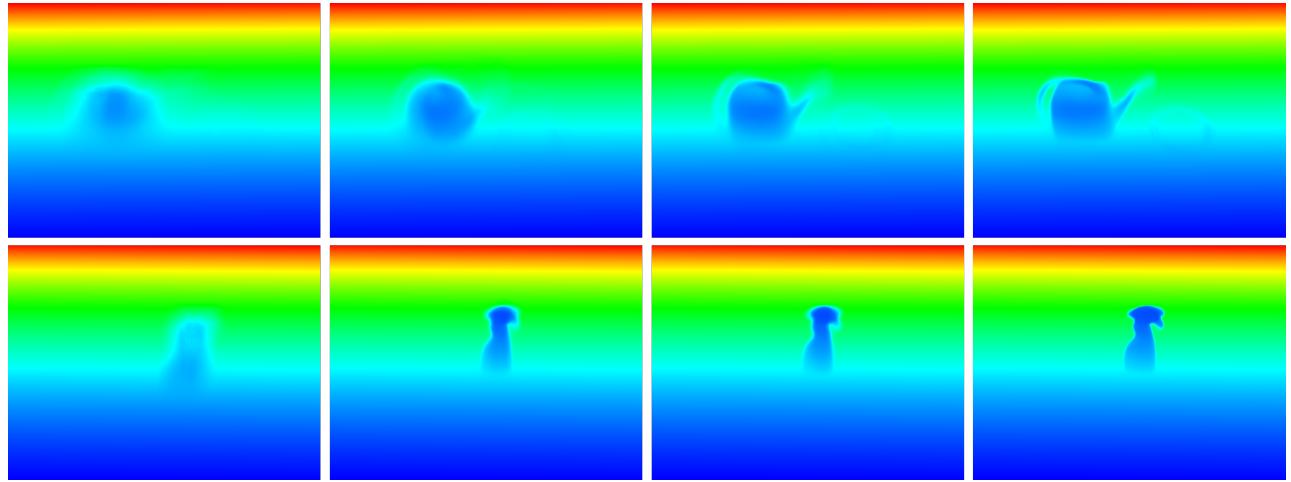


Fig. 5: Top row: spray bottle, bottom row: watering pot and bowl. Convergence to the goal poses using particle filtering. Each image represents the average rendered scene of all particles at time t . Each column corresponds to iteration 0, 300, 600 or 1000.

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

The term $p(x_t|x_{t-1}, u_{t-1})$ is used to track the robot action u_{t-1} in spite of the observed static scene. This method is inspired by [7], as those N weighted particles, $\{x_t^{(j)}, w_t^{(j)}\}_{j=1}^N$, is used to approximate this sequential Bayesian filter, thus:

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

The re-sampling of particles x_t with weight w_t is performed by importance sampling to generate a new set of scenes S , which are rendered depth images \hat{z}_t as shown in Figure 4b. The pose of each object in the scene is perturbed by normal distributions in the space of DOF. In order to measure the difference between the current observation z_t (Figure 4a) and the rendered depth image, we use the sum square of distance function $SSD(I, I')$:

$$SSD(I, I') = \sum_{(a,b) \in z} (I - I')^2, \quad (3)$$

where I is the current observation and I' is the rendered image. Therefore, the likelihood term becomes

$$p(z_t|x_t^{(j)}) = e^{-\lambda \cdot SSD(z_t, \hat{z}_t^{(j)})}, \quad (4)$$

with a constant scaling factor λ .

As the posterior distribution converges, the most likely particle \hat{x}_t produces a scene \hat{S}_t , which represents the best goal pose estimation (\hat{p}_g, \hat{R}_g) . Figure 5 illustrates the convergence process of particle filtering. Each image is the mean rendered scene $\frac{1}{N} \sum_j S_t^{(j)}$ of all particles at time t . The blurriness indicates uncertainty of the distribution over

the estimated scene. As t increases, SSD decreases and the rendered scene becomes more clear.

2) *MCMC*: MCMC [14] is another generative method that we use as an alternative to particle filtering. In particular, we use a single-site Metropolis-Hastings algorithm to estimate the goal pose (\hat{p}_g, \hat{R}_g) by approximating the target distribution $p(x|z)$ as a Markov chain. At iteration t , x_t is generated by a proposal distribution $q(x_t|x_{t-1})$. In our case,

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t|x_{t-1}, \Sigma_{x_{t-1}}), \quad (5)$$

where $\Sigma_{x_{t-1}}$ is the covariance of a normal distribution that centered on the pose of the previous sample $x_{t-1} = (p_{t-1}, R_{t-1})$. Since the proposal distribution is symmetric, the new sample can be either accepted or rejected by the acceptance probability,

$$r(x_{t-1}, x_t) = \min(1, \frac{p^*(x_t)}{p^*(x_{t-1})}), \quad (6)$$

where p^* is specified in Equation 4.

Followed by generating N samples $X^* = \{x^{(i)}\}_{i=1}^N$, the goal pose (\hat{p}_g, \hat{R}_g) can be retrieved by finding the \hat{x} with maximum likelihood p^* ,

$$\hat{x} = \arg \max_{X^{*(i)}} p^*(X^{*(i)}). \quad (7)$$

3) *ICP*: contrary to the previous two methods, ICP [3] is a discriminative matching algorithm for aligning two templates. In our case, ICP is used to find the transformation T_{init}^g between the initial pose (p_{init}, R_{init}) and the goal pose (p_g, R_g) by matching the point cloud generated by the object geometry model v and a target point cloud b , the energy function for T_{init}^g becomes

$$E(R, t) = \frac{1}{N_v} \sum_{i=1}^{N_v} \| b_i - R(v_i) - t \|^2, \quad (8)$$

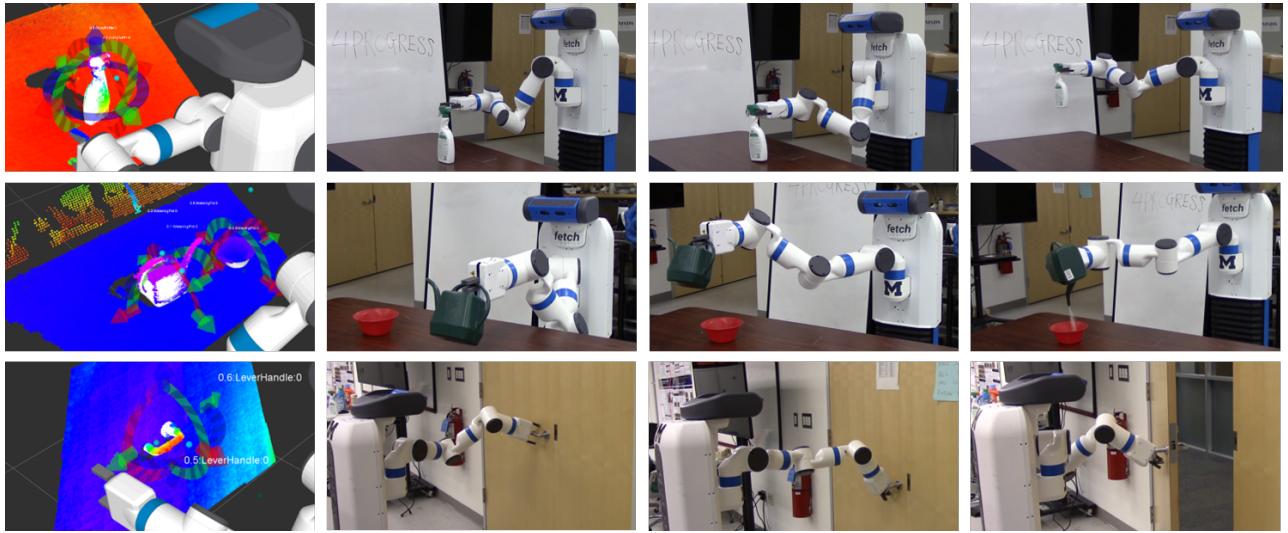


Fig. 6: First row: *using a spray*; second row: *pouring*; third row: *door opening*. The first column shows the object models and task environments in RViz. The rest columns show each task's snapshots and real environments.

where R and t is rotation and translation; b_i and v_i are corresponding points. By minimizing the sum of the squared error $E(R, t)$, we can obtain the transformation $\hat{T}_{init}^g = (\hat{R}_{init}^g, \hat{t}_{init}^g)$,

$$\hat{T}_{init}^g = \arg \min_{R, t} E(R, t). \quad (9)$$

Thus, the estimated goal pose $\hat{x} = \hat{T}_{init}^g * x_{init}$.

IV. RESULTS

A. Implementation

Our implementation used the Robot Operating System (ROS) [23]. The affordance template software is openly available as a ROS package¹, which can program, adjust and execute robot applications in the ROS RViz environment [11]. All communication between pose estimation and the affordance template software was performed using ROS services. For motion planning between end-effector waypoints, we utilized TracIK planner [2], which is an inverse kinematics solver concurrently runs both improved KDL- and SQP-based methods. We used PCL library² for our ICP implementation. Our experiments were performed using a Fetch³ mobile manipulator with a seven degree of freedom arm and a pan-tilt head equipped with a RGB-D camera (Figure 1).

When using particle filtering, the number of the particles were set to 500. For MCMC matching, we ran 5000 iterations.

B. Experiments

In this section, we demonstrate pose estimation for registration of affordance templates as a viable approach to

spectrum of autonomy for object manipulation tasks. We additionally present a pose estimation comparison among our matching algorithms against human baseline.

1) *Task scenes*: We further defined three affordance templates to accomplish three different tasks in three scenes:

- *Using a spray*: the robot must pick up the object *spray bottle*, move the nozzle to point to a surface, and then squeeze the trigger to spray that surface (Figure 6 row one),
- *Pouring*: the robot must pick up the object *watering pot*, move it over another object *bowl*, and pour its contents into the bowl (Figure 6 row two), and
- *Door opening*: robot must grab a *door handle*, turn it, and pull open the door (Figure 6 row three).

Figure 6 further depicts the robot executing these three tasks in RViz (the first two columns), and in the real world (the third column).

We assume that objects are standing on top of the table surface and the door handle is attached to the door in a standard location. Thus, our pose estimator allows three DOF for each object.

2) *Ground truth affordance registration*: In order to obtain the ground truth pose for each object in each scene, the user manually registered each object's affordance using ROS interactive marker control (Figure 3d). The registered pose is expressed as $(x_{gt}, y_{gt}, Yaw_{gt})$ for *using a spray* and *pouring* scenes, while $(y_{gt}, z_{gt}, Roll_{gt})$ for *door opening* scene.

3) *Spectrum of autonomy comparison*: For each task, we conducted *snap-to-grid*, *click initialization* and *global search* to perform pose estimation and compare the result against human registered ground truth. We denote *snap-to-grid* as *STG*, *click initialization* as *CI* and *global search* as *GS*. For each scenario, we performed 50 trials and collected the results. When conducting *STG*, 50 initial poses (position and orientation) were given. For *CI*, only 50 initial positions were

¹<http://traclabs.com/projects/affordance-templates/>

²<http://pointclouds.org/>

³<http://fetchrobotics.com/research/>

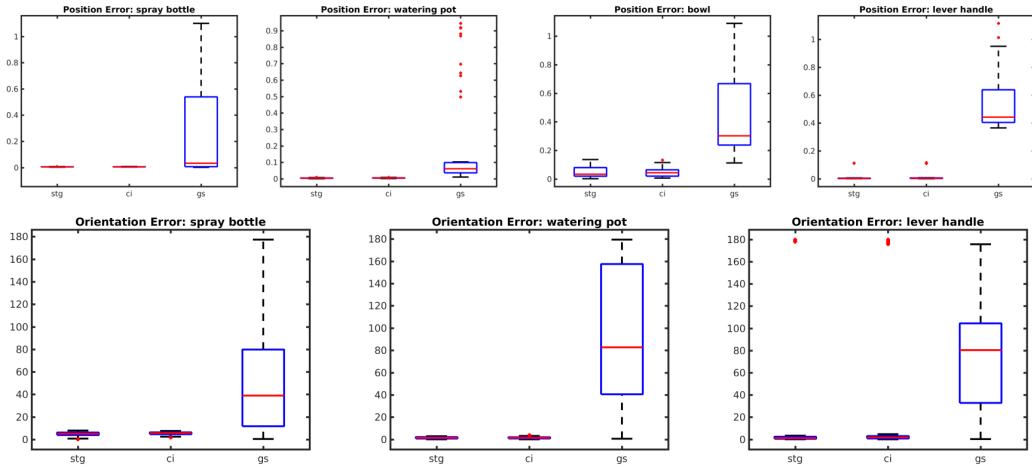


Fig. 7: Position error (top row) and orientation error (bottom row) for *STG*, *CI* and *GS* and each scenario. The metric for position is the Euclidean distance (meter) between predicted pose and ground truth pose and the metric for orientation is degree. For spray bottle scene and watering pot/bowl scene, the orientation is yaw, and for door handle scene, the orientation is roll. Note that orientation for *bowl* is not included due to its symmetrical geometry.

given while orientations were random. There's no need to initialize for *GS*.

First, we compared registration quality using different scenarios on the spectrum. We chose particle filtering as our pose estimation method.

Note that both *watering pot* and *bowl* have to be estimated for *pouring* scene using our pose estimation algorithms. However, we did not account for the orientation error for *bowl* since it is symmetric along the z-axis.

Figure 7 shows the box-plots of the position error and orientation error for each object in each task. Across the object categories, *STG* and *CI* consistently outperform *GS*, which indicates that user's initialization did help pose estimation methods better find the correct poses. Figure 8 further shows the percentage of correct pose given position error thresholds — 0.005, 0.01 and 0.05 meters. *STG* outperforms *CI* in the *door handle* scene under all three thresholds, while perform similarly in the other two scenes. That is because the particle filtering method may rotate the *door handle* object model by 180° from the correct pose to match the scene due to its quasi-symmetric nature. On the other hand, *watering pot* and *spray bottle* are cylindrical-like objects when standing on the tabletop; therefore, the orientation initialization has limited effect.

4) Pose estimation methods comparison: We chose *STG* as our scenario for pose estimation methods comparison. Given the same initial poses, we ran particle filtering, MCMC and ICP on all three scenes and compared results. As shown in Figure 9, particle filtering outperforms the other two methods for *watering pot* and *door handle*. Both particle filtering and MCMC are generative methods that use a rendering engine as a mean to generate samples. Particle filtering is more adaptive to this circumstance due to the number of particles it can sample within one iteration. ICP, on the other hand, is a discriminative method. Since the

depth camera can only capture a small portion of the object's geometry, it may be difficult to fit the entire object model to the scene, especially when object is small, whereas a rendering engine-based generative method, such as particle filtering, is able to fully explore the local search space given enough samples. Therefore, ICP performs the worst in this setting.

5) Affordance template tasks: Once a predicted pose is given, affordance template will perform the action associated with each waypoint. However, the completion of the task depends on many factors. For instance, motion planning between two waypoints may fail due to object pose or end-effector pose. Thus we only consider if the robot is able to grasp the object successfully regardless of the completion of the task.

The success rate of using a spray is 90%, pouring is 70% and door opening is 100%. The reason why grasping is not successfully every time is because of two reasons. First, the pouring task, for instance, is more challenging compared to the other two tasks since the lower half of the end-effector (gripper) has to be threaded through the pot handle as shown in Figure ?? (second row, the first image). Therefore this process has low tolerance of trajectory deviation. Thus, the design of pre-grasp gripper pose is crucial to the success of task execution. Secondly, the robot arm of Fetch may sometimes deviate from the designed waypoints before grasping, which would cause grasping failure. The deviation of the robot arm is random; thus grasping may fail even if the affordance registration is perfect.

V. CONCLUSION

We have presented a human-in-the-loop approach to affordance registration for robot manipulation. Through the use of spectrum of autonomy, we demonstrate that we can achieve shared autonomy by combining user initialization

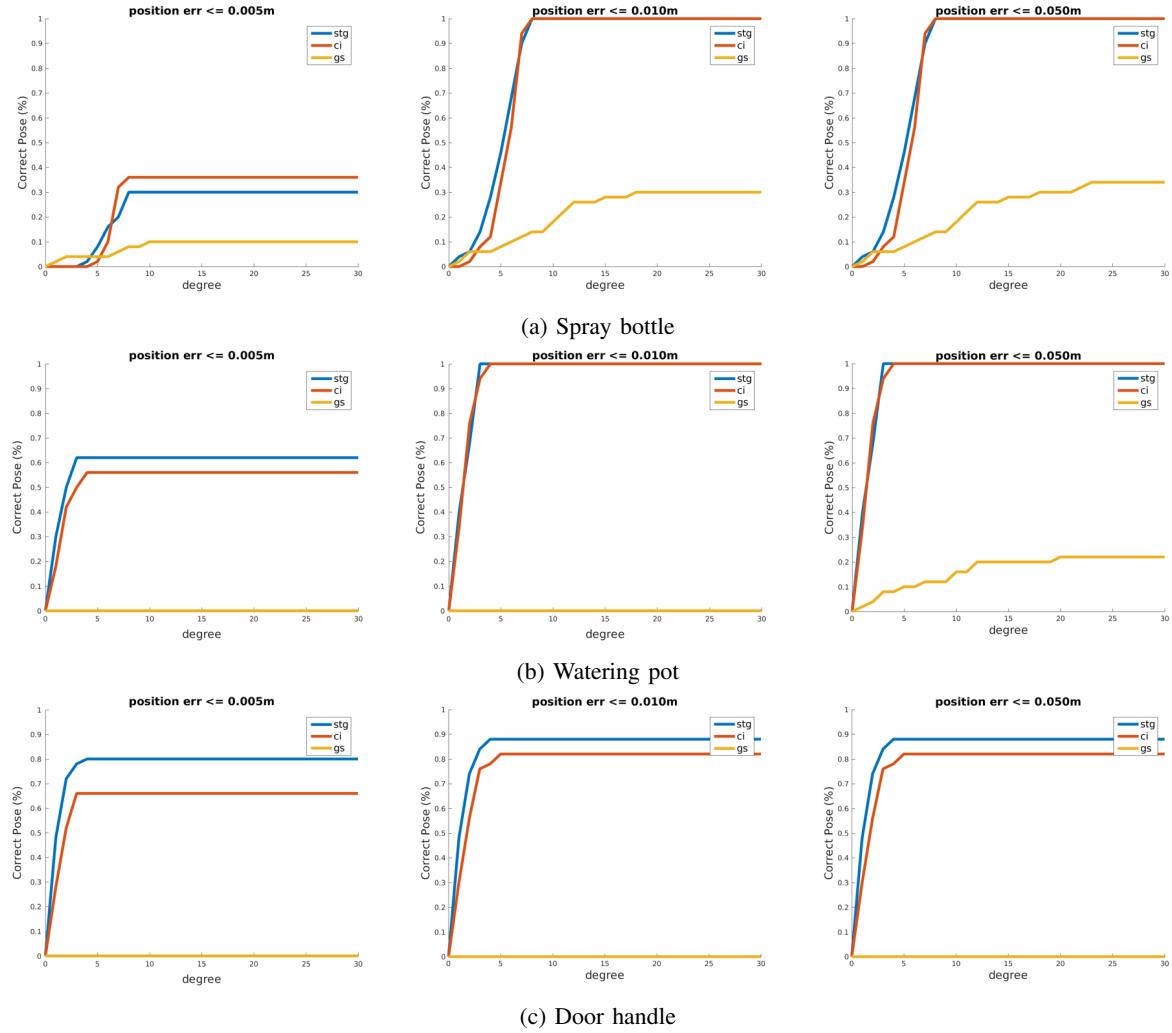


Fig. 8: Spectrum of autonomy comparison using different initialization scenarios

and generative pose estimator. We posit that perception of objects and their affordances is the critical bottleneck for autonomous execution of robot manipulation, which we offer one step towards this goal. However, successful affordance registration does not necessarily mean successful task execution since task completion also relies on other factors, such as accurate trajectory following of robotic arm. As future work, that state estimation framework can be extended to accommodate multiple objects, such as for the Axiomatic Particle Filter [26], for goal-directed manipulation over general affordances.

REFERENCES

- [1] J. Badger, D. Gooding, K. Ensley, K. Hambuchen, and A. Thackston. Ros in space: A case study on robonaut 2. In *Robot Operating System (ROS)*, pages 343–373. Springer, 2016.
- [2] P. Beeson and B. Ames. TRAC-IK: An open-source library for improved solving of generic inverse kinematics. In *Proceedings of the IEEE RAS Humanoids Conference*, Seoul, Korea, November 2015.
- [3] P. J. Besl and N. D. McKay. Method for registration of 3-d shapes. In *Robotics-DL tentative*, pages 586–606. International Society for Optics and Photonics, 1992.
- [4] A. Collet, M. Martinez, and S. S. Srinivasa. The moped framework: Object recognition and pose estimation for manipulation. *The International Journal of Robotics Research*, page 0278364911401765, 2011.
- [5] G. Coppin and F. Legras. Autonomy spectrum and performance perception issues in swarm supervisory control. *Proceedings of the IEEE*, 100(3):590–603, 2012.
- [6] M. DeDonato, V. Dimitrov, R. Du, R. Giovacchini, K. Knoedler, X. Long, F. Polido, M. A. Gennert, T. Padr, S. Feng, H. Moriguchi, E. Whitman, X. Xinjilefu, and C. G. Atkeson. Human-in-the-loop control of a humanoid robot for disaster response: A report from the darpa robotics challenge trials. *Journal of Field Robotics*, 32(2):275–292, 2015.
- [7] 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.
- [8] M. Fallon, S. Kuindersma, S. Karumanchi, M. Antone, T. Schneider, H. Dai, C. Perez D’Arpino, R. Deits, M. DiCicco, D. Fourie, T. Koolen, P. Marion, M. Posa, A. Valenzuela, K.-T. Yu, J. Shah, K. Iagnemma, R. Tedrake, and S. Teller. Affordance-based Perception and Whole-body Planning in the DARPA Robotics Challenge. Technical Report MIT-CSAIL-TR-2014-003, MIT, Cambridge, MA, 2014.
- [9] P. Fitzpatrick, G. Metta, L. Natale, S. Rao, and G. Sandini. Learning about objects through action-initial steps towards artificial cognition. In *Robotics and Automation, 2003. Proceedings. ICRA’03. IEEE International Conference on*, volume 3, pages 3140–3145. IEEE, 2003.
- [10] D. Gossow, A. Leeper, D. Hershberger, and M. Ciocarlie. Interactive markers: 3-d user interfaces for ros applications [ros topics]. *IEEE Robotics & Automation Magazine*, 18(4):14–15, 2011.

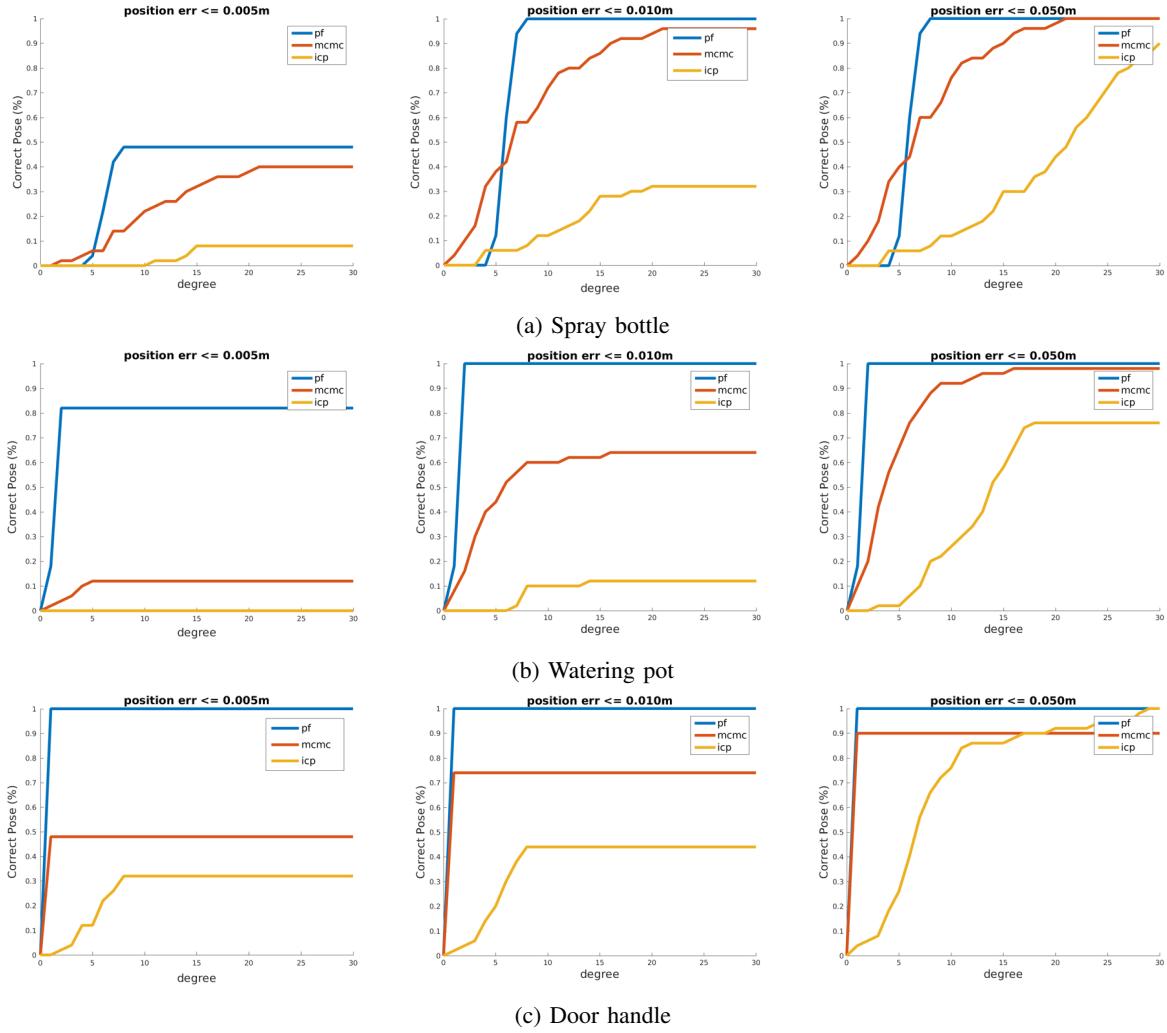


Fig. 9: Pose estimation methods comparison. All three methods use the same STG initialization.

- [11] 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.
- [12] S. Hart and R. Grupen. Intrinsically motivated affordance learning. 2009.
- [13] S. Hart and R. Grupen. Intrinsically motivated affordance discovery and modeling. In *Intrinsically Motivated Learning in Natural and Artificial Systems*, pages 279–300. Springer, 2013.
- [14] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.
- [15] J. J. Gibson. The theory of affordances. *Perceiving, acting and knowing: toward an ecological psychology*, pages 67–82, 1977.
- [16] J. James, Y. Weng, S. Hart, P. Beeson, and R. Burridge. Prophetic goal-space planning for human-in-the-loop mobile manipulation. In *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, pages 1185–1192. IEEE, 2015.
- [17] L. P. Kaelbling and T. Lozano-Pérez. Integrated task and motion planning in belief space. *The International Journal of Robotics Research*, page 0278364911436019, 2012.
- [18] T. Koolen and J. Smith. Summary of Team IHMC’s Virtual Robotics Challenge Entry. In *IEEE-RAS International Conference on Humanoid Robots*, Atlanta, Georgia, 2013. IEEE-RAS.
- [19] J. Modayil and B. Kuipers. Autonomous development of a grounded object ontology by a learning robot. 2007.
- [20] 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.
- [21] R. Parasuraman, T. B. Sheridan, and C. D. Wickens. A model for types and levels of human interaction with automation. *IEEE Transactions on systems, man, and cybernetics-Part A: Systems and Humans*, 30(3):286–297, 2000.
- [22] B. Pitzer, M. Styer, C. Bersch, C. DuHadway, and J. Becker. Towards perceptual shared autonomy for robotic mobile manipulation. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 6245–6251. IEEE, 2011.
- [23] M. Quigley, J. Faust, T. Foote, and J. Leibs. Ros: an open-source robot operating system.
- [24] E. Şahin, M. Çakmak, M. R. Doğar, E. Uğur, and G. Üçoluk. To afford or not to afford: A new formalization of affordances toward affordance-based robot control. *Adaptive Behavior*, 15(4):447–472, 2007.
- [25] A. Stoytchev. Toward learning the binding affordances of objects: A behavior-grounded approach. 2005.
- [26] 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.
- [27] T. Witzig, J. M. Zöllner, D. Pangercic, S. Osentoski, R. Jäkel, and R. Dillmann. Context aware shared autonomy for robotic manipulation tasks. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5686–5693. IEEE, 2013.