Title

Household Navigation and Manipulation for Everyday Object Rearrangement Tasks

Permalink

https://escholarship.org/uc/item/5114d600

Authors

Iyer, Shrutheesh R
Pal, Anwesan
Hu, Jiaming
et al.

Publication Date

2023-12-11

DOI

10.1109/irc59093.2023.00021

Copyright Information

Household navigation and manipulation for everyday object rearrangement tasks

Shrutheesh R. Iyer*[2], Anwesan Pal[1], Jiaming Hu[1], Akanimoh Adeleye[1], Aditya Aggarwal[1], and Henrik I. Christensen[1]

[1]Contextual Robotics Institute, UC San Diego
[2]Aurora Operations, Inc.
[1,2]{siyer, a2pal, jih189, akadeley, a9aggarwal, hichristensen}@ucsd.edu

***Abstract*—We consider the problem of building an assistive robotic system that can help humans in daily household cleanup tasks. Creating such an autonomous system in real-world environments is inherently quite challenging, as a general solution may not suit the preferences of a particular customer. Moreover, such a system consists of multi-objective tasks comprising – (i) Detection of misplaced objects and prediction of their potentially correct placements, (ii) Fine-grained manipulation for stable object grasping, and (iii) Room-to-room navigation for transferring objects in unseen environments. This work systematically tackles each component and integrates them into a complete object rearrangement pipeline. To validate our proposed system, we conduct multiple experiments on a real robotic platform involving multi-room object transfer, user preference-based placement, and complex pick-and-place tasks. Additional details including video demonstrations of our work are available at https://sites.google.com/eng.ucsd.edu/home-robot.**

***Index Terms*—long-term navigation, real-world manipulation, preferential object placement**

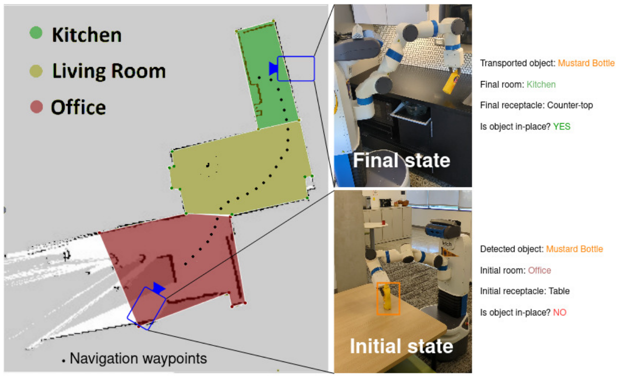


Fig. 1. An example of a home-robot rearrangement task. At the initial state, the robot identifies the `mustard_bottle` object and determines that it is misplaced in the `office`. Subsequently, the robot transports it to its correct location in the `kitchen` on top of the `counter-top`. The semantic map used for the navigation task is shown on the left with the robot's trajectory.

I. Introduction

Creating autonomous agents to aid human beings in everyday household chores has long been considered to be the holy grail of service robotics research. In this work, we take a step towards that goal by proposing a complete system for an indoor tidy-up task. Usually, this comprises of identifying misplaced objects in the environment, and transferring them to their desired locations. Several aspects of this inherently long-horizon task make it particularly challenging in a real-world environment. Firstly, recognizing out of place objects in a noisy environment is a non-trivial problem. While state-of-the-art open-vocabulary object detectors [1]–[4] are quite adept at localizing objects in a zero-shot manner, determining whether they belong in a particular environment is more complicated, as it also involves understanding scene context. Secondly, user preferences for placing objects in the "correct" room and surface (hereafter called *receptacle*), are often subjective, thereby inhibiting the sole use of generic common-sense reasoning models. Thirdly, manipulating unknown objects in a cluttered environment is still an open research problem due to the difficulty of affordance estimation and motion planning. Finally, delivering an object to a previously unlabeled receptacle in the target room is particularly challenging, specially if the precise location of said receptacle is unknown.

In this paper, we address each component for rearranging household objects in a real-world setting utilizing the Fetch [5] mobile manipulation platform. To ensure robustness and scalability within the physical world, we propose a modular system that is capable of performing (i) user-preference based reasoning through collaborative filtering, (ii) fine-grained pick-up of unknown objects and placement on previously unlabeled receptacles, and (iii) multi-room rearrangement. All these functionalities are coordinated by behavior trees that can handle failure at different levels. An example of the operation is shown in Figure 1.

The rest of the paper is organized as follows. Section II discusses existing approaches for object rearrangement in home-robot environments. Section III has a description of each component we use to perform the overall task, with a summary of the integrated system in Section IV. We explain the conducted experiments in detail in Section V, and provide a summary of our work with some future goals in Section VI.

II. Related Work

Recently, indoor object rearrangement tasks using mobile robots have received a lot of attention from the robotics and

*Work done while at the Contextual Robotics Institute, UC San Diego computer vision community. Due to the increasing number of Embodied AI platforms available [6]–[11], several approaches have been proposed for solving the complete mobile manipulation task in a number of home environments. However, most of these methods [6], [9], [12]–[14] are entirely trained in simulation, and therefore rarely generalize to real-world environments. Other works have adopted the task planning approach, but are either restricted to specific tasks such as folding clothes [15] and rearranging kitchens [16], or follow a pre-defined template [17]. Some approaches [18]–[21] focus on the human-robot interaction aspect, but not on autonomy. Lately, large language models (LLMs) have gained popularity for robotic manipulation, both for task planning [22]–[24], as well as end-to-end execution [25]–[28]. While these large foundational models are proficient at reasoning about object semantics, accurately grounding the offline acquired knowledge in a dynamic physical environment is still considered to be a non-trivial problem. Two efforts closest to ours are that of Wu *et al.* [22] and Castro *et al.* [29]. Wu *et al.* [22] use LLMs to infer generalized user preferences and use it to tidy a room. However, they do not handle fine-grained manipulation, need rigorous prompt engineering to understand user preferences, and are limited to within-room navigation. Castro *et al.* [29] do consider room-to-room navigation, but they rely on manually annotated prior semantic maps for querying the exact locations of target rooms and receptacles. In contrast, we only build a simple 2D geometric map with rough room locations, and proceed to identify receptacles in the environment on the fly.

III. Components

Our proposed ensemble system for home-robot rearrangement contains four primary modules: scene recognition and mapping, object rearrangement, manipulation, and navigation.

A. Semantic mapping and visual recognition

Our detection module perceives the environment in two stages. The first stage involves construction of a semantic map of the environment for localization, while the second stage deals with recognition of objects in the environment. The localization system uses Cartographer mapper [30] to generate a LiDAR-based 2D occupancy-grid environment map. For simplicity, we manually annotate the locations in the map with a semantic label of the room category. This manual annotation step can also be replaced by an automated module such as [31]. We do not annotate locations of receptacles however, as knowing their exact positions apriori is a strong assumption in dynamic environments. For object recognition, we use the DETIC [4] model trained on twenty-thousand object classes. With this detector, we can detect both manipulable objects and receptacle surfaces for the rearrangement task.

B. Object rearrangement

The rearrangement module involves repositioning objects in the home, using both *common-sense reasoning* (to determine target rooms) and *human preferences* (for selecting target receptacles). We utilize a large human-labeled dataset [32] for object placement preferences in homes, creating a knowledge base to predict likely room locations for objects. Then, we utilize user preference to capture diversity in human choices for receptacle locations. However, the dataset does not contain a particular user's preference for *all* the objects, leading to a sparse user-preference matrix. Thus, given scores and user ranked preferences, we use collaborative filtering [33] to fill out the sparse matrix. Subsequently, matrix factorization [34] is performed to predict user ratings $r_{u,i}$ for user u and item i. In our case, an item i refers to an object's placement in a particular room and receptacle. We can predict a user's rating using $f(u,i) = \gamma_u * \gamma_i$. Here, $\gamma_u \in \mathbb{R}^d$ and $\gamma_i \in \mathbb{R}^d$ are latent vectors representing the row of a user in matrix γ_U and column of an item in matrix γ_I, and d is the lower dimensional space. To choose parameters $\gamma = \{\gamma_u, \gamma_i\}$ to closely fit the data, we minimize a loss function using Mean Squared Error with an L2 regularization term.

$$\arg\min_{\gamma} \frac{1}{|\tau|} \sum_{r_{u,i} \in \tau} w_{u,i}(r_{u,i} - f(u,i))^2 + \lambda\Omega(\gamma) \qquad (1)$$

where τ is our corpus of ratings and $\Omega(\gamma)$ is $\ell2$ norm $||\gamma||_2^2$. Our approach allows us to estimate the full preferences of users' desired correct object placement locations.

Object rearrangement involves two main steps: (1) Identifying misplaced objects by checking if their current location is in the top-k (10 in this work) likely locations from our user-preference matrix, and (2) Predicting preference-based placement by first determining the target room using common-sense reasoning, and then identifying various potential receptacle locations within that room based on a sampled user identity.

C. Manipulation of objects

The manipulation module includes planning to understand and construct a scene, analyzing interaction methods with the target object, and planning the required motion for effective interaction, all aligned with the task goal.

Before constructing the planning scene, the robot in this work possesses some prior knowledge of the environment. For instance, it understands that most objects should be positioned on a flat receptacle such as a `table`, or `counter-top`. Therefore, the receptacle serves as a common obstacle during our manipulation tasks, making it beneficial to prioritize its search once an object is detected. Finally, the receptacle is added as a single entity in the planning scene for efficient collision detection, while a voxel set represents the remaining non-target objects, optimizing resource usage.

Even though the robot knows the planning scene, interacting with the target object is crucial. In this work, grasping is the prevailing contact approach. For this, a learning-based grasp prediction [35] model is utilized to estimate a set of possible grasping poses. However, pick-and-place is not the only manipulation action available. The robot must also account for potential object motions based on the task requirements. For instance, it might need to open a drawer before placing an object. Consequently, the robot must compute the required

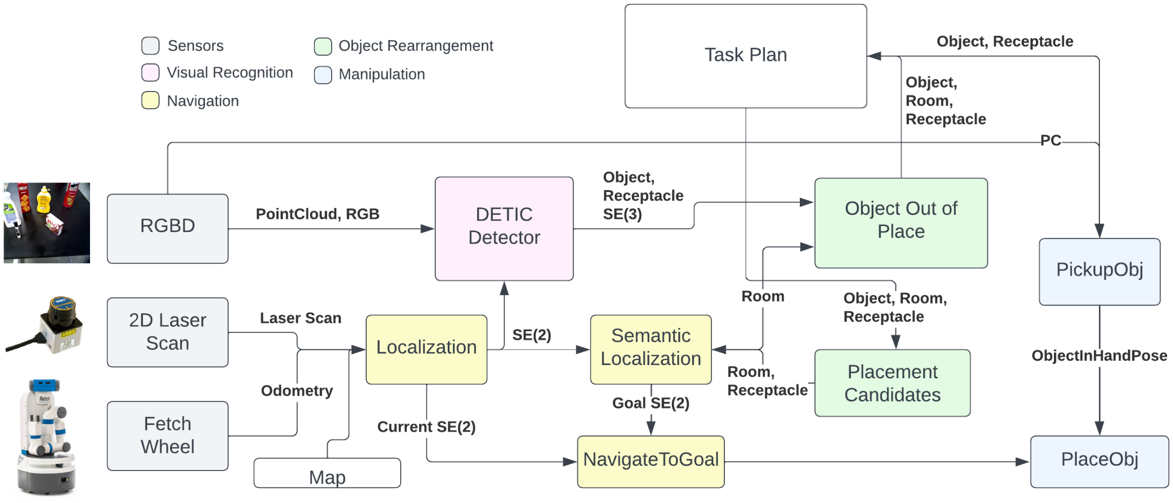


Fig. 2. The overall architecture of the proposed system, as discussed in IV-A

motion to open it after identifying a set of arm configurations to grasp the drawer handle. The robot may explore alternative approaches if the motion is found before the timeout.

D. Semantic navigation

The navigation module aims to move the robot between different locations for the rearrangement task, and is considered in two stages – (i) room-to-room navigation for planning a path to the target room, and (ii) receptacle navigation for navigating to the correct receptacle in the target room.

For room-to-room navigation, the 2D coordinate of the center of the target room is first computed from the annotated semantic map. Using this destination point, a heuristic point-goal navigation algorithm is adopted to plan a trajectory by avoiding obstacles along the way with the Navfn planner. Upon reaching the target room, the receptacle navigation module is called. First, the entire room is scanned for possible receptacles for the held object, and the position of each candidate receptacle is updated in the map by re-projecting the detected object from the depth map of the camera. Then, the most likely target receptacle is chosen as per the rearrangement module III-B. Finally, a second heuristic planner is called to make the robot move as close to the goal receptacle position as is feasible in collision-free space, which is achieved through the Carrot Planner.

IV. System Integration

In this section, we first outline the primary structure of our proposed system, and then discuss the flow of control using behavior trees.

A. System Architecture

Figure 2 depicts the overall architecture of the proposed system. The task plan is provided in the form of behavior trees, as discussed in the next section. The localization module reads the semantic map, along with sensor data, to get the robot's current coordinates in the room. The detector module reads the sensor data, along with the robot's location, and identifies objects in the environment along with their 3D locations in the map. The object rearrangement module obtains a list of $(object, receptacle, room)$ tuples from the perception and localization modules to identify misplaced objects and propose "correct" placements. The manipulation module picks up the misplaced object. The target room for placement provides the goal location for the navigator module, which then calls the perception module to locate the target receptacle and navigate to it. The manipulation module finally places the object either on the receptacle or inside the receptacle, depending on the specified goal from the rearrangement module.

B. Use of Behavior Trees for Integration

A key component of the complex home-robot system is the composition of the different capabilities of the robot to execute the task robustly and continuously. This calls for a control architecture that is modular and capable of switching between tasks such that the different tasks can be called anywhere during the workflow. Consequently, Behavior Trees (BTs) are used to monitor and orchestrate the flow of the entire system. BTs is a modular control architecture developed for controlling autonomous agents that supports reactive behavior. [36] A BT consists of control nodes and leaf nodes, where the leaf nodes are atomic operations that include actuation and sensing. The control nodes are behavior nodes that chain together multiple nodes. Each node (with its children) is a behavior that the robot can exhibit. A behavior can be composed of multiple behaviors. For instance, picking up a misplaced object is composed of two behaviors: identifying a misplaced object and picking up a target object.

Figure 3 shows the BT of the home-robot tidy module. The system begins by calling the misplaced object identification

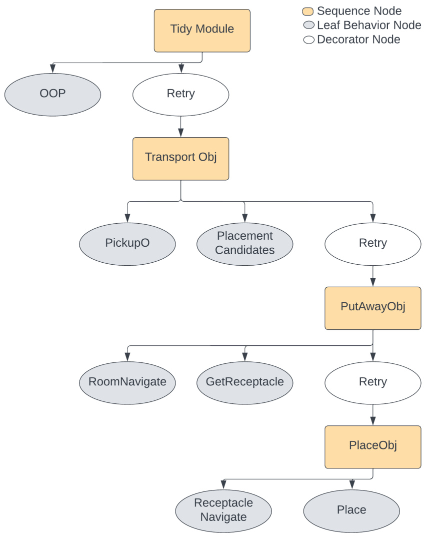


Fig. 3. The complete behavior tree of the home-robot tidy module

(OOP) method of the rearrangement module III-B. For every object, potential placement candidates (PlacementCandidates) are computed III-B. The Pickup Behavior III-C is called on the misplaced object. RoomNavigator, followed by ReceptacleNavigator modules are executed, given by the placement candidates. The PlaceBehavior is finally called to place the object. If the place action fails, then the robot tries other candidate receptacles until one succeeds, highlighting the BT's advantages. This is implemented through multiple Decorator Nodes that can facilitate retry behaviors. The different messages from each behavior are passed around through blackboard mechanisms. The visual recognition module constantly runs in the background throughout the episode. The system continues to run until the robot either makes an unrecoverable mistake (such as dropping the object or a hardware failure) or all items are correctly placed.

V. Experiments

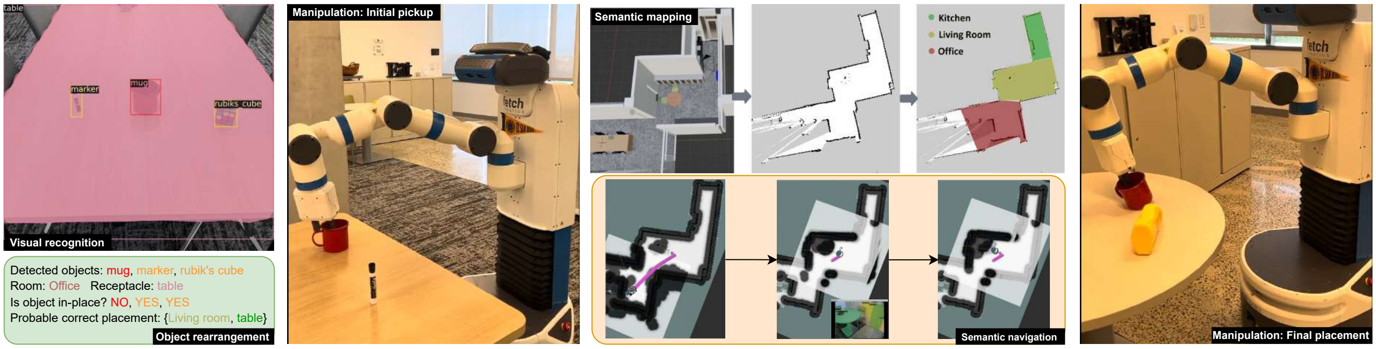
We test our system through various real-world experiments, involving (i) *Semantic mapping and visual recognition* for generating coarse semantic environment representations and detecting target objects and receptacle surfaces, (ii) *Object rearrangement* for identifying and repositioning misplaced objects, (iii) *Object manipulation* for ensuring stable object interactions, and (iv) *Semantic navigation* for robot's trajectory planning with the generated semantic map. Figure 4 contains a pictorial representation of each of our individual modules at work for a tidy-up task. All the experiments are performed in the real world using a simple apartment environment, created from an actual communal office space within our laboratory. The overall environment has an office space, living room, and a kitchen as shown in the semantic map in Figure 4. The following sections describe the different types of experiments.

A. *Long-horizon object rearrangement*

The first experiment that we consider is a long-horizon tidy-up task, where the robot has to identify multiple misplaced objects, and move them to their respective target locations spanning multiple rooms. The rearrangement episode typically begins with detecting a misplaced object, o_1, in the environment. The entire tidy module is called to rearrange the object to the correct location. Upon reaching the destination, the robot further scans the environment for any other misplaced objects. If it finds another such object o_2, it repeats the entire process sequentially until o_2 has also been correctly placed. Figure 5 illustrates the process where $o_1 = \texttt{mug}$ is transported from an office table to the living-room table, and $o_2 = \texttt{mustard\_bottle}$ is then moved from the living-room table to the kitchen counter-top.

B. *User-preference based object tidy-up*

Our second experiment focuses on transferring an object o to different locations, catering to individual user preferences. This experiment acknowledges the subjective nature of object placement in homes. Section III-B describes a collaborative-filtering approach for generating a user matrix about how objects can be placed differently based on human preference. For this experiment, we sampled two users, U_1 and U_2, and tabulated their preference regarding target room locations and receptacle surfaces for eight different objects in Table I.



Detected objects: mug, marker, rubik's cube
Room: Office Receptacle: table
Is object in-place? NO, YES, YES
Probable correct placement: {Living room, table}

Fig. 4. All proposed system components. Visual recognition module detects both target objects and receptacle surfaces. Object rearrangement module identifies misplaced objects and suggest their desired location. The manipulation module ensures the reliability of each pick and place action. Mapping module builds a 2D environment map and semantically paints it with room labels. Finally, the navigation module uses the semantic map to plan the robot's trajectory.

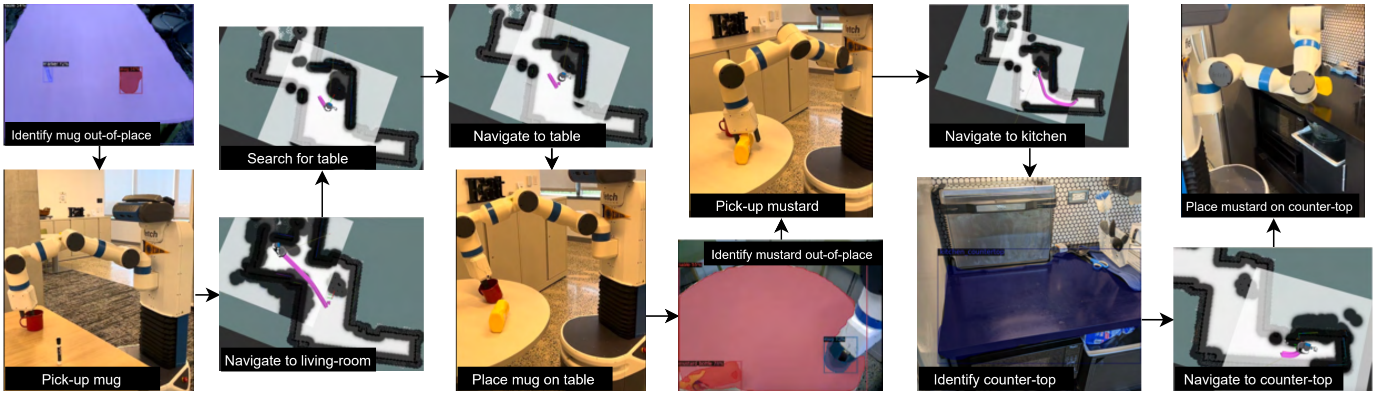


Fig. 5. Long horizon rearrangement task. Initially, a `mug` is identified to be incorrectly placed on the office table. Then, the robot picks it up, and navigates to the desired target location by first going to the livingroom, and then moving towards the table receptacle. After placing the mug, a second object `mustard_bottle` is found misplaced on the livingroom table. Subsequently, the robot picks the bottle, and transports it to the countertop in the kitchen.

We conducted real-world experiments using the `mug` object. As per Table I, U_1 considers the preferred target room to be `kitchen`, with the top-2 receptacles surfaces being `counter` and `sink`. In contrast, U_2 desires the `mug` to be primarily placed in the `livingroom`, with the top-2 receptacles being `table` and `shelf`. Thus, we perform multiple real-world episodes by sampling the preferences of U_1 and U_2 as our object rearrangement module, respectively.

C. *Complex interactions*

A rearrangement task may require the robot to interact with the environment before proceeding with object placement beyond just picking and placing. For instance, placing an object inside a *closed* receptacle. In this work, we demonstrate this concept through the task of placing a Rubik's Cube inside a drawer. Because the drawer is initially closed, the robot has to perform multiple sub-tasks based on the behavior tree shown in Figure 6. Furthermore, as depicted in Figure 7, the robot estimates a temporary location for the Rubik's Cube and predicts grasp poses to open the drawer. Following that, the robot places the cube into the temporary location and opens the drawer, so it can grasp and place the cube inside the drawer.

TABLE I
PREFERRED OBJECT PLACEMENTS FOR TWO SAMPLED USERS

	Sampled user U_1		**Sampled user U_2**	
Objects	**Preferred rooms**	**Preferred receptacles**	**Preferred rooms**	**Preferred receptacles**
rubik's cube	office kitchen livingroom	[shelf, table] [counter, table] [drawer, table]	livingroom office kitchen	[drawer, table] [table, drawer] [drawer, table]
mustard bottle	kitchen livingroom office	[drawer, counter] [table, sofa] [table, drawer]	kitchen livingroom office	[shelf, counter] [table, drawer] [drawer, table]
marker	livingroom office kitchen	[drawer, shelf] [table, drawer] [drawer, table]	office kitchen livingroom	[table, drawer] [table, drawer] [table, shelf]
cracker box	kitchen livingroom office	[drawer, table] [drawer, table] [drawer, shelf]	office kitchen livingroom	[shelf, drawer] [drawer, table] [drawer, sofa]
bleach cleanser	livingroom office kitchen	[drawer, table] [shelf, table] [shelf, drawer]	office kitchen livingroom	[shelf, table] [drawer, table] [table, drawer]
gelatin box	office kitchen livingroom	[table, shelf] [drawer, counter] [drawer, table]	livingroom office kitchen	[table, drawer] [table, shelf] [drawer, counter]
potted meat can	kitchen livingroom office	[counter, shelf] [drawer, table] [drawer, table]	office kitchen livingroom	[drawer, table] [counter, shelf] [drawer, table]
mug	kitchen livingroom office	[counter, sink] [shelf, sofa] [drawer, table]	livingroom office kitchen	[table, shelf] [drawer, table] [sink, drawer]

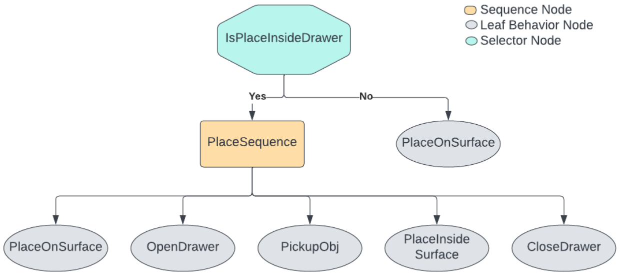


Fig. 6. The behavior tree to place an object into the drawer.

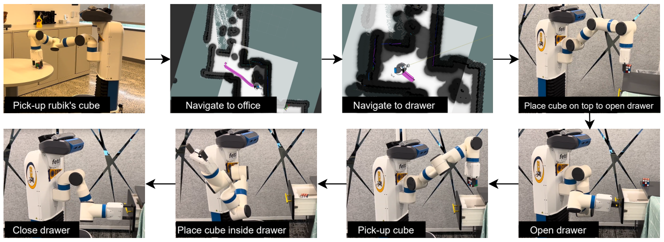


Fig. 7. In a multifaceted task such as placing a Rubik's cube into a drawer, a robot must undertake a series of interrelated actions. Initially, the robot approaches the drawer. Recognizing that the drawer must be opened to place the cube inside, it then discerns the need to temporarily set down the Rubik's cube. Only after opening the drawer can it successfully place the cube within.

VI. Summary

The world needs a home robot that can do more than vacuuming. We have presented key components for navigating robustly in a home setting, for detecting of objects and receptacles and determining if they are out of place. Skills for manipulating and handling objects in a daily setting for a task such as clean-up or reset of a home to a nominal setting are introduced to allow clean-up. Finally, using a combination of common-sense reasoning and recommender systems, a strategy to detect objects out of place and suggest improved locations to put them is discussed. All these techniques are integrated into a consistent and robust framework using behavior trees and implemented on the Fetch robot using a ROS based architecture. We have demonstrated the final system and how it can be used in a real-world scenario with modest complexity, for clean-up of a space by placement of objects in appropriate locations.

In this work, we mainly demonstrated an integration of fundamental robotic skills in a modular representation for house-hold tidy-up tasks. In the future, we want to aim for longer-term testing across multiple environments, with an additional goal of optimizing the system for speed in space cleanup, such that it can compete with humans.

References

[1] A. Zareian, K. D. Rosa, D. H. Hu, and S.-F. Chang, "Open-vocabulary object detection using captions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[2] X. Gu, T.-Y. Lin, W. Kuo, and Y. Cui, "Open-vocabulary object detection via vision and language knowledge distillation," *arXiv preprint arXiv:2104.13921*, 2021.

[3] M. Minderer, A. Gritsenko, A. Stone, M. Neumann, D. Weissenborn, A. Dosovitskiy, A. Mahendran, A. Arnab, M. Dehghani, Z. Shen *et al.*, "Simple open-vocabulary object detection," in *European Conference on Computer Vision*, 2022.

[4] X. Zhou, R. Girdhar, A. Joulin, P. Krähenbühl, and I. Misra, "Detecting twenty-thousand classes using image-level supervision," in *European Conference on Computer Vision*, 2022.

[5] M. W. Wise, M. Ferguson, D. King, E. Diehr, and D. Dymesich, "Fetch and freight: Standard platforms for service robot applications," in *Workshop on Autonomous Mobile Service Robots, held at the 2016 International Joint Conference on Artificial Intelligence, NYC*, 2016.

[6] X. Puig, K. Ra, M. Boben, J. Li, T. Wang, S. Fidler, and A. Torralba, "Virtualhome: Simulating household activities via programs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[7] M. Shridhar, J. Thomason, D. Gordon, Y. Bisk, W. Han, R. Mottaghi, L. Zettlemoyer, and D. Fox, "Alfred: A benchmark for interpreting grounded instructions for everyday tasks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020.

[8] M. Shridhar, X. Yuan, M.-A. Côté, Y. Bisk, A. Trischler, and M. Hausknecht, "Alfworld: Aligning text and embodied environments for interactive learning," *arXiv preprint arXiv:2010.03768*, 2020.

[9] A. Szot, A. Clegg, E. Undersander, E. Wijmans, Y. Zhao, J. Turner, N. Maestre, M. Mukadam, D. S. Chaplot, O. Maksymets *et al.*, "Habitat 2.0: Training home assistants to rearrange their habitat," *Advances in Neural Information Processing Systems*, 2021.

[10] V.-P. Berges, A. Szot, D. S. Chaplot, A. Gokaslan, R. Mottaghi, D. Batra, and E. Undersander, "Galactic: Scaling end-to-end reinforcement learning for rearrangement at 100k steps-per-second," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[11] C. Li, F. Xia, R. Martín-Martín, M. Lingelbach, S. Srivastava, B. Shen, K. Vainio, C. Gokmen, G. Dharan, T. Jain *et al.*, "igibson 2.0: Object-centric simulation for robot learning of everyday household tasks," *arXiv preprint arXiv:2108.03272*, 2021.

[12] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, "Language models as zero-shot planners: Extracting actionable knowledge for embodied agents," in *International Conference on Machine Learning*, 2022.

[13] S. Y. Min, D. S. Chaplot, P. Ravikumar, Y. Bisk, and R. Salakhutdinov, "Film: Following instructions in language with modular methods," *arXiv preprint arXiv:2110.07342*, 2021.

[14] D. Batra, A. X. Chang, S. Chernova, A. J. Davison, J. Deng, V. Koltun, S. Levine, J. Malik, I. Mordatch, R. Mottaghi *et al.*, "Rearrangement: A challenge for embodied ai," *arXiv preprint arXiv:2011.01975*, 2020.

[15] S. Srivastava, S. Zilberstein, A. Gupta, P. Abbeel, and S. Russell, "Tractability of planning with loops," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29, 2015.

[16] B. Wu, R. Martin-Martin, and L. Fei-Fei, "M-ember: Tackling long-horizon mobile manipulation via factorized domain transfer," *arXiv preprint arXiv:2305.13567*, 2023.

[17] G. Cui, W. Shuai, and X. Chen, "Semantic task planning for service robots in open worlds," *Future Internet*, vol. 13, no. 2, 2021.

[18] C. Schaeffer and T. May, "Care-o-bot-a system for assisting elderly or disabled persons in home environments," *Assistive technology on the threshold of the new millenium*, vol. 3, 1999.

[19] B. Graf, M. Hans, and R. D. Schraft, "Care-o-bot ii—development of a next generation robotic home assistant," *Autonomous robots*, vol. 16, no. 2, 2004.

[20] U. Reiser, C. Connette, J. Fischer, J. Kubacki, A. Bubeck, F. Weisshardt, T. Jacobs, C. Parlitz, M. Hägele, and A. Verl, "Care-o-bot® 3-creating a product vision for service robot applications by integrating design and technology," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009.

[21] R. Kittmann, T. Fröhlich, J. Schäfer, U. Reiser, F. Weißhardt, and A. Haug, "Let me introduce myself: I am care-o-bot 4, a gentleman robot," *Mensch und computer 2015–proceedings*, 2015.

[22] J. Wu, R. Antonova, A. Kan, M. Lepert, A. Zeng, S. Song, J. Bohg, S. Rusinkiewicz, and T. Funkhouser, "Tidybot: Personalized robot assistance with large language models," *arXiv preprint arXiv:2305.05658*, 2023.

[23] Y. Ding, X. Zhang, C. Paxton, and S. Zhang, "Task and motion planning with large language models for object rearrangement," *arXiv preprint arXiv:2303.06247*, 2023.

[24] H. Chang, K. Gao, K. Boyalakuntla, A. Lee, B. Huang, H. U. Kumar, J. Yu, and A. Boularias, "Lgmcts: Language-guided monte-carlo tree search for executable semantic object rearrangement," 2023.

[25] A. Brohan *et al.*, "Rt-1: Robotics transformer for real-world control at scale," 2023.

[26] Y. Jiang, A. Gupta, Z. Zhang, G. Wang, Y. Dou, Y. Chen, L. Fei-Fei, A. Anandkumar, Y. Zhu, and L. Fan, "Vima: General robot manipulation with multimodal prompts," 2023.

[27] A. Brohan *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," 2023.

[28] M. Ahn *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," 2022.

[29] S. Castro, "Behavior Trees for Home Service Robotics Tasks," https://www.youtube.com/watch?v=xbvMnpwXNPk, 2022.

[30] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2d lidar slam," in *2016 IEEE international conference on robotics and automation (ICRA)*, 2016.

[31] A. Pal, C. Nieto-Granda, and H. I. Christensen, "Deduce: Diverse scene detection methods in unseen challenging environments," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[32] Y. Kant, A. Ramachandran, S. Yenamandra, I. Gilitschenski, D. Batra, A. Szot, and H. Agrawal, "Housekeep: Tidying virtual households using commonsense reasoning," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIX*, 2022.

[33] N. Abdo, C. Stachniss, L. Spinello, and W. Burgard, "Robot, organize my shelves! tidying up objects by predicting user preferences," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[34] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008.

[35] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, "Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[36] M. Colledanchise and P. Ögren, *Behavior trees in robotics and AI: An introduction*. CRC Press, 2018.