Built to Order: A Virtual Reality Interface for Assigning High-Level Assembly Goals to Remote Robots

Portia Wang
Columbia University
New York, NY, USA
pw2491@columbia.edu

Shreeya Jain
Columbia University
New York, NY, USA
sj3131@columbia.edu

Manxueying Li
Columbia University
New York, NY, USA
ml4529@columbia.edu

Shutaro Aoyama
Columbia University
New York, NY, USA
sa4168@columbia.edu

Xuezhen Wang
Columbia University
New York, NY, USA
ww2604@columbia.edu

Shuran Song
Columbia University
New York, NY, USA
ss5941@columbia.edu

Jen-Shuo Liu
Columbia University
New York, NY, USA
jl5004@columbia.edu

Steven Feiner
Columbia University
New York, NY, USA
feiner@cs.columbia.edu

ABSTRACT

Many real-world factory tasks require human expertise and involvement for robot control. However, traditional robot operation requires that users undergo extensive and time-consuming robot-specific training to understand the specific constraints of each robot. We describe a user interface that supports a user in assigning and monitoring remote assembly tasks in Virtual Reality (VR) through high-level goal-based instructions rather than low-level direct control. Our user interface is part of a testbed in which a motion-planning algorithm determines, verifies, and executes robot-specific trajectories in simulation.

ACM Reference Format:

Portia Wang, Shreeya Jain, Manxueying Li, Shutaro Aoyama, Xuezhen Wang, Shuran Song, Jen-Shuo Liu, and Steven Feiner. 2023. Built to Order: A Virtual Reality Interface for Assigning High-Level Assembly Goals to Remote Robots. In *The 2023 ACM Symposium on Spatial User Interaction (SUI '23), October 13–15, 2023, Sydney, NSW, Australia.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3607822.3618005

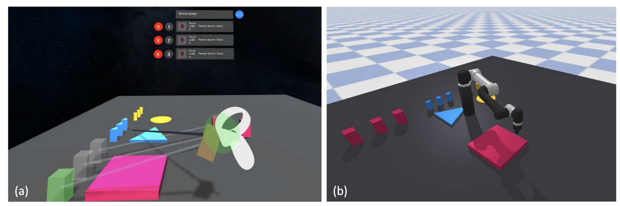

Figure 1: Testbed. (a) VR user interface. (b) Simulation environment.

1 INTRODUCTION

Much research has explored robotic teleoperation systems that are based on direct low-level control [2, 6, 7], often in real time [2, 7]. These systems assume the participation of domain experts who possess the necessary knowledge to operate specific robots, and that real-time errors are easily fixable or avoidable, which may not always hold true [8]. An alternative approach is for users to provide high-level control: The user specifies the high-level actions of robots [5, 9] or the goal states [4, 10] for objects. This approach allows users to operate robots without needing to have any knowledge about the specific robots, such as their individual capabilities.

Many interfaces for goal-based high-level control focus on scenarios in which a user sends a single batch of instructions to a robot, and upon execution of those instructions, the robot's task is completed [4, 10]. In more complex assembly tasks, there is a need to be able to give the robot multiple instructions in parallel. On the other hand, there are user interfaces for action-based high-level control designed for simultaneous assignment of multiple tasks [5, 9]. However, these user interfaces address robot actions and not task goals.

To address this, we introduce a teleoperation user interface for assembly tasks that is (1) based on high-level goals for robots and (2) designed to support specifying multiple goals in parallel. Our Virtual Reality (VR) testbed system (Figure 1) shows the user virtual replicas of task objects to control in 6DoF using VR controllers. It enables the user to assign multiple high-level assembly goals to a robot arm and monitor their progress. Our system uses an interactive communication architecture that allows the user to define multiple goal-based tasks, validate, and confirm their execution. We highlight the significance of visualizing the statuses of each assigned goal, to ensure that the user understands what is happening as the goals are accomplished.

2 SYSTEM DESIGN AND IMPLEMENTATION

Our testbed (Figure 1) consists of a VR user interface in Unity 2021.3.4f1 and a remote robot simulation in PyBullet 3.21 [1]. The user wears a Meta Quest 2 VR headset connected to a computer powered by an Intel® Core™ i9-9900K processor and an Nvidia GeForce RTX 3090 graphics card. Two hand-held VR controllers are used to assign high-level goal-based instructions for 6DoF manipulation. The remote simulation environment includes a Universal Robots UR5 robot arm with a suction gripper.

In our workflow, the user first assigns high-level end-goal poses individually by translating and rotating objects. Once the user clicks the *validity check button*, these intended scene modifications are

SUI '23, October 13–15, 2023, Sydney, NSW, Australia

ACM ISBN 979-8-4007-0281-5/23/10.
https://doi.org/10.1145/3607822.3618005

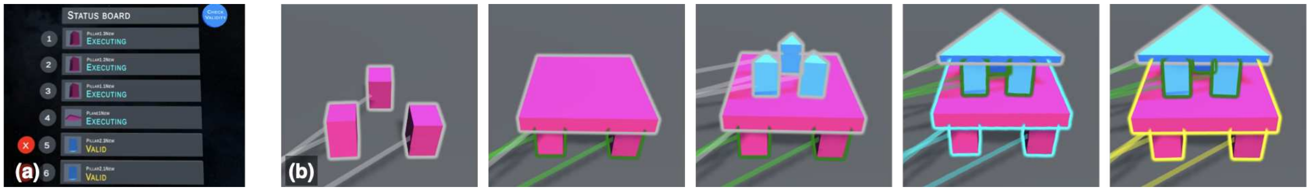

Figure 2: Visualizations of status of assigned goals. (a) Status board showing chronologically ordered list of goals. (b) Colored halos indicate goal status. *Unverified Goal* (white), *Valid Goal* (green), *Executing* (cyan), *Execution Completed* (yellow). Each goal object is connected by a line to its original position.

communicated to the simulation environment. Given a pair of initial and end poses for a goal, our testbed uses the Rapidly-exploring Random Trees (RRT) algorithm [3] to try to find a feasible path. The simulation verifies the robot trajectory for each step and communicates the validity status to the user through the VR user interface. If RRT cannot find a valid path, the user is notified and can assign alternative goals. Otherwise, the user can click the *execute button*, which prompts a second simulation to execute the robot action. Unlike the initial validity check, this second simulation mimics the actual time required for the real robot to perform the task. It is employed for the purposes of our upcoming user study; in real-world settings, our testbed will interface with an actual UR5 robot.

Once the goals specified thus far have been executed, the simulation communicates the new poses for the objects, which are subsequently updated in the VR user interface. The user then visually checks the results to determine whether there are any errors, and if so, whether they are acceptable. If an error is unacceptable, the user could attempt to fix it by assigning a new goal (or, in a more complete version of our system, ask for assistance from a technician with robot-specific knowledge). Otherwise, the user then proceeds to confirm the executed tasks by clicking the *confirm button*. It is crucial to note that this system allows for concurrent operations on multiple objects. While any step is being validated or executed, the user is free to assign goals to other objects.

Since the user can provide multiple goal-based instructions for objects in parallel, some objects can be having their goals validated or executed while other objects are waiting for goals to be assigned. To plan the next operation, the user needs to understand the current status of each object. We employ two kinds of visualization for goal status: (a) a status board and (b) colored halos, as illustrated in Figure 2. The status board displays all tasks in the queue and their respective statuses in chronological order. Colored halos represent the current status of each object through distinct colors. With these visualizations, users can acquire an overview of all tasks or understand the status of individual objects as needed.

3 DISCUSSION

We have conducted internal tests with simple assembly tasks. In our informal evaluation, we observed that our approach empowers users to assign multiple goals concurrently within a unified 3D environment and to understand the statuses of these goals. However, while the system offers an encompassing view, it also presents challenges. Specifically, the simultaneous display of both ongoing and completed tasks can sometimes make it difficult for users to quickly distinguish between them, suggesting a need to further refine our visualizations.

4 CONCLUSIONS AND FUTURE WORK

We are developing a VR teleoperation user interface to assign multiple high-level assembly goals, potentially in parallel, validate them with a path-planning algorithm, and execute them. By allowing the user to specify only high-level goals, our system avoids the need for the user to know the operational details of the available robots and, therefore, can be used for different robot environments without additional training.

When interfacing with a physical UR5 robot, we will use real-world sensor data instead of the simulation. This will allow users to compare the actual data with their specified goals when confirming goal execution.

While displaying all specified goals simultaneously in a single scene, no matter their status, might help some users understand the overall plan, it could also be overwhelming to others. As an alternative approach, we could visualize goals in a timeline of views of the environment. This way, users could more easily follow the progression of the robot's tasks and better understand when goals are achieved.

ACKNOWLEDGMENTS

This research was funded in part by National Science Foundation Grant CMMI-2037101.

REFERENCES

[1] E. Coumans and Y. Bai. 2016–2023. PyBullet, a Python module for physics simulation for games, robotics and machine learning. http://pybullet.org.
[2] A. Franzluebbers and K. Johnson. 2019. Remote Robotic Arm Teleoperation through Virtual Reality. In *Proc. SUI 2019*. Article 27.
[3] S. M. LaValle. 1998. *Rapidly-exploring random trees*. http://lavalle.pl/rrt/
[4] Y. Li, S. Agrawal, J.-S. Liu, S. Feiner, and S. Song. 2022. Scene Editing as Teleoperation: A Case Study in 6DoF Kit Assembly. In *Proc. IROS 2022*. 4773–4780.
[5] K. Liu, D. Sakamoto, M. Inami, and T. Igarashi. 2011. Roboshop: Multi-Layered Sketching Interface for Robot Housework Assignment and Management. In *Proc. CHI 2011*. 647–656.
[6] K. Mahadevan, Y. Chen, M. Cakmak, A. Tang, and T. Grossman. 2022. Mimic: In-Situ Recording and Re-Use of Demonstrations to Support Robot Teleoperation. In *Proc. UIST 2022*. Article 40.
[7] A. Poncela and L. Gallardo-Estrella. 2015. Command-based voice teleoperation of a mobile robot via a human-robot interface. *Robotica* 33, 1 (2015), 1–18.
[8] D. J. Rea and S. H. Seo. 2022. Still Not Solved: A Call for Renewed Focus on User-Centered Teleoperation Interfaces. *Front. Robot. AI* 9 (2022), 704225.
[9] E. Senft, M. Hagenow, K. Welsh, R. Radwin, M. Zinn, M. Gleicher, and B. Mutlu. 2021. Task-Level Authoring for Remote Robot Teleoperation. *Front. Robot. AI* 8 (2021).
[10] E. Yigitbas, K. Karakaya, I. Jovanovikj, and G. Engels. 2021. Enhancing Human-in-the-Loop Adaptive Systems through Digital Twins and VR Interfaces. In *Proc. SEAMS 2021*. 30–40.