2024 IEEE Conference Virtual Reality and 3D User Interfaces (VR) | 979-8-3503-7402-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/VR58804.2024.00066

Asynchronously Assigning, Monitoring, and Managing Assembly Goals in Virtual Reality for High-Level Robot Teleoperation

Shutaro Aoyama*† Jen-Shuo Liu*‡ Portia Wang Shreeya Jain Xuezhen Wang Jingxi Xu
Shuran Song Barbara Tversky Steven Feiner§

Columbia University

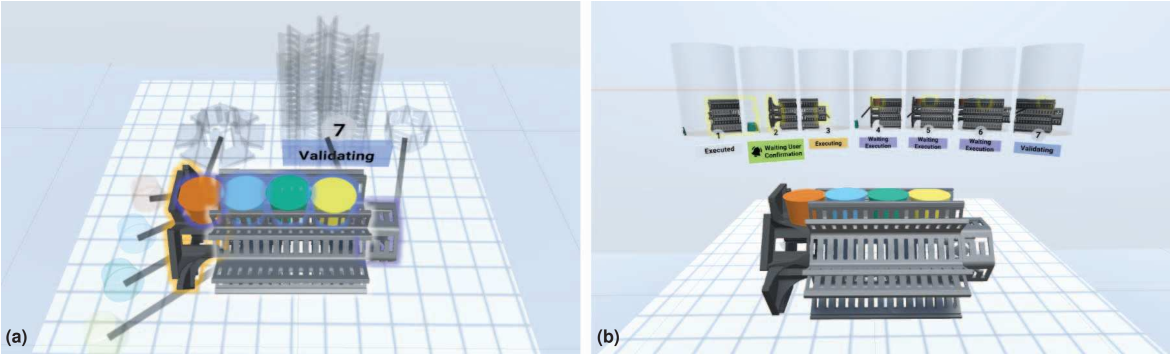

Figure 1: Virtual reality for asynchronous high-level robot teleoperation. The user specifies goal poses for virtual replicas of objects to be manipulated at a remote site. (a) Aggregated View. (b) Timeline View.

Abstract

We present a prototype virtual reality user interface for robot teleoperation that supports high-level specification of 3D object positions and orientations in remote assembly tasks. Users interact with virtual replicas of task objects. They asynchronously assign multiple goals in the form of 6DoF destination poses without needing to be familiar with specific robots and their capabilities, and manage and monitor the execution of these goals. The user interface employs two different spatiotemporal visualizations for assigned goals: one represents all goals within the user's workspace (Aggregated View), while the other depicts each goal within a separate world in miniature (Timeline View). We conducted a user study of the interface without the robot system to compare how these visualizations affect user efficiency and task load. The results show that while the Aggregated View helped the participants finish the task faster, the participants preferred the Timeline View.

Index Terms: Human-centered computing—Human computer interaction (HCI)—Interaction paradigms—Virtual reality; Human-centered computing—Interaction design—Interaction design process and methods—User interface design; Computer systems organization—Embedded and cyber-physical systems Robotics—External interfaces for robotics

1 Introduction

Robots can perform household chores, participate in factory assembly lines, and conduct mechanical inspections, handling tasks that are repetitive, dangerous, or beyond human ability. However, independently accomplishing novel, complex, and dynamically changing tasks in real-world environments is challenging for robots. In addition, creative tasks without predefined goals, such as custom furniture creation, also require human involvement. Moreover, a human user can provide corrective suggestions when robots have made errors. Consequently, the role of human-in-the-loop interfaces for robot teleoperation remains critical for these applications [20].

Many robot teleoperation systems rely on low-level control, whether direct [10, 12, 29, 40] or indirect [25], often accomplished through synchronous communication [10,12,40]. These systems presuppose the involvement of domain experts who are knowledgeable about the specific robots they manage and able to correct or prevent real-time errors—a requirement that may be difficult to meet [41].

On the other hand, in high-level robot teleoperation systems, users indicate high-level actions for robots to perform [28, 43] or define target goals (e.g., object positions or orientations) for them to achieve [23,33,52]. Users of these systems do not need to understand the capabilities of the robots or how to operate them. Studies [23,33] have shown that high-level teleoperation systems can be more efficient than low-level ones. Further, high-level robot teleoperation systems often use Virtual Reality (VR), as VR can achieve lower workload and higher usability compared with desktop-based teleoperation [51].

In many high-level teleoperation systems, a user sends a single set of instructions to a robot, and then waits for them to be executed. [23, 52]. Other systems allow instructions to be assigned asynchronously, without waiting for previous ones to be completed [28, 43]. While such asynchronous systems can help users be more efficient, prior work primarily focuses on teleoperation methods and does not compare and evaluate different user interfaces and visualizations.

To address this, we are developing a VR system for teleoperation that is (1) based on high-level object-pose goals and (2) designed to support asynchronous control. In our system, the user manipulates virtual replicas of objects to assign high-level six–degrees-of-freedom (6DoF) goal poses to these objects. We designed and compared two visualizations for the goals. The *Aggregated View* presents

[*] Equal contribution.
[†] e-mail: sa4168@columbia.edu
[‡] e-mail: jl5004@columbia.edu
[§] e-mail: feiner@cs.columbia.edu

979-8-3503-7402-5/24/$31.00 ©2024 IEEE
DOI 10.1109/VR58804.2024.00066

all assigned goals simultaneously within the user's workspace, as shown in Figure 1(a) and schematically in Figure 2(a). The *Timeline View* presents each goal in a separate world in miniature within a timeline, as shown in Figure 1(b) and schematically in Figure 2(b). We also compared two variants of the Aggregated View, which differ as to how much information is shown. In addition, for the Aggregated and Timeline Views, we compared different ways of using line style to distinguish steps in the task flow. We performed a user study of the VR interface without the robot system. Our study showed that the Aggregated View improves efficiency, while participants subjectively prefer the Timeline View.

Overall, we make the following contributions:

- Design and implementation of two visualizations for asynchronously assigning, monitoring, and managing assembly goals in high-level teleoperation: Aggregated View and Timeline View.
- A VR user study showing that both variants of the Aggregated View increase user efficiency, while users prefer the Timeline View.
- An evaluation of line styles used with the Aggregated View and the Timeline View to help users distinguish steps in the task flow, showing that line style has a significant impact on error rate, but not on task completion time.

2 Related Work

Beer et al. [3] proposed a taxonomy of ten levels of robot autonomy for human–robot interaction systems, ranging from manual to full autonomy. Many teleoperation systems employ direct control [5, 10, 12, 29, 35, 40, 42, 51], resulting in lower levels of autonomy. In these systems, performance largely relies on the user's understanding of the system's physical constraints including range of motion, current joint configuration, and the size and velocity of joints. On the other hand, fully autonomous systems [13, 15, 19, 36] also possess inherent limitations, particularly in executing novel, complex, and dynamic tasks in unstructured real-world environments, a challenge underscored in the existing literature [20].

We focus on the level between manual and full autonomy: teleoperation interfaces in which the user decides on and assigns high-level goals, while the robot system performs low-level planning and execution. This is classified in Beer's taxonomy as "Executive control."

2.1 Interfaces for High-level Goal-based Teleoperation

Previous work has explored teleoperation interfaces for assigning a high-level goal to a robot [2, 23, 28, 33, 38, 52, 54]. Li et al. [23] present a system in which users can specify a high-level goal 6DoF pose for an object to be manipulated by a remote robot arm. Meng et al. [33] allow users to specify high-level goals by manipulating the scene in VR. Yigitbas et al. [52] also present a VR interface to assign high-level goals with transparency and controllability. These systems demonstrate the efficiency of high-level control compared with direct control methods. They also highlight that this control method enables effective robot manipulation without requiring users to understand the specific capabilities of individual robots [28, 52]. However, these systems support only synchronous control, allowing users to assign a single batch of high-level goals. In contrast, we explore interfaces for teleoperation systems with high-level *asynchronous* control.

2.2 Interfaces for Asynchronous High-Level Goal-based Teleoperation

Several teleoperation systems enable users to assign multiple high-level goals asynchronously [9, 28, 43, 50]. While the robot is executing the assigned goals, users can assign further goals in advance while also monitoring the execution. For example, Walker et al. [50] introduced an augmented reality (AR) system for drone manipulation through the assignment of multiple waypoints linked with high-level goals, and Senft et al. [43] developed a 2D AR system to author instructions for a robot arm. Liu et al. [28] developed an interface to assign household goals by sketching on a bird's-eye view of an environment, supporting multiple assignments by providing multiple layers to sketch. However, prior work has predominantly concentrated on control methods, and not the comparison of different interfaces and visualizations for asynchronous high-level goal-based teleoperation.

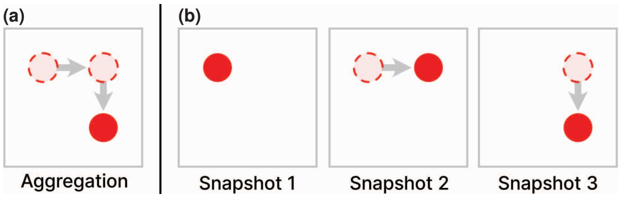

Figure 2: (a) Aggregation of all spatiotemporal data in a single scene. (b) Snapshots of different moments in time.

The visualization strategies employed in these studies primarily aggregate all assigned goals within a single view, leading to potential visual clutter as more goals are added. Walker et al. [50] displayed waypoints within a 3D environment and Senft et al. [43] incorporated an overlay of all tasks in a 2D view. Since increasing the number of assigned goals amplifies scene information, thereby complicating the user's visual experience, there is a need for user interfaces for teleoperation systems that clearly distinguish between multiple assigned goals and their current statuses. While Liu et al. [28] provide a feature for toggling the visibility of representations of tasks, its primary application is the scheduling of housework over a relatively long period of time and cannot be directly applied to the assignment and monitoring of assembly goals.

2.3 Spatiotemporal History Visualizations

There are two distinct approaches to visualizing spatiotemporal history: (1) aggregating all moments into a single scene (Figure 2a) and (2) presenting different moments as a timeline of individual snapshots (Figure 2b).

Among interfaces that aggregate multiple moments of time into a single scene, Büschel et al. [6] and Lilija et al. [24] overlay object movement paths on 3D worlds to visualize spatial recordings. Su et al. [46] show graphical editor history by overlaying graphical annotations on a scene. Moreover, Zhang et al. [55] layer "space-time cubes" to present spatiotemporal data, aggregating moments in one scene and allowing users to view specific snapshots.

In contrast, interfaces that use a timeline include work by Kurlander and Feiner [21] that visualizes graphical editor history as a timeline of snapshots at user-controlled levels of temporal granularity. Denning et al. [8] similarly summarize the edit history of 3D meshes in a timeline. In VR, Worlds in Miniature (WIMs) [45] have been used to depict snapshots at specific points in time. For example, Mahadevan et al. [30] present a WIM-based interface allowing users to query spatiotemporal recordings, and Zhang et al. [53] offer a version control interface for 3D scenes, displaying commit history as a graph of WIMs.

Drawing inspiration from this prior work, our research explores spatiotemporal visualizations for presenting user-assigned goals for high-level teleoperation.

3 Our System

3.1 Manipulating Virtual Replicas in VR

In the virtual environment depicted in Figure 1 (a–b), the user observes virtual replicas of objects at a remote site. They can manip-

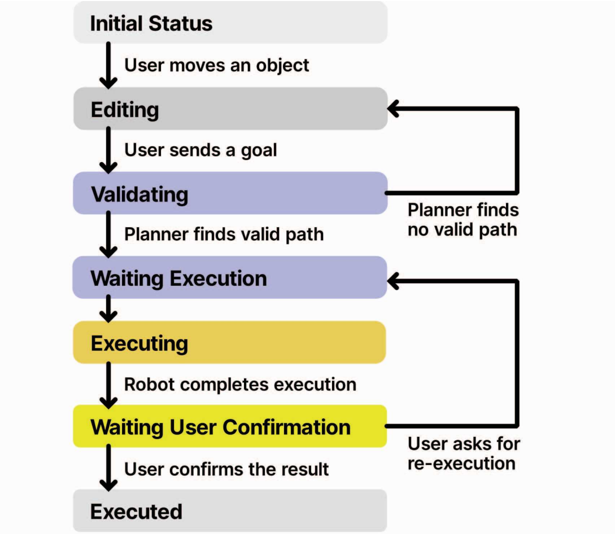

Figure 3: Status of goals. Please see Section 3.2 for an explanation of each phase.

ulate these virtual objects using their controllers to specify 6DoF goal poses. Since the environment is virtual, the user can manipulate objects unbound by physical constraints.

3.2 Assigning High-level Goals

As shown in Figure 3, the user initiates the execution of a high-level goal by assigning a 6DoF goal pose to a virtual object. They can manipulate an object to try out possible poses, before finalizing their decision by selecting the object and clicking the *Send* button, which triggers the path-planning system. This system assesses the scene and the destination object pose, determining the feasibility of the user-specified goal. If a valid path is found, the system proceeds to execution by the robot; otherwise, the user is alerted and may input an alternative goal. Once the assigned goal has 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 if they are acceptable and can request to retry the execution if they are not. Otherwise, the user clicks the *Confirm* button.

3.3 Asynchronous Control

While a goal is being validated or executed by the system, the user is free to assign additional goals. Figure 4 shows the hypothetical timeline of a user of our system. The user can send multiple goals and confirm execution results while the planner finds valid paths and the robot executes the assigned goals. Although our current testbed contains a single remote robot (an articulated arm with a parallel gripper), we believe that such asynchronous control systems can generalize to multiple robots with an appropriate planner.

3.4 Implementation

We developed our user interface in Unity 2022.2.1f1 [48]. The user wears a Meta Quest 2 [34] VR headset connected to a computer running Windows 10 powered by an Intel® CoreT i9-9900K processor and an Nvidia GeForce RTX 3090 graphics card. A hand-held Meta Quest 2 controller is used to assign high-level goal-based instructions. Our path-planning system is based on a remote robot simulation in PyBullet 3.21 [7]. The simulation environment includes a UR5 robot arm [49] with a parallel gripper (Figure 5). It uses the Rapidly-exploring Random Trees (RRT) algorithm [22] to try to find a feasible path from the initial state to the goal state. Communication between the VR user interface and the path-planning system occurs through designated directory files. The user interface generates a JSON file for each validation request, transmitting data to the path-planning system. After validation, the path-planning system dispatches the results in a text file, which the user interface reads.

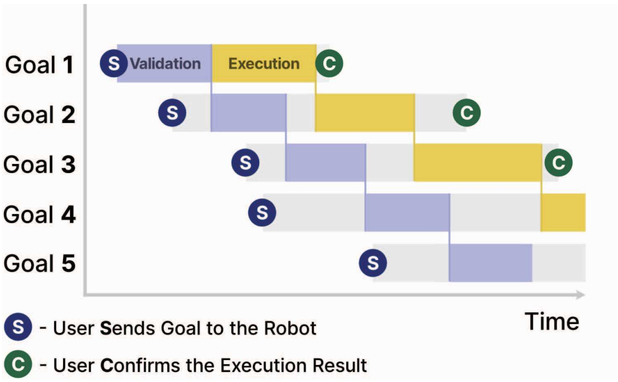

Figure 4: Gantt chart representing asynchronous control in our system. Goals are assigned and sent by the user, validated by the planner, executed by the robot, and confirmed by the user. Assignment and confirmation of goals are done by the user, while validation and execution are done by the planner and the robot. Since times for sending goals and confirming results are independent of the robot, waiting times (gray phases) are of varying lengths.

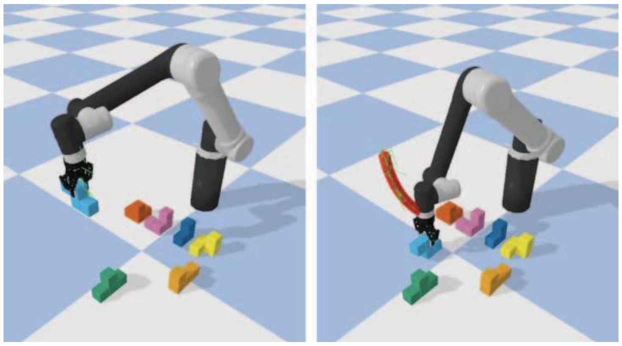

Figure 5: Path Planning System built with PyBullet [7]. The red line shows the planned path.

4 Visualization Conditions

4.1 Design Requirements

To allow the user to complete the workflow described in Section 3, we defined the following design requirements.

- **Authoring New Goals:** The user should be able to set new goals by directly manipulating virtual objects in the VR environment.
- **Monitoring Statuses of Goals:** The user should be able to monitor the status of a goal they have assigned. This should be possible even in complex scenarios with multiple goals.
- **Confirming Execution Results:** Once a goal has been executed, the user should be able to see the result, confirm whether the robot has executed the goal correctly, and ask for re-execution if it has not.

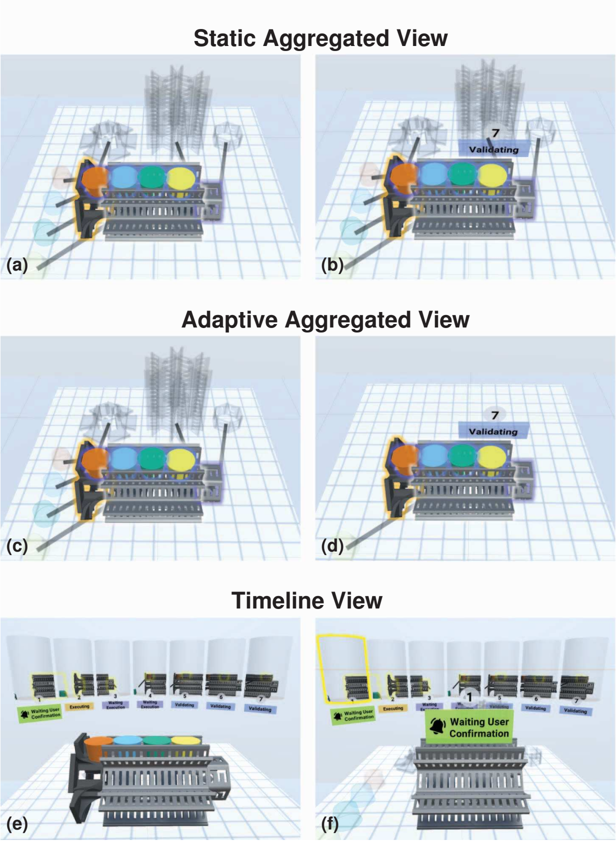

Figure 6: Views. (a) In the Static Aggregated View, the user sees previous and goal poses for all objects with goals. (b) When the user touches an object with a goal, they also see a placard presenting its current status. (c) In the Adaptive Aggregated View, the user also sees previous and goal poses for all objects with goals by default. (d) When the user touches an object with a goal, they see the snapshot of the moment of time when that goal was assigned. (e) In the Timeline View, the user sees the scene with the current object by default. (f) The user can click a WIM in the timeline to see the snapshot of the moment when that goal was assigned.

Given these requirements, a significant challenge arises: users need to interact with the current scene, while also monitoring and interacting with goals assigned in the past. This leads to a research question: *How can we best present the assigned goals from multiple points in time?*

Our definition of a "best" design will be based on several metrics. Efficiency is one; the interface should streamline the process of assigning and monitoring goals, reducing the time spent on each operation. The task load on the user should also be minimized to ensure that they can use the system for extended periods without fatigue or errors. Furthermore, the user should be able to assign new goals and confirm executed goals simultaneously, allowing for continuous workflow and maximizing the robot's operational time.

4.2 Views of the Objects and the Goals

4.2.1 Aggregated Views

Aggregated views aim to present a comprehensive scene by aggregating all the goals within a single scene. This design was inspired by prior work on spatiotemporal history visualizations that aggregates all moments into a single scene (Section 2.3), as well as some of the prior systems for high-level goal-based teleoperation [43, 50] (Section 2.2).

The goal pose of an object is presented in its actual color, while the previous pose is presented as semi-transparent. We use placards and halos around objects with goals, where the color of a placard or halo indicates goal status (Figure 7). Halos are always visible, while a placard is shown only when the user hovers over its object.

We have two variants of aggregated view: Static Aggregated View and Adaptive Aggregated View.

Static Aggregated View *(SA)*: The SA view (Figure 6a–b) is an aggregated view in which the user sees each object in its goal pose and previous pose.

Adaptive Aggregated View *(AA)*: The AA view (Figure 6c–d) adds adaptive filtering features to the SA view. In its default state, with the user not touching any object, the interface displays each object in its goal pose and previous pose, just like the SA view (Figure 6c). However, when the user touches a goal pose or a previous pose, non-essential information is filtered out (Figure 6d), displaying only the following: First, for the object of touched pose, its original pose and the previous poses assigned before the touched pose will be shown. This helps users understand the sequence of goals leading up to the touched one. Second, for every other object in the scene, the goal pose that was last assigned before the touched goal will be shown. This provides the context of the scene when the touched goal was assigned.

When the user touches the goal pose, this filtering effect continues as the user grabs and manipulates it to assign a new goal. This feature helps the user focus on the current scene, enabling precise specification of the new goal. We tested both SA and AA to explore whether showing all info in the aggregated view will be helpful.

4.2.2 Timeline View

Timeline View *(T)*: The T view (Figure 6e–f) presents a timeline of WIMs [45], where each WIM represents a specific goal (Figure 8). This design was inspired by prior work on spatiotemporal history visualization that presents individual snapshots of different moments (Section 2.3).

The index and status of goals are displayed on a placard below each WIM. WIMs are arranged in a curved layout, inspired by the design space suggested by Fouché et al. [11]. To clearly indicate which object was manipulated in each WIM, these scenes zoom in on the manipulated object, which is highlighted, inspired by work on editable graphical histories [21].

By default, the user sees the scene with the goal poses of every object, and they can assign new goals by grabbing and manipulating them. (Figure 6e) The user can point their controller to a WIM and click on it to see a *snapshot* of the scene corresponding to the selected WIM's goal. (Figure 6f) In the snapshot, the following will be displayed: First, for the object of the selected WIM's goal, its original pose and the sequence of previous poses. Second, for all other objects in the scene, the snapshot will display the previous pose that was assigned before the selected WIM's goal was assigned. This snapshot will help the user observe the assigned goal pose with the context of the scene at the point of time the goal was assigned, understand the status of the goal, and perform actions such as confirmation of execution when needed. The user cannot manipulate objects and assign new goals in this snapshot scene. Clicking the WIM again will return the user to the default scene, so they can again assign new goals.

This view can avoid the clutter of the main scene by presenting information about each goal and its status in a separate snapshot. The user can refer to the timeline to check the status of each goal. However, this approach requires the user to look at two different places, unlike the aggregated views. Furthermore, a WIM provides limited information about the scene surrounding each goal, and the user has to select on each WIM to see the goal within the context of

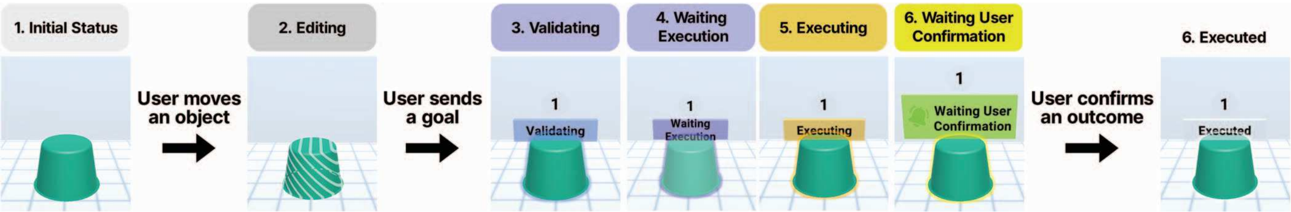

Figure 7: Visualizations of goal status in the Static and Adaptive Aggregated Views. The color of the halo around the goal pose of an object and the placard above it indicates its status. The stripe pattern in the editing mode indicates that the object is still being manipulated, and its goal has not yet been assigned.

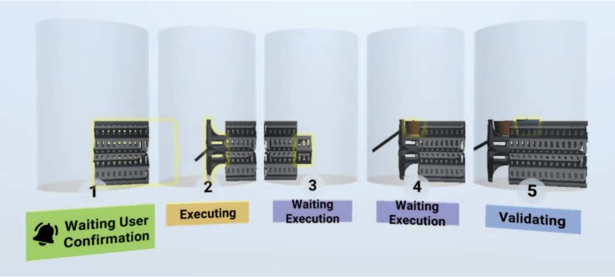

Figure 8: WIMs in the Timeline View. The scene in each WIM is a snapshot that zooms in on the object assigned a goal at that time, which is surrounded by a yellow halo.

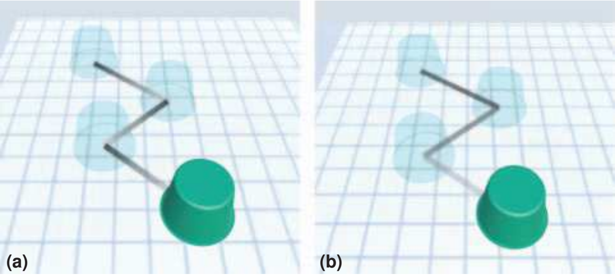

Figure 9: Line styles. (a) Segmented Gradient. (b) Continuous Gradient.

the surrounding scene. When the user wishes to review past goals or confirm execution results by viewing the surrounding scene, this introduces an additional step.

4.3 Representation of the Direction of Lines

Users can assign multiple goals to a single object. These goals are displayed together in the scene and connected by lines. It is crucial to clearly represent the direction of lines to ensure users understand the order of assigned goals.

Perin et al. [39] identified gradients as effective tools to represent the direction and flow of time in lines. Therefore, we used gradients transitioning from dark gray to light gray in two distinct styles: segmented and continuous.

4.3.1 Segmented Gradient (SG)

The segmented gradient (Figure 9a) represents the direction of lines between poses. Each gradient starts from one object and ends at another. This segmented approach offers a more granular view of the order of directions, allowing users to understand which of the two connected poses is the earlier goal.

4.3.2 Continuous Gradient (CG)

Inspired by the line design in Liu et al. [26], this approach (Figure 9b) visualizes the order of sequence of poses by a continuous gradient. The gradient begins at the initial pose, traverses through all the previous poses, and ends at the goal pose. This continuous gradient enables them to understand the sequence of tasks and their relative orders.

In essence, while the continuous gradient offers a macroscopic view of the object's entire timeline, the segmented gradient provides a microscopic perspective, highlighting the order of individual goals. The choice between these visualizations would be contingent on the user's need for either a comprehensive overview or detailed insights into the orders of specific goals.

5 User Study

We conducted a formal user study of the user interface, without the planner and robot, to explore how user performance is affected by the visualizations. We also conducted pilot studies before the formal user study to refine the visualizations and determine the appropriate length of our user study. The review board at our institution approved these studies.

5.1 Pilot Studies

In our first pilot study with four participants, we compared the transparency of instruction objects in the Aggregated Visualization. We determined that a 20% alpha value minimized user task load. Since every participant completed the study faster than the maximum experiment duration, we decided to increase the number of tasks from 15 to 24 to collect more data and add practice trials to the formal study.

Our second pilot study involved five participants and tested various view and line conditions. Based on user feedback, we improved the adaptive behavior of AA upon user interaction with current objects. The revised approach displays a scene exclusively containing current objects, with previous goals omitted.

5.2 Hypotheses

Based on our design requirements, we formulated the following hypotheses:

H1. *The SA and the AA conditions will allow participants to finish tasks in a shorter time than with the T condition.* We hypothesize this because we anticipate that SA and AA provide the user with a better overview of the tasks that have been proposed, as they appear in the same workspace. In addition, participants will require fewer interactions in the SA and AA conditions since they do not have to select which snapshots to view.

H2. *The CG condition will allow participants to finish tasks in a shorter time than the SG condition.* We hypothesized this because the CG condition provides the participant with a more comprehensive view of the order of the tasks already assigned, so they can better focus on the current task.

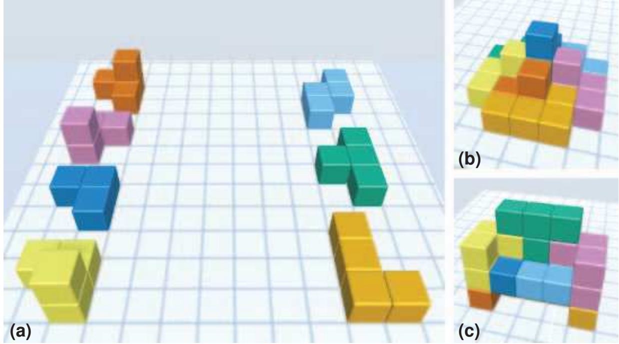

Figure 10: (a) The seven pieces from a Soma cube. (b–c) Examples of assembled shapes.

H3. *Participants will prefer the T condition to the SA and AA conditions.* We believe that the T condition presents the least amount of information in the focused scene, so participants do not need to differentiate between the main object and the preceding ones. As a result, participants would prefer it to SA and AA, which are harder to parse.

5.3 Methods

5.3.1 Participants

We recruited 19 participants by email to our department email lists and posted flyers. Of the 19 participants, two did not complete the study: one failed to follow the instructions given by the study coordinator, and the other did not complete the study due to time constraints. For the remaining 17 participants (four female, age 18–33, average 23.6) who completed the user study, seven had no prior experience with AR/VR, seven had used AR/VR several times, one owned a VR headset for gaming, and two used VR for long-term jobs or research. Each participant received a 15 USD gift card as compensation. No participant in our formal study participated in the pilot studies in Section 5.1.

5.3.2 Study Design

We used a within-subject design for our study. The study had two variables: (SA, AA, T) × (CG, SG), resulting in six conditions. The order in which the blocks appeared was counterbalanced based on the condition to which each block belonged, ensuring each participant experienced a different order of conditions. Since we found in our pilot studies that participants were much slower in the first two trials, we added three practice trials at the beginning of the study.

In our study, participants were asked to complete 24 tasks in total, each requiring them to send goals for the assembly of a specified target shape and subsequently confirm its execution.

To ensure a consistent workload across tasks while maintaining comprehensibility, we employed seven pieces from a Soma cube. This design mirrors many real-world assembly tasks where specific assembly orders, diverse parts, and target shapes are prevalent. Each virtual piece was distinctly colored using the Color Universal Design palette [37], facilitating clear differentiation between parts.

Two types of tasks were given to the user:

- **Assembly Task (A):** Participants began with individual Soma pieces laid out on a table (Figure 10a) and were tasked with assembling a target shape (Figure 10b).
- **Disassemble-then-assemble Task (DA):** Participants started with a pre-assembled shape and were required to dismantle and lay out those objects and then assemble them to achieve the target shape (Figure 10b). This adds the need for an additional step for each object.

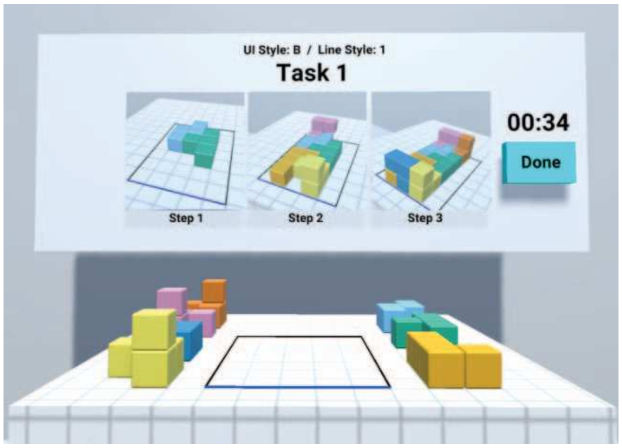

Figure 11: Environment for the user study. The front panel shows the current task number, conditions, timer, and a guide to build the goal shape. Users click the *"Done"* button upon completion of each task.

Participants worked in the environment shown in Figure 11. For both task types, A and DA, a three-step guide image was provided on a front panel to assist participants in reaching the final shape. The guide was structured as follows: the first step showed the placement of the lowest two pieces, the second showed the next three, and the final step showed the last two.

For each of the six conditions, a participant ran two assembly task trials and two disassemble-then-assemble task trials. The participant was told to click *Done* once they complete assigning all goals and handle every confirmation request. Task completion time is defined as the duration from the start of the task to when the *Done* button is pressed.

To maintain consistency and stability across study participants and trials, we implemented a simplified simulation of execution rather than using our actual UR5 robot and path planner. We did not use the real robot system because its noisy sensor inputs could cause confounds. We suppressed path planning because RRT and many other path planners use randomization, so the same goal might yield different paths each time, resulting in additional confounds. Each instruction was executed sequentially with a fixed duration of 20 seconds, and participants were always provided with a correct execution result for confirmation. Since the decision to accept the error or request a retry on execution involves subjective discrepancies, we consistently presented accurate simulations, eliminating the need for users to request retries.

5.3.3 Procedure

We used the equipment mentioned in Section 3.4. Before each session, the headset was sanitized using 70% isopropanol. Each participant was first welcomed by the study coordinator and presented with an information sheet. After giving informed consent, the participant was then introduced to the study flow and given the Stereo Optical Co. Inc. Stereo Fly Test (SFT) [44], which contains nine questions, to screen for stereo vision, the Ishihara Pseudo-Isochromatic Plate (PIP) test [17] to screen for color deficiencies, to screen for spatial ability. Two participants answered five of the SFT questions correctly, one answered six correctly, two answered seven correctly, one answered eight correctly, and the rest answered all correctly. For the PIP test, all participants answered all questions correctly. While the SFT and PIP results were not used to determine eligibility for the study, we use them to help explain the performance differences between participants in Section 6.1.

The study coordinator next explained the concept of high-level

Table 1: Number of outlier trials in each condition. Each task type under a condition has 34 trials in total.

	SA		AA		T	
	SG	CG	SG	CG	SG	CG
A	3	1	2	2	2	4
DA	0	1	0	1	3	0

instructions, followed by a demonstration of the task and three types of views. Next, the study coordinator put the headset on the participant and was handed one Quest 2 controller for their dominant hand. The study coordinator then started the study program. After starting the program, the study coordinator first calibrated the target positions to the height of the table. The participant then entered a "practice mode," in which the study coordinator explained the study mechanism and how to move objects, send instructions, and confirm executions. The participant completed three tasks, each in different view conditions, and proceeded to the formal study with 24 tasks. We computed the time the participant finished the tasks and will use it as the dependent variable in Section 6.1.

Throughout the study, the headset and controller tracking data and the interactions (Grab, Send, Confirm) were recorded. Over the session, the study coordinator monitored the participant's interaction through a separate display.

After finishing all tasks, the participant was asked to fill out a questionnaire that included questions on their demographics, an unweighted NASA TLX [14], and a request to rank the techniques based on their effectiveness. The TLX survey was modified to use a 1–7 scale [32], with 1 as best, rather than the original 0–20 scale. The use of a seven-point scale for workload estimate is justified by Ames and George [1]. Each participant rated the view conditions and line conditions for each TLX metric. Images of each visualization were displayed during the rating process to remind participants of the visualizations used. We gave only a single questionnaire at the end of the study. This helped reduce the length of the study, and avoided the issue that the participant's criteria for answering the questionnaire might shift over time. We use the TLX and the preference rankings results as the dependent variables in Section 6.2. The whole process took about 60 to 90 minutes for a typical participant to complete.

6 Results

6.1 Task Completion Time

The purpose of our study was to investigate the effectiveness of each visualization, as measured by trial completion time in Section 5.3.2. Once the study was concluded, we processed the completion-time results generated automatically by our system before analyzing them. To identify outliers, we applied Tukey's outlier filter [47], computing the *outside fence* individually for each condition and user. Trials that exceeded the value of the third quartile plus 1.5 times the interquartile range were considered outliers. We anticipated that the task type would have a significant impact on completion time, and we observed variations in performance among different users, so we computed the *outside fence* for each task type for each participant. Table 1 shows the number of outlier trials in each condition.

The average trial completion time after outlier removal under each condition is shown in Figure 12. Note that the trial completion time does not follow a normal distribution, as it was affected by the task difficulty and individual user performance. Therefore, rather than directly calculating the standard error of all trials, we used linear mixed-effects models to calculate the contribution of each factor. (See the linear mixed-effects model results in the supplementary material for the standard error.)

All hypotheses were evaluated with a significance level of $\alpha = .05$. We employed the MATLAB Statistics and Machine Learning Toolbox [31] to fit a linear mixed-effects model to our dataset. The trial completion time served as the observation, while the fixed-effect variables included the line type, the view, and the task type (assemble vs disassemble). Additionally, the random-effect variables consisted of task (defined as the initial and the final shapes assembled with the Soma pieces) and user ID. There is no interaction term between random-effect variables. We chose these variables after comparing the current model to alternative ones using a likelihood ratio test. Specifically, we compared two models, one with interaction terms between the fixed-effect variables and one without. The results of the likelihood ratio test showed that both models fit the data equally well, indicating a weak interaction between the fixed-effect variables. As such, we selected the model without interaction terms between the fixed-effect variables. By comparing different linear mixed-effects models, we found that neither gender, AR/VR experience, color vision, nor stereo vision are significant factors. This may be because the user ID random-effect term could absorb the impacts of gender, AR/VR experience, stereo vision, and even factors affecting user performance that were not measured in our study.

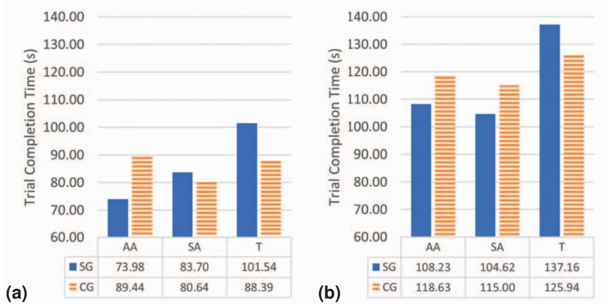

Figure 12: Formal study step-completion time. (a) A tasks. (b) DA tasks. We do not draw the error bars as the data do not follow a normal distribution.

That the interaction between the fixed-effect variables is weak indicates that while the disassembly task took the participants longer to finish, it did not cause a specific method to perform better or worse.

To test **H1**, we analyzed the p-values of the fixed-effect terms for visualization in our linear mixed-effects model. The p-values for the SA and the T visualizations are .6198 and .0002, respectively, indicating that participants completed the steps faster with AA compared to T. To evaluate whether the participants performed faster with SA than with T, we computed the contrast between the two. The resulting p-value is .0010, showing the participants performed faster with SA than with T. Therefore, our results support **H1**.

To evaluate **H2**, we analyzed the p-values of the fixed-effect terms for Line Type in our linear mixed-effects model. The p-value for the individual condition is .5892, indicating that the individual line type does not significantly improve the performance. Therefore, **H2** is not supported.

The p-values for the visualization and the task type fixed-effect terms are $< .0001$, providing strong evidence for their significant effects. The model summary can be found in the supplementary material. The model also yielded effect sizes of $\eta^2 = 0.3229$ (large) and Cohen's $d = 0.5592$ (medium) [4].

6.2 User Feedback

The participants were asked to rank different visualization styles based on their preferences (Figure 13). The results show that T was the most preferred style, followed by AA, and finally SA. We ran a Friedman test on the preference data, which shows that the differences in preference are significant ($p < .0001$). We further analyzed the data using Wilcoxon signed-rank tests to evaluate the difference in user preference between pairs. The results indicate that the user preference for T and AA was significantly higher than for

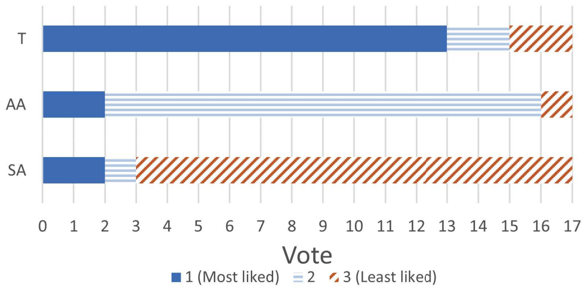

Figure 13: Preferences for visualization styles.

Table 2: Number of additional commits made in each condition across all participants.

	SA		AA		T	
	SG	CG	SG	CG	SG	CG
A	21	17	25	24	38	11
DA	20	18	23	21	24	15

SA, with p-values of .0049 and .0253, respectively. Therefore, H3. is supported. The user preference for AA is higher than SA, with a p-value of .0068.

To avoid type-I errors, we used the Holm–Bonferroni method [16]. We checked a total of four p-values for the validated hypotheses (two for **H1**, two for **H3**). They are .0002, .0010, .0049, and .0253. With this order, the p-values are smaller than $.05/4$, $.05/3$, $.05/2$, and $.05/1$, respectively, meaning they survive their corresponding Holm–Bonferroni-corrected α.

The unweighted NASA TLX results for our study are displayed in Figure 14, along with the p-values for each metric calculated using Friedman tests. The results indicate that the differences in physical demand, temporal demand, and effort are statistically significant. However, there is no significant difference in mental demand, performance, and frustration ratings between the visualizations.

To further examine the TLX results, we ran Wilcoxon signed-rank tests on the user ratings on the visualizations. The results show that T is more physically demanding than SA ($p = .0317$) and AA ($p = .0088$), and T is less temporally demanding than SA ($p = .0469$). All other pairs are insignificant. It is worth noting while the participants found T least temporally demanding than SA and AA, they spent more time on T than on SA and AA, as shown in Figure 12.

With regard to user preferences about the line styles, six participants preferred SG to CG, while the remaining 11 preferred CG to SG. A Friedman test showed that the difference in preference is not significant.

For the NASA TLX results, many participants gave the same rating for the two line styles. Friedman tests show no significant difference in the TLX metrics for the line styles. The summary of the TLX results for the line styles can be found in the supplementary materials.

6.3 Error Analysis

We analyzed the number of times that participants moved the Soma pieces to the wrong places, committed the goals, and needed to fix the goals and recommit them. We did this to gain insight into the errors made by participants in the study. Table 2 shows the number of additional commits caused by errors the participants made in each condition.

We compared the number of additional commits made in different groups of conditions. We first compared it by visualization. SA, AA, and T have 76, 93, and 88 errors, respectively. A Chi-square test showed that AA had more errors than SA ($p = .033$), but the effect size is small ($\phi = 0.18$). The differences between (AA, T) and (SA, T) are not significant.

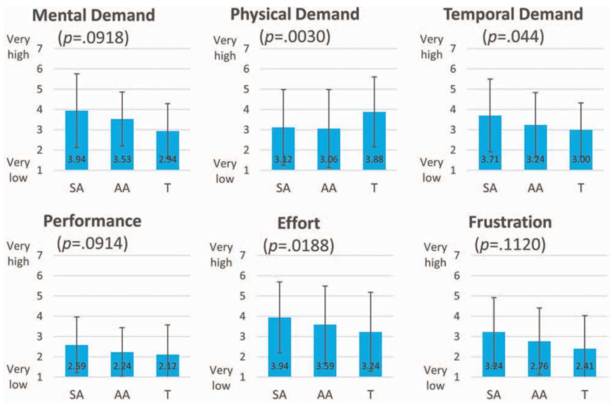

Figure 14: Unweighted NASA TLX results for the visualizations.

We then moved to compare the errors by line styles. The SG and the CG styles had 151 and 106 errors, respectively. A Chi-square test showed the difference is significant ($p < .0001$) and the effect size of the Chi-square test is medium ($\phi = 0.32$). While our hypothesis that CG would make participants perform faster than with SG (**H2**) was not supported, here the data show that the benefits might lie in a lower error rate.

7 Discussion

7.1 Comparison between the Aggregated Views and the Timeline View

The completion time results of our study support **H1**. Participants performed better in our tasks with visualizations aggregating the goals rather than showing them in different WIMs. Previous research has explored ways to visualize time-series data through aggregation [18] or through separate views [21]. We investigate the case of showing spatiotemporal history in VR. One possible explanation for the completion time results is that while separating the views can avoid visual clutters, it makes it harder for the users to follow the flow. Meanwhile, the subjective preference data support **H3**. The participants prefer T to SA and AA. One possible explanation is that the Aggregated Views made it easy to follow the flow, but it was more cluttered, and the participants needed to spend effort parsing them, which caused lower preference than T.

7.2 Comparison between the Static and Adaptive Aggregated Views

For the two Aggregated Views, the results in Sections 6.1 to 6.3 show that there is no significant difference in the completion time, the error rate is slightly higher in AA than in SA, and the study participants prefer AA to SA. One possible reason causing the two aggregated to have similar performance might be because, in our study, each Soma piece was moved only once or twice. If each piece is required to be moved more times, it is possible that the scene will become more cluttered, and the user will need to rely on the adaptive view mechanism to filter out unnecessary information. Future work can include examining the performance of the two views with more complicated tasks.

7.3 Comparison between Line Styles

Our completion time data does not support **H2**. Still, the error data in Section 6.3 shows that the line styles have an impact on the performance. With CG, the participants made fewer errors than with

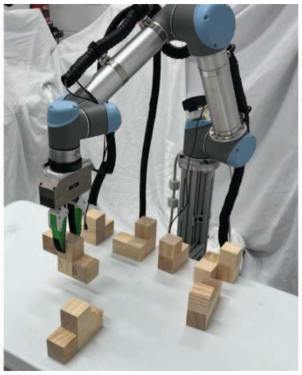

Figure 15: UR5 robot accomplishing assigned goals at a remote site in the same setting used in the user study with Soma cubes.

SG. This may be because while the work by Liu et al. [26] provides a strategy of helping users to distinguish visualizations by using size, brightness, and transparency, the time limitation in our task is slightly different from theirs. In our task, the user did not need to work on the task objects linked by the lines very fast. They only need to understand them and can work on another different object. In our task, the benefit lies in a lower error rate rather than in the completion time data.

7.4 Lessons Learned

From our user study, we found the two Aggregated Views help the user perform faster than the Timeline View. On the other hand, the users prefer the Timeline View to the two Aggregated Views. Based on these findings, we suggest that when creating visualizations for showing multiple steps for different time points, one should consider properly aggregating the visualizations in a single view. However, for a specific application, one might need to conduct a small-scale study to make sure the level of aggregation is acceptable to the users.

In addition, that SG caused more errors than CG suggests it is helpful to show the overall task flow using lines. This is similar to what was suggested in Liu et al. [26], though the task domains are different. When showing sequential information, one might consider using the same approach of using different sizes and colors to help users distinguish successive visualizations.

8 Prototype with UR5 Robot

To demonstrate that our system works with a real robot, we integrated our system with a physical UR5 robot arm equipped with a parallel gripper. We made physical Soma pieces with 5.08cm×5.08cm×5.08cm wooden blocks (Figure 15). We ran the VR user interface with the UR5 to manipulate the physical Soma pieces, as demonstrated in the supplementary video.

9 Limitations and Future Work

9.1 Subject Population

The participants in our study were recruited from our institution, and all were relatively young (Section 5.3.1). Thus it is important to note that further research is required to establish whether our findings are applicable to older people. In addition, the proportion of male participants is relatively high. However, the results in Section 6.1 do not indicate a relationship between gender and performance. Furthermore, our study employed a single-session design and thus did not measure the performance of trained users, for whom it is possible that the results will be different. We would like to address these limitations in the future.

9.2 User Studies with Real Robot

Our VR user study does not use a real robot. We will be conducting user studies that incorporate the path planner and the robot shown in Section 8. To achieve this, we will integrate real-time sensor feedback from the remote site so that the user can judge the results to decide whether a goal needs to be re-executed.

9.3 Working with Multiple Robots at Multiple Sites

Asynchronous control, where users specify future steps, also allows human–robot collaboration systems to work with multiple robots at multiple sites, such as the scenario considered in the testbed by Liu et al. [27]. In such a case, multiple steps will be executed concurrently, while the user can work on and specify only one or two steps at a time. Managing and monitoring multiple robots at multiple sites is multitasking and is challenging. Future work can include designing user interfaces to help the user seamlessly transition from assigning tasks at one site to assigning tasks to another site.

9.4 Interfaces for Multiple Actions

Our user interface focuses on handling multiple assembly goals. Some systems, such as the one developed by Senft et al. [43], allow the user to assign simple actions such as tightening a screw. Such actions cannot be visualized simply with the task objects' final pose. In this case, additional investigation might be needed for designing interfaces to visualize these actions. As visualizations for showing actions can be more complicated for the user to understand, aggregating visualizations could create a more cluttered scene.

10 Conclusions

We explored two approaches for asynchronously assigning, monitoring, and managing assembly goals in VR for high-level robot teleoperation. The Aggregated Views aggregate the visualizations, while the Timeline View shows each step in a separate WIM. Through a user study, we showed that participants performed better with the Aggregated Views than with the Timeline View but preferred the Timeline View. We did not find differences in the impact of different line styles on task completion time or participants' preferences. However, the error analysis showed that participants made significantly more errors with the SG style than with the CG style. The findings can be applied to future designs for multi-step goal-based teleoperation.

Acknowledgments

This research was funded in part by National Science Foundation Grant CMMI-2037101. We thank the participants in our studies and the anonymous reviewers of our paper for their feedback.

References

[1] L. L. Ames and E. J. George. Revision and verification of a seven-point workload estimate scale. *Technical Information Manual, Air Force Flight Test Center, Edwards Air Force Base*, 1993.

[2] S. Arevalo Arboleda, F. Rücker, T. Dierks, and J. Gerken. Assisting manipulation and grasping in robot teleoperation with augmented reality visual cues. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21. Association for Computing Machinery, New York, NY, USA, 2021. doi: 10.1145/3411764.3445398

[3] J. M. Beer, A. D. Fisk, and W. A. Rogers. Toward a framework for levels of robot autonomy in human–robot interaction. *J. Hum.-Robot Interact.*, 3(2):74–99, jul 2014. doi: 10.5898/JHRI.3.2.Beer

[4] M. Brysbaert and M. Stevens. Power analysis and effect size in mixed effects models: A tutorial. *Journal of Cognition*, 1(1), 2018. doi: 10.5334/joc.10

[5] A. Burghardt, D. Szybicki, P. Gierlak, K. Kurc, P. Pietruś, and R. Cygan. Programming of industrial robots using virtual reality and digital twins. *Applied Sciences*, 10(2), 2020. doi: 10.3390/app10020486

[6] W. Büschel, A. Lehmann, and R. Dachselt. MIRIA: A mixed reality toolkit for the in-situ visualization and analysis of spatio-temporal interaction data. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21. Association for Computing Machinery, New York, NY, USA, 2021. doi: 10.1145/3411764.3445651

[7] E. Coumans and Y. Bai. PyBullet, a Python module for physics simulation for games, robotics and machine learning. `http://pybullet.org`, 2016–2023.

[8] J. D. Denning, V. Tibaldo, and F. Pellacini. 3DFlow: Continuous summarization of mesh editing workflows. *ACM Trans. Graph.*, 34(4), 2015. doi: 10.1145/2766936

[9] H. Fang, S. Ong, and A. Nee. Interactive robot trajectory planning and simulation using augmented reality. *Robotics and Computer-Integrated Manufacturing*, 28(2):227–237, 2012. doi: 10.1016/j.rcim.2011.09.003

[10] I. Farkhatdinov and J.-H. Ryu. Teleoperation of multi-robot and multi-property systems. In *2008 6th IEEE International Conference on Industrial Informatics*, pp. 1453–1458, 2008. doi: 10.1109/INDIN.2008.4618333

[11] G. Fouché, F. Argelaguet Sanz, E. Faure, and C. Kervrann. Timeline design space for immersive exploration of time-varying spatial 3D data. In *Proceedings of the 28th ACM Symposium on Virtual Reality Software and Technology*, VRST '22. Association for Computing Machinery, New York, NY, USA, 2022. doi: 10.1145/3562939.3565612

[12] A. Franzluebbers and K. Johnson. Remote robotic arm teleoperation through virtual reality. In *Symposium on Spatial User Interaction*, SUI '19. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3357251.3359444

[13] S. Grzonka, G. Grisetti, and W. Burgard. A fully autonomous indoor quadrotor. *IEEE Transactions on Robotics*, 28(1):90–100, 2012. doi: 10.1109/TRO.2011.2162999

[14] S. G. Hart and L. E. Staveland. Development of NASA-TLX (task load index): Results of empirical and theoretical research. In P. A. Hancock and N. Meshkati, eds., *Human Mental Workload*, vol. 52 of *Advances in Psychology*, pp. 139–183. North-Holland, 1988. doi: 10.1016/S0166-4115(08)62386-9

[15] P. Higgins, R. Barron, D. Engel, and C. Matuszek. Lessons from a small-scale robot joining experiment in VR. In *2023 ACM/IEEE International Conference on Human–Robot Interaction Workshops*, 2023.

[16] S. Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

[17] S. Ishihara. *Ishihara's Tests for Colour-blindness*. Kanehara Shuppan, 1972.

[18] U. Jugel, Z. Jerzak, G. Hackenbroich, and V. Markl. M4: A visualization-oriented time series data aggregation. *Proc. VLDB Endow.*, 7(10):797–808, jun 2014. doi: 10.14778/2732951.2732953

[19] J.-H. Kim, G. Sharma, and S. S. Iyengar. Famper: A fully autonomous mobile robot for pipeline exploration. In *2010 IEEE International Conference on Industrial Technology*, pp. 517–523, 2010. doi: 10.1109/ICIT.2010.5472748

[20] O. Kroemer, S. Niekum, and G. Konidaris. A review of robot learning for manipulation: Challenges, representations, and algorithms. *J. Mach. Learn. Res.*, 22(1), 2022.

[21] D. Kurlander and S. Feiner. A history-based macro by example system. In *Proceedings of the 5th Annual ACM Symposium on User Interface Software and Technology*, UIST '92, p. 99–106. Association for Computing Machinery, New York, NY, USA, 1992. doi: 10.1145/142621.142633

[22] S. M. LaValle. *Rapidly-exploring random trees*. `http://lavalle.pl/rrt/`, last accessed on 09/01/2023.

[23] Y. Li, S. Agrawal, J.-S. Liu, S. K. Feiner, and S. Song. Scene editing as teleoperation: A case study in 6DoF kit assembly. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4773–4780, 2022. doi: 10.1109/IROS47612.2022.9982158

[24] K. Lilija, H. Pohl, and K. Hornbæk. Who put that there? Temporal navigation of spatial recordings by direct manipulation. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, p. 1–11. Association for Computing Machinery, New York, NY, USA, 2020. doi: 10.1145/3313831.3376604

[25] J. I. Lipton, A. J. Fay, and D. Rus. Baxter's homunculus: Virtual reality spaces for teleoperation in manufacturing. *IEEE Robotics and Automation Letters*, 3(1):179–186, 2018. doi: 10.1109/LRA.2017.2737046

[26] J.-S. Liu, C. Elvezio, B. Tversky, and S. Feiner. Using multi-level precueing to improve performance in path-following tasks in virtual reality. *IEEE Transactions on Visualization and Computer Graphics*, 27(11):4311–4320, 2021. doi: 10.1109/TVCG.2021.3106476

[27] J.-S. Liu, C. Wang, B. Tversky, and S. Feiner. A testbed for exploring virtual reality user interfaces for assigning tasks to agents at multiple sites. In *Proceedings of the 2023 ACM Symposium on Spatial User Interaction*, SUI '23. Association for Computing Machinery, New York, NY, USA, 2023. doi: 10.1145/3607822.3618004

[28] K. Liu, D. Sakamoto, M. Inami, and T. Igarashi. Roboshop: Multi-layered sketching interface for robot housework assignment and management. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, p. 647–656. Association for Computing Machinery, New York, NY, USA, 2011. doi: 10.1145/1978942.1979035

[29] K. Mahadevan, Y. Chen, M. Cakmak, A. Tang, and T. Grossman. Mimic: In-situ recording and re-use of demonstrations to support robot teleoperation. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, UIST '22. Association for Computing Machinery, New York, NY, USA, 2022. doi: 10.1145/3526113.3545639

[30] K. Mahadevan, Q. Zhou, G. Fitzmaurice, T. Grossman, and F. Anderson. Tesseract: Querying spatial design recordings by manipulating worlds in miniature. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23. Association for Computing Machinery, New York, NY, USA, 2023. doi: 10.1145/3544548.3580876

[31] The MathWorks, Inc. *Matlab Statistics and Machine Learning Toolbox*. `https://www.mathworks.com/help/stats/index.html`, last accessed on 09/30/2023.

[32] MeasuringU. *10 Things to Know about the NASA TLX*. `https://measuringu.com/nasa-tlx/`, last accessed on 06/04/2023.

[33] L. Meng, J. Liu, W. Chai, J. Wang, and M. Q.-H. Meng. Virtual reality based robot teleoperation via human–scene interaction. *arXiv preprint arXiv:2308.01164*, 2023. doi: 10.48550/arXiv.2308.01164

[34] Meta. *Meta Quest 2*. `https://www.meta.com/quest/products/quest-2/`, last accessed on 09/01/2023.

[35] A. Naceri, D. Mazzanti, J. Bimbo, Y. T. Tefera, D. Prattichizzo, D. G. Caldwell, L. S. Mattos, and N. Deshpande. The Vicarios virtual reality interface for remote robotic teleoperation: Teleporting for intuitive tele-manipulation. *Journal of Intelligent & Robotic Systems*, 101, 2021.

[36] A. A. F. Nassiraei, Y. Kawamura, A. Ahrary, Y. Mikuriya, and K. Ishii. Concept and design of a fully autonomous sewer pipe inspection mobile robot "KANTARO". In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pp. 136–143, 2007. doi: 10.1109/ROBOT.2007.363777

[37] M. Okabe and K. Ito. *Color Universal Design (CUD)—How to make figures and presentations that are friendly to colorblind people*. `https://jfly.uni-koeln.de/color/`, last accessed on 09/30/2023.

[38] M. Pascher, K. Kronhardt, and J. Gerken. PhysicalTwin: Mixed reality interaction environment for ai-supported assistive robots. In *2023 ACM/IEEE International Conference on Human–Robot Interaction Workshops*, 2023.

[39] C. Perin, T. Wun, R. Pusch, and S. Carpendale. Assessing the graphical perception of time and speed on 2D+time trajectories. *IEEE Transactions on Visualization and Computer Graphics*, 24(01):698–708, jan 2018. doi: 10.1109/TVCG.2017.2743918

[40] A. Poncela and L. Gallardo-Estrella. Command-based voice teleoperation of a mobile robot via a human–robot interface. *Robotica*, 33(1):1–18, 2015. doi: 10.1017/S0263574714000010

[41] D. J. Rea and S. H. Seo. Still not solved: A call for renewed focus on user-centered teleoperation interfaces. *Frontiers in Robotics and AI*, 9:704225, 2022. doi: 10.3389/frobt.2022.704225

[42] D. Sakamoto, K. Honda, M. Inami, and T. Igarashi. Sketch and run: A

stroke-based interface for home robots. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, p. 197–200. Association for Computing Machinery, New York, NY, USA, 2009. doi: 10.1145/1518701.1518733

[43] E. Senft, M. Hagenow, K. Welsh, R. Radwin, M. Zinn, M. Gleicher, and B. Mutlu. Task-level authoring for remote robot teleoperation. *Frontiers in Robotics and AI*, 8:1–15, 2021. doi: 10.3389/frobt.2021.707149

[44] Stereo Optical Co. Inc. *Original Stereo Fly Stereotest*. https://www.stereooptical.com/products/stereotests-color-tests/original-stereo-fly/, last accessed on 05/31/2022.

[45] R. Stoakley, M. J. Conway, and R. Pausch. Virtual reality on a WIM: Interactive worlds in miniature. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, p. 265–272. ACM Press/Addison-Wesley Publishing Co., USA, 1995. doi: 10.1145/223904.223938

[46] S. L. Su, S. Paris, F. Aliaga, C. Scull, S. Johnson, and F. Durand. Interactive visual histories for vector graphics. *MIT CSAIL Technical Reports*, 2009.

[47] J. W. Tukey. *Exploratory data analysis*, vol. 2. Reading, Mass., 1977.

[48] Unity. *Unity*. https://unity.com/, last accessed on 09/01/2023.

[49] Universal Robots. *UR5*. https://www.universal-robots.com/products/ur5-robot/, last accessed on 09/01/2023.

[50] M. E. Walker, H. Hedayati, and D. Szafir. Robot teleoperation with augmented reality virtual surrogates. In *2019 14th ACM/IEEE International Conference on Human–Robot Interaction (HRI)*, pp. 202–210, 2019. doi: 10.1109/HRI.2019.8673306

[51] D. Whitney, E. Rosen, E. Phillips, G. Konidaris, and S. Tellex. Comparing robot grasping teleoperation across desktop and virtual reality with ROS reality. In N. M. Amato, G. Hager, S. Thomas, and M. Torres-Torriti, eds., *Robotics Research*, pp. 335–350. Springer International Publishing, Cham, 2020. doi: 10.1007/978-3-030-28619-4_28

[52] E. Yigitbas, K. Karakaya, I. Jovanovikj, and G. Engels. Enhancing human-in-the-loop adaptive systems through digital twins and VR interfaces. In *2021 International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, pp. 30–40, 2021. doi: 10.1109/SEAMS51251.2021.00015

[53] L. Zhang, A. Agrawal, S. Oney, and A. Guo. VRGit: A version control system for collaborative content creation in virtual reality. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23. Association for Computing Machinery, New York, NY, USA, 2023. doi: 10.1145/3544548.3581136

[54] N. Zhang, T. Qi, and Y. Zhao. Real-time learning and recognition of assembly activities based on virtual reality demonstration. *Sensors*, 21(18), 2021. doi: 10.3390/s21186201

[55] Y. Zhang, B. Ens, K. A. Satriadi, A. Prouzeau, and S. Goodwin. Timetables: Embodied exploration of immersive spatio-temporal data. In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 599–605, 2022. doi: 10.1109/VR51125.2022.00080