Extending Affordance Templates to Support Remote Robot Activities

Michael W. Lanighan Kenji Brameld

***Abstract*— The Affordance Template task primitive framework has been garnering increased interest from researchers in the field as a conceptual format to define robot manipulation tasks within a supervised/shared autonomy paradigm. Affordance Templates provide graphical utilities to facilitate human-in-the-loop operation and can also support robot autonomy. This paper presents several extensions to the Affordance Template framework that (1) expand the types of behaviors that can be encoded beyond manipulation to include inspection and navigation, (2) facilitate their use in multi-agent tasks, and (3) outline how roles encoded in Affordance Templates can be dynamically assigned at run-time to match the capabilities of deployed systems. We formally introduce these extensions and provide example demonstrations that leverage these new definitions.**

I. Introduction

Effective control of remote assets is a longstanding problem in robotics. Due to bandwidth and time-delay limitations, direct control of such assets becomes more difficult as the distance increases. A conceptual framework known as Affordance Templates (AT) [1], [2] has been gathering increased interest from researchers in the field as a framework to define robot manipulation tasks within a supervised/shared autonomy paradigm [3]. ATs allow an operator to define high-level decisions when appropriate while supporting manual teleoperation in pathological situations that arise from *uncertainty* in unstructured environments, sensing, and actuation.

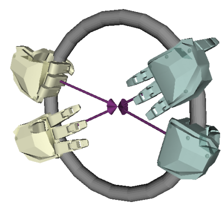
To date, ATs have been restricted to manipulation tasks on a single robot. There is increasing interest in maturing tools to control heterogeneous teams of robots for tasks beyond pure manipulation. To address this, this paper outlines several extensions to the AT framework [2] to (1) expand the types of behaviors that can be encoded beyond manipulation to include inspection and navigation, (2) facilitate their use in multi-agent tasks, and (3) outline how roles encoded in ATs can be dynamically assigned at run-time to match the capabilities of deployed systems.

II. Related Work

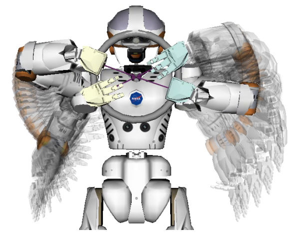
A. Affordance Templates

ATs were originally designed to support the supervisory control of humanoids such as NASA's Valkyrie and Boston Dynamics' Atlas for remote field operations [4], [5]. Specifically, ATs define trajectories of adjustable spatial waypoint goals in the coordinate frames of virtual objects. Waypoint goals include path or motion parameters (constraints, tolerances, gains, etc.), and semantic references to

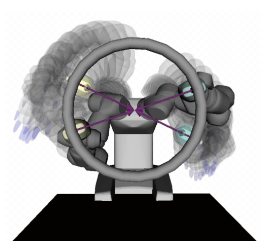
Authors are with TRACLabs Inc. 1331 Gemini Street; Houston, TX 77058. Corresponding Author: `mlanighan@traclabs.com`. This work was supported by NASA contract 80NSSC21C0622.



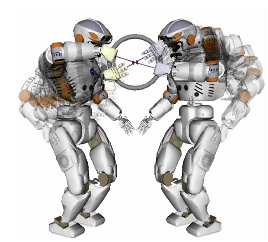
(a) Bimanual Wheel Turning AT instantiated for NASA Valkyrie



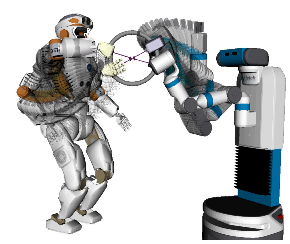
(b) Realizing AT motions on NASA Valkyrie



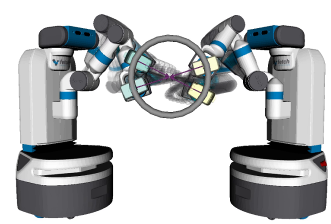
(c) Realizing AT motions remapped to an ABB YuMi



(d) Realizing AT motions remapped to two NASA Valkyries



(e) Realizing AT motions remapped to NASA Valkyrie and Zebra Fetch robot



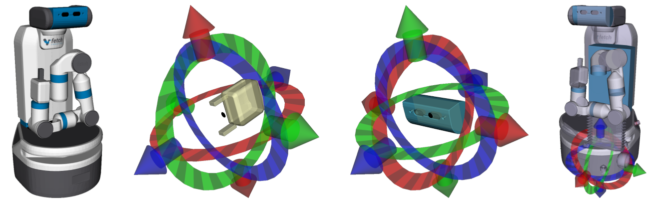
(f) Realizing AT motions remapped to two Zebra Fetch robots

Fig. 1. (a) AT for two-handed counter-clockwise turn of a wheel. (b)-(f) the same template demonstrated on several single and multi-agent homogeneous and heterogeneous teams. Planned motions are shown via transparent ghost "rasters" which show the planned motion ending with opaque robots in the goal configuration. In every demonstration, roles are remapped to available chains. No modifications are made beyond repositioning the template to be within each system's manipulable workspace.

end effector configurations ("open", "closed", etc.) that affect how the robot moves and what it does at each step. Since its original introduction, a number of efforts have been made to investigate how autonomous algorithms can alleviate the burden on developers and operators when instantiating ATs for a given run-time context and to investigate how to encode additional robot behaviors in the AT waypoint specification.

Initial extensions to ATs moved beyond simple waypoint sequences. Researchers incorporated mobile robot stance goals [6], defined manipulation as potential fields to handle constraints (*Affordance Wayfields*) [7], and added non-spatial goals for compliant interaction (*Affordance Primitives*) [8]. These methods, however, still focused on single-agent manipulation.

Subsequent work has focused on reducing operator burden and improving generalization. To automate template registration, some approaches bootstrap real-time alignment to point clouds using Monte Carlo Localization [9] or articulated object structure [10]. Others improve matching by decomposing objects into independently perceived parts (*Affordance Coordinate Frames*) [11]. To generalize across different robots and contexts, Hart *et al.* integrated autonomous grasping and stance generation to ensure waypoint reachability [2], while Calvert *et al.* used Coactive Design to generate templates on-the-fly, reducing the need for manual, context-specific tuning [12].



(a) Fetch Robot (b) $\tau = manipulation$ (c) $\tau = sensor$ (d) $\tau = mobility$

Fig. 2. (a) Simulated Fetch robot used to demonstrate waypoint types. (b)-(d) The three waypoint types as realized by the Fetch mobile manipulator. (b) Manipulation waypoint displaying the end-effector of the kinematic chain of the arm. (c) Sensor waypoint displaying the termination of the chain from the base of the robot to the head. (d) Mobility waypoint. Although defined from the base to the reference frame, when used in planning the chain is inverted such that the termination of the inverted virtual chain contains the entire robot.

III. Extending Affordance Templates for Heterogeneous Robot Activities

A. Defining the Aggregate Multi-agent System

We model a robot as a set of kinematic chains, C, one for each end-effector and imager[1]. Each chain's tip frame follows a standard coordinate convention. To create a unified model, we introduce a virtual 6-DOF chain, $c_{virtual}$, connecting a global reference frame to the robot's base. This virtual chain's active degrees of freedom (DOFs) match the robot's mobility (e.g., 6 for a free-flyer, 3 for a planar base, or 0 for a static arm), thereby encoding the robot's global pose.

A multi-agent system of n robots is then the union of all individual chains, $S = \bigcup_{i=1}^{n} C_i$. By including a virtual chain for each robot, the entire system can be treated as a single, high-DOF kinematic tree rooted at the global reference frame.

B. Extended Affordance Template Definition

Extending the definitions from Hart *et al.* [2], a multi-agent AT is defined as follows: Each AT in a library, $a \in A$, consists of a set of virtual objects

$$O = \{o_1, \ \ldots, \ o_\xi\} \tag{1}$$

and a set of waypoint/keyframe trajectories

$$T = \{t_1, \ \ldots, \ t_m\} \tag{2}$$

such that $a = \langle O, T \rangle$. Each object o_i has a coordinate frame H_i and geometry model M_i such that $o_i = \langle H_i, M_i \rangle$. Each trajectory specifies a temporal ordering of goals for a set of k kinematic chains. For each waypoint trajectory $t_j \in [1, m]$,

$$t_j \quad = \quad (s_1, \ \ldots, \ s_\phi) \tag{3}$$

Each temporal step s_ζ, $\zeta \in [1, \phi]$, contains goals for up to k kinematic chains,

$$s_\zeta = \{wp_1, \ \ldots, \ wp_k\} \tag{4}$$

[1] "Imager" is used here for any depth, image, or range sensor.

The waypoint configuration for the g^{th} chain contained in step s_ζ is defined as

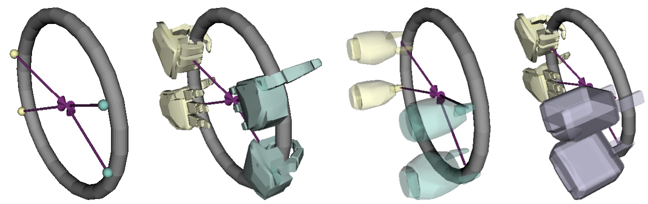
$$wp_g = \langle H_i, \ \tau, \ p \rangle. \tag{5}$$

H_i is a Cartesian goal defined in the reference frame of object $o_i \in O$. τ defines the *type* of goal at the waypoint such that $\tau \in \{manipulator, \ sensor, \ mobility\}$ (see Fig. 2 for an example). The addition of *type* information for each goal allows templates to encode tasks that were previously unattainable, such as non-contact tasks, including inspection. Specifying trajectories that sequence mobility, manipulation, and sensor goals can encode actions that navigate to target objects, manipulate them, and verify that task outcomes have been achieved. Trajectories t_j need not define waypoint goals wp_g for all k chains at each step. If a goal for chain g is not defined at a step, it maintains its previous configuration, possibly the configuration of the chain at the trajectory start.

IV. Dynamic Role Assignment

AT trajectories are defined for abstract roles that must be mapped to a robot's physical kinematic chains. This process has traditionally required developers to define these mappings a priori—a rigid approach that limits the reuse of ATs and is untenable for multi-agent systems due to the combinatorial complexity of potential assignments.

We propose replacing these static, pre-defined mappings with dynamic, run-time role assignment. In our approach, developers still enumerate a system's capabilities but now only declare the type for each chain (e.g., manipulator, sensor). This configuration naturally extends to multi-agent systems by simply creating a list of all available chains. We frame this dynamic assignment as a bipartite matching problem. We construct a graph $G = (U, V, E)$ where U is the set of available system chains and V is the set of required AT roles. An edge exists between a chain $u \in U$ and a role $v \in V$ if their types match. This edge can be weighted by a cost, w_{uv}, that captures factors such as kinematic displacement, estimated execution time, or whether a chain is already tasked.

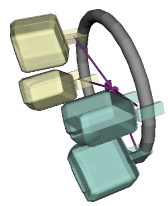


(a) Uninitialized Bimanual Wheel Turning AT

(b) Template remapped for NASA Valkyrie

(c) Template remapped for ABB YuMi

(d) Template remapped for team of NASA Valkyrie and Zebra Fetch



(e) Template remapped for team of two Zebra Fetch

Fig. 3. (a) The abstract, uninitialized AT for a bimanual wheel turn. (b)-(e) The same template after being dynamically remapped to the single and multi-agent systems of Fig. 1. Waypoints for each unique chain are colored according to a predefined color palette [16] in order to better visually aggregate goals for the same chain.

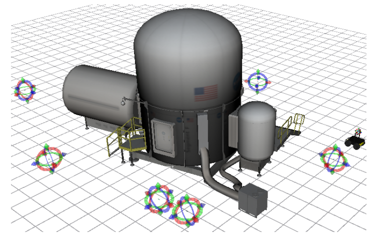
Finding a maximum matching in G therefore solves two problems: it assigns roles to realize the AT and determines if the task is feasible (i.e., if all required roles can be filled). This matching can be found efficiently in polynomial time [13], [14], [15]. Fig. 3 shows the effect of this remapping on the template from Fig. 1(a) for the systems in Fig. 1. After remapping, the goals are initialized for the assigned chains, matching the visualized goals to the specific system.

V. Demonstrations

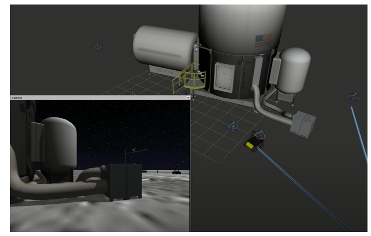
All demonstrations are performed using TRACLabs' CRAFTSMAN planning and control framework [17]. However, the presented extensions to the AT definition are agnostic of the specific planning framework utilized. The following demonstrations highlight the new capabilities that can now be encoded as an AT.

A. Remapping roles for deployed systems

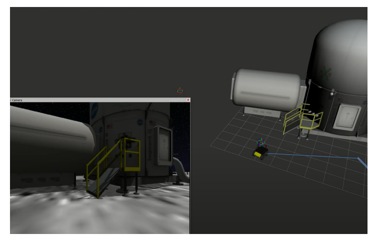
Inspired by the original wheel turning task featured in Hart *et al.* [1], we demonstrate the flexibility of the updated AT definitions by presenting a similar valve turning task deployed on several single and multi-agent teams, as shown in Fig. 1. Fig. 1(b)-(f) illustrate the planned motions for each system after remapping roles. These various systems did not require careful handling of roles by the developer *a priori*, as dynamic role assignment (See Section IV) was used to map AT roles to the roles available in the deployed system. The template was repositioned to place it in each system's manipulable workspace; however, it was otherwise unmodified for the various systems.



(a)



(b)



(c)

Fig. 4. Inspection of a simulated lunar habitat encoded as an AT. (a) The template encodes important viewing geometries to verify habitat state. (b)-(c) When executed, a supervisor can inspect images taken at waypoints to verify habitat state matches expectations (e.g., connections are correctly mated (b), air locks are closed (c), etc.)

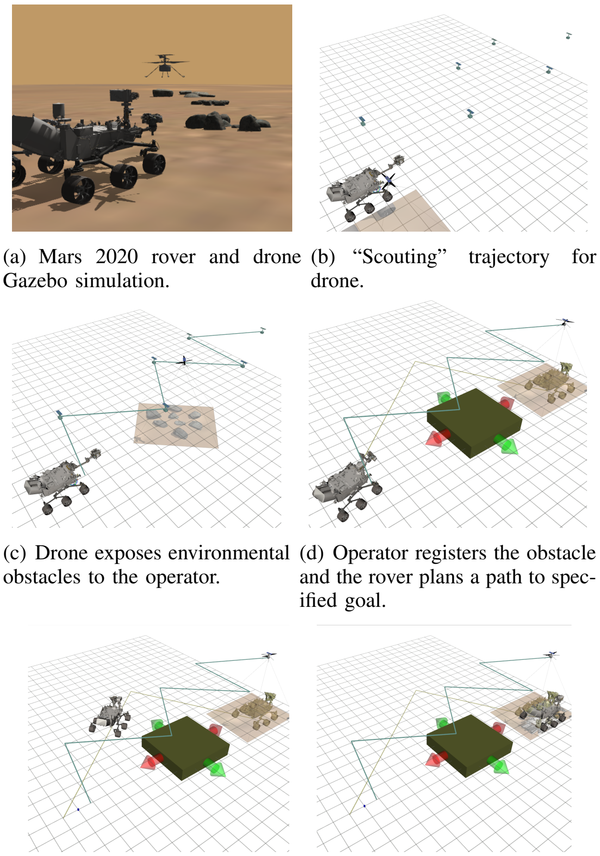
B. Inspection Task

With sensor waypoints, non-manipulation based tasks such as inspection can now be encoded. Fig. 4 illustrates a simulated inspection of a lunar habitat by a rover. The template trajectory defines important viewing geometries relative to a modeled habitat (see Fig. 4(a)). An operator plans and executes movements to each waypoint, where they inspect images taken from the rover (shown in Fig. 4(b)-(c)) to verify the condition of the habitat.

C. Heterogeneous Scouting Behavior

The introduced sensor and mobility types enable multi-agent cooperative non-manipulation tasks to be encoded as ATs. An example of such a task is demonstrated in Fig. 5, which illustrates a "scouting" behavior for a simulated Mars 2020 rover and drone team[2]. The template defines two trajectories: one to scout the terrain in front of the rover and one to navigate the rover to a user-defined goal. These motions are defined relative to the rover, such that the template trajectories define forward motion. The template can be rotated with respect to the robot to define motion in any direction. While executing the scouting trajectory, a supervisor can monitor the feedback from the drone to

[2] https://mars.nasa.gov/mars2020



(a) Mars 2020 rover and drone Gazebo simulation.

(b) "Scouting" trajectory for drone.

(c) Drone exposes environmental obstacles to the operator.

(d) Operator registers the obstacle and the rover plans a path to specified goal.

(e) Rover executes planned motions.

(f) Rover achieves goal configuration, confirmed by visual feedback from the overhead drone.

Fig. 5. AT enabling a scouting behavior to support navigation on a simulated Mars 2020 rover/drone team (a). (b) A trajectory for the drone to scout the terrain in front of the rover indicated by teal drone models. (c) The drone plans and executes the trajectory, exposing environmental features to the operator. (d) The operator registers obstacles exposed by the drone (green cube) between the desired goal (shown as yellow rover model) and the current rover position. A plan is generated for the rover (path shown in yellow). (e) The rover executes the planned motions to navigate around the registered obstacles. (f) The rover arrives at the operator defined goal, which is confirmed using visual feedback from the overhead drone.

register obstacles that lie in the rover's path. With this information, the rover can plan collision-free motion around these obstacles to reach the specified location.

VI. Conclusions

This paper introduces extensions to the AT concept to facilitate their use in multi-agent settings and in tasks beyond manipulation. Techniques to dynamically remap roles in ATs to ease their use in multi-agent settings were outlined. The extended formal definitions of the AT concept allow a variety of cooperative multi-agent behaviors to be encoded within a supervised autonomy paradigm. The included demonstrations highlight the potential of the new definition. While these demonstrations leveraged an operator to sequence motions, future work may investigate the use of tools such as finite state machines, behavior trees, or planning and scheduling techniques to enable sequences of trajectories to be autonomously achieved.

Acknowledgement

The authors would like to thank their colleagues at TRACLabs, with specific recognition to former colleagues Nicu Stiurca, Stephen Hart, Oscar Youngquist, and Patrick Beeson for their contributions to this work.

References

[1] S. Hart, P. Dinh, and K. Hambuchen, "The affordance template ROS package for robot task programming," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[2] S. Hart, A. H. Quispe, M. Lanighan, and S. Gee., "Generalized Affordance Templates for Mobile Manipulation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2022.

[3] M. A. Goodrich, D. R. Olsen, J. W. Crandall, and T. J. Palmer, "Experiments in adjustable autonomy," in *Proceedings of IJCAI Workshop on autonomy, delegation and control: interacting with intelligent agents*, 2001.

[4] N. A. Radford, P. Strawser, K. Hambuchen, J. S. Mehling, W. K. Verdeyen, A. S. Donnan, J. Holley, J. Sanchez, V. Nguyen, L. Bridgwater, *et al.*, "Valkyrie: NASA's first bipedal humanoid robot," *Journal of Field Robotics*, vol. 32, no. 3, pp. 397–419, 2015.

[5] J. James, Y. Weng, S. Hart, P. Beeson, and R. Burridge, "Prophetic goal-space planning for human-in-the-loop mobile manipulation," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2015, pp. 1185–1192.

[6] S. J. Jorgensen, M. W. Lanighan, S. S. Bertrand, A. Watson, J. S. Altemus, R. S. Askew, L. Bridgwater, B. Domingue, C. Kendrick, J. Lee, *et al.*, "Deploying the NASA Valkyrie humanoid for IED response: An initial approach and evaluation summary," in *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*, 2019.

[7] T. McMahon, O. Jenkins, and N. Amato, "Affordance wayfields for task and motion planning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 10 2018, pp. 2955–2962.

[8] A. Pettinger, P. F. Cassidy Elliott, and M. Pryor, "Reducing the teleoperator's cognitive burden for complex contact tasks using affordance primitives," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 11 513–11 518.

[9] Z. Ye, J. Y. Song, Z. Sui, S. Hart, J. Vilchis, W. S. Lasecki, and O. C. Jenkins, "Human-in-the-loop Pose Estimation via Shared Autonomy," in *26th International Conference on Intelligent User Interfaces*. College Station TX USA: ACM, Apr. 2021, pp. 387–391.

[10] M. Hagenow, M. R. Zinn, T. Fong, E. Laske, and K. A. Hambuchen, "Affordance template registration via human-in-the-loop corrections," *Advances in Space Robotics and Back to Earth Workshop at IROS 2021*, 2021.

[11] X. Chen, K. Zheng, Z. Zeng, C. Kisailus, S. Basu, J. Cooney, J. Pavlasek, and O. C. Jenkins, "Manipulation-oriented object perception in clutter through affordance coordinate frames," in *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*. IEEE, 2022, pp. 186–193.

[12] D. Calvert, D. Anderson, T. Bialek, S. McCrory, L. Penco, J. Pratt, and R. Griffin, "Authoring and operating humanoid behaviors on the fly using coactive design principles," *arXiv preprint arXiv:2307.12944*, 2023.

[13] Y. A. Dinitz, "An algorithm for the solution of the problem of maximal flow in a network with power estimation," in *Doklady Akademii nauk*, vol. 194, no. 4. Russian Academy of Sciences, 1970, pp. 754–757.

[14] J. Edmonds and R. M. Karp, "Theoretical improvements in algorithmic efficiency for network flow problems," *Journal of the ACM (JACM)*, vol. 19, no. 2, pp. 248–264, 1972.

[15] J. E. Hopcroft and R. M. Karp, "An n^5/2 algorithm for maximum matchings in bipartite graphs," *SIAM Journal on computing*, vol. 2, no. 4, pp. 225–231, 1973.

[16] M. Harrower and C. A. Brewer, "Colorbrewer.org: an online tool for selecting colour schemes for maps," *The Cartographic Journal*, vol. 40, no. 1, pp. 27–37, 2003.

[17] P. Beeson, S. Hart, and S. Gee, "Cartesian motion planning & task programming with CRAFTSMAN," in *Robotics: Science and Systems Workshop on Task and Motion Planning*, 2016.