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

Zhefan Ye
University of Michigan
zhefanye@umich.edu

Odest Chadwicke Jenkins
University of Michigan
ocj@umich.edu

Zhiqiang Sui
University of Michigan
zsui@umich.edu

Stephen Hart
TRACLabs Inc.
swhart@traclabs.com

***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 combination of human-in-the-loop particle filtering-based pose estimator and user initialization to achieve shared autonomy, followed by manipulation tasks execution using affordance template framework. We further compare the performance of different pose estimation methods and demonstrate the efficacy of our approach in three scenes.**

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* [3] to enable mobile manipulation and humanoid robot control, such as for robots in the DARPA Robotics Challenge [4] 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 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.

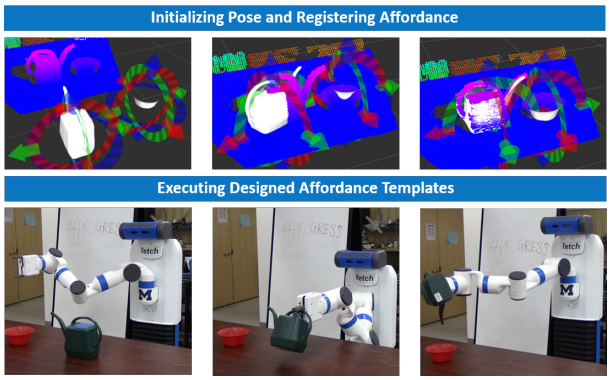

Fig. 1: **Top row**: user gives a rough initial pose estimation, after which a pose estimation method refines the pose to register the affordance template. **Bottom row**: affordance template framework performs a series of designed waypoints to perform a task.

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 pose estimation methods and shed insight on their registration qualities.

II. Related Work

Shared autonomy is a popular method for 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 [6]. Shared autonomy grasping has been demonstrated with RViz interactive markers by Gossow et al. [2].

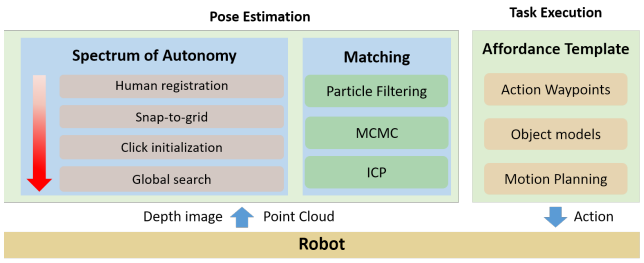

Fig. 2: The top left block illustrates our spectrum of autonomy and various pose estimators. The red arrow indicates the level of autonomy (human registration being no autonomy, whereas global search being full autonomy). The top right block shows the components of affordance template framework for executing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

This paper builds on the Axiomatic Particle Filter proposed by Sui et al. [5] for generative scene estimation to perform goal-directed manipulation.

Our approach uses the affordance template framework [3, 4] that defines tasks in terms of adjustable object geometries and end effector sequences expressed in the coordinate systems of those objects. The affordance template 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 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 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.

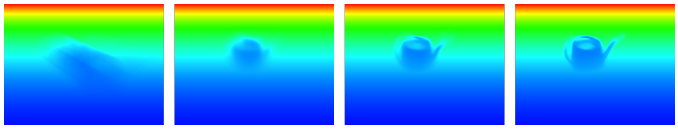

Fig. 3: Convergence to the goal poses using PF of watering and bowl scene. Each image represents the mean image of the rendered scenes of all particles at time t. Columns one to four correspond to iteration 0, 300, 600 or 1000. As the images show, particles gradually converge to a final state.

The affordance template software is openly available as a ROS package[1]

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.

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 $x_m = (p_m, R_m)$, which does not have to be accurate. Finally, a matching algorithm will perform pose estimation using the provided pose (p_m, R_m) and produce the goal pose estimation $(\hat{p}_g, \hat{R}_g)$.

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

Global search: in this scenario, the entire registration process is done without any human input. Thus, given a random initial pose $x_{rand} = (p_{rand}, R_{rand})$, 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 (PF), Markov chain Monte Carlo (MCMC) and iterative closest point (ICP).

1) Particle filtering: PF 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, as those

[1] http://traclabs.com/projects/affordance-templates/

N weighted particles, $\{x_t^{(j)}, w_t^{(j)}\}_{j=1}^{N}$, is used to approximate this sequential Bayesian filter, thus:

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

The re-sampling of particles x_t with weight w_t is performed by importance sampling to generate a new set of scenes $\mathbf{S}$, which are rendered depth images $\hat{z}_t$. 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 and the rendered depth image, we use the sum square of distance function $\mathrm{SSD}(I, I')$:

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

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

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

with a constant scaling factor λ.

As the posterior distribution converges, the most likely particle $\hat{x}_t$ produces a scene $\hat{\mathbf{S}}_t$, which represents the best goal pose estimation $(\hat{p}_g, \hat{R}_g)$. Figure 3 illustrates the convergence process of PF. Each image is the mean rendered scene $\frac{1}{N}\sum_j \mathbf{S}_t^{(j)}$ 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. We use 500 particles for all experiments.

2) MCMC: MCMC is another generative method that we use as an alternative to PF. 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.

3) ICP: contrary to the previous two methods, ICP 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 and point cloud in the scene. We use PCL library[2] for our ICP implementation.

IV. RESULTS

A. 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.

[2]http://pointclouds.org/

1) Task scene: 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,
- *Pouring*: the robot must pick up the object *waterpot*, move it over another object *bowl*, and pour its contents into the bowl (Figure 4), and
- *Door opening*: robot must grab a *door handle*, turn it, and pull open the door.

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. 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 conduct *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 perform 50 trials and collect the results. When conducting *STG*, 50 initial poses (position and orientation) were given. For *CI*, only 50 initial positions are given while orientations are random. There's no need to initialize for *GS*.

First, we compare registration quality using different scenarios on the spectrum. We choose PF as our pose estimation method.

Figure 5 further shows the percentage of correct pose given position error thresholds — 0.005, 0.01 and 0.05 meters. Both *STG* and *CI* outperform *GS* since user reduce the search space for our pose estimators and this would dramatically improve pose registration quality.

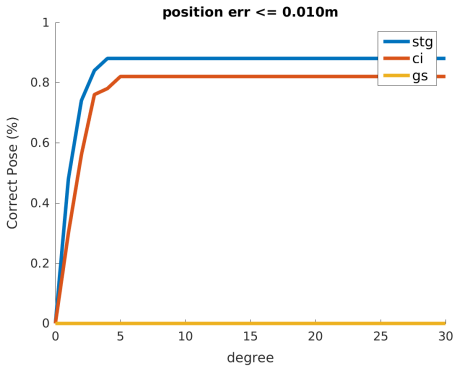

Fig. 5: Spectrum of autonomy comparison of waterpot scene using different initialization scenarios

4) Pose estimation methods comparison: We choose *STG* as our scenario for pose estimation methods comparison.

Fig. 4: *Pouring* task; the first column shows the object models and task environments in RViz. The rest columns show a task's snapshots and real environments.

Given the same initial poses, we ran PF, MCMC and ICP on all three scenes and compared results. As shown in Figure 6, PF outperforms the other two methods for *waterpot* and *door handle*. Both PF and MCMC are generative methods that use a rendering engine as a mean to generate samples. PF 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 PF, is able to fully explore the local search space given enough samples.

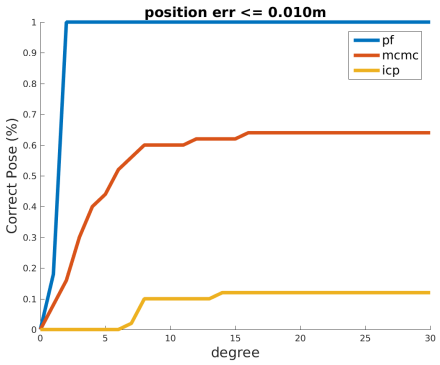

Fig. 6: Pose estimation methods comparison of waterpot scene. All three methods use the same *STG* initialization.

5) Affordance template tasks: Once a predicted pose is given, affordance templates 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 related task (e.g. *using a spray*) 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 4. Therefore, this process has low tolerance of trajectory deviation. Therefore, the design of pre-grasp gripper pose is crucial to the success of task execution. Secondly, the arm of the Fetch robot may sometimes miss the designed waypoints, and cause potential failures in various stages.

V. CONCLUSION

We present a human-in-the-loop approach to affordance registration for robot manipulation. We demonstrate that shared autonomy can be achieved by the combination of user initialization and pose estimator. Furthermore, we posit that the poses of objects and their affordances is a critical bottleneck for autonomous execution of robot manipulation. Hence, by taking advantage of particle filtering-based pose estimator, we offer one step closer 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 grasp pose and trajectory following of the robotic arm.

REFERENCES

[1] Julia Badger, Dustin Gooding, Kody Ensley, Kimberly Hambuchen, and Allison Thackston. Ros in space: A case study on robonaut 2. In *Robot Operating System (ROS)*, pages 343–373. Springer, 2016.

[2] David Gossow, Adam Leeper, Dave Hershberger, and Matei Ciocarlie. Interactive markers: 3-d user interfaces for ros applications [ros topics]. *IEEE Robotics & Automation Magazine*, 18(4):14–15, 2011.

[3] Stephen Hart, Paul Dinh, and Kimberly 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.

[4] Joshua James, Yifan Weng, Stephen Hart, Patrick Beeson, and Robert 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.

[5] Zhiqiang Sui, Odest Chadwicke Jenkins, and Karthik 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.

[6] Thomas Witzig, J Marius Zöllner, Dejan Pangercic, Sarah Osentoski, Rainer Jäkel, and Rüdiger 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.