Periscope: A Robotic Camera System to Support Remote Physical Collaboration

PRAGATHI PRAVEENA, Department of Computer Sciences, University of Wisconsin–Madison, USA
YEPING WANG, Department of Computer Sciences, University of Wisconsin–Madison, USA
EMMANUEL SENFT, Idiap Research Institute, Switzerland
MICHAEL GLEICHER, Department of Computer Sciences, University of Wisconsin–Madison, USA
BILGE MUTLU, Department of Computer Sciences, University of Wisconsin–Madison, USA

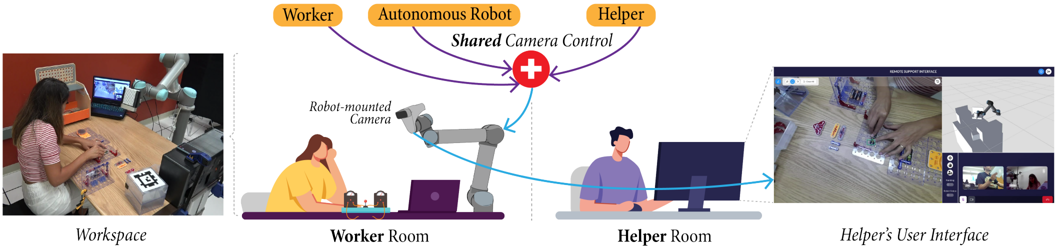

Fig. 1. This paper introduces *Periscope*, a robotic camera system that allows two people to collaborate remotely on physical tasks. With *Periscope*, a local worker can complete an assembly task with guidance from a remote helper who views the workspace through a robot-mounted camera. We use a *shared camera control* approach in which the worker, the helper, and the autonomous robot all contribute to camera control and design a set of modes that uniquely combine inputs from these three sources to move the camera. *Periscope* facilitates remote collaboration by providing the worker and the helper with shared visual information that enhances their verbal communication and coordination processes.

We investigate how robotic camera systems can offer new capabilities to computer-supported cooperative work through the design, development, and evaluation of a prototype system called *Periscope*. With *Periscope*, a local worker completes manipulation tasks with guidance from a remote helper who observes the workspace through a camera mounted on a semi-autonomous robotic arm that is co-located with the worker. Our key insight is that the helper, the worker, and the robot should all share responsibility of the camera view—an approach we call *shared camera control*. Using this approach, we present a set of modes that distribute the control of the camera between the human collaborators and the autonomous robot depending on task needs. We demonstrate the system's utility and the promise of shared camera control through a preliminary study where 12 dyads collaboratively worked on assembly tasks and discuss design and research implications of our work for future robotic camera system that facilitate remote collaboration.

Authors' addresses: Pragathi Praveena, Department of Computer Sciences, University of Wisconsin–Madison, Madison, WI, USA, pragathi@cs.wisc.edu; Yeping Wang, Department of Computer Sciences, University of Wisconsin–Madison, Madison, WI, USA, yeping@cs.wisc.edu; Emmanuel Senft, Idiap Research Institute, Martigny, Switzerland, esenft@idiap.ch; Michael Gleicher, Department of Computer Sciences, University of Wisconsin–Madison, Madison, WI, USA, gleicher@cs.wisc.edu; Bilge Mutlu, Department of Computer Sciences, University of Wisconsin–Madison, Madison, WI, USA, bilge@cs.wisc.edu.

XXXX-XXXX/2023/5-ART $15.00
https://doi.org/XXXXXXX.XXXXXXX

CCS Concepts: • **Computer systems organization → Robotics**; • **Human-centered computing →** *Interactive systems and tools*; **Computer supported cooperative work**.

Additional Key Words and Phrases: human-robot interaction, remote collaboration, shared control

ACM Reference Format:
Pragathi Praveena, Yeping Wang, Emmanuel Senft, Michael Gleicher, and Bilge Mutlu. 2023. *Periscope*: A Robotic Camera System to Support Remote Physical Collaboration. 1, 1 (May 2023), 35 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Remote collaboration on physical tasks is valuable in scenarios such as experts assisting novices with manual assembly or repair tasks, particularly when it is inconvenient, time-consuming, or expensive to travel and assist someone in person. For example, a field technician might seek guidance from an expert to repair a wind turbine; an expert might provide training to car mechanics on how to repair a new engine model; or an astronaut might get help from ground control to maintain critical infrastructure on the space station. Such scenarios typically involve a local "worker" manipulating physical artifacts with guidance from a remote "helper." The helper views the workspace through one or more cameras, which may be fixed or movable. Ideally, the helper is able to observe various key sources of information including the worker, the task objects, and the environment [45]. Additionally, the requirements on these views may change over the course of the task [23]. For example, the helper monitors the worker's actions during assembly, recognizes incorrect actions, and intervenes with new instructions, which requires looking at task objects while attempting to identify the component required for the next step. Finally, the helper may need to examine artifacts in the workspace from various angles, such as the interior of a drawer or the top of an object, and in varying levels of detail, such as a close-up view to see fine details or a wide-angle view to see more context [47]. A core challenge for technologies that facilitate remote collaboration is *providing the helper with diverse, informative, and task-relevant views*, which is not only critical for the helper to maintain awareness throughout the task but also for the helper and the worker to develop a shared understanding during the collaboration process [21].

The focus of recent research on remote collaboration in human-computer interaction (HCI) and computer-supported cooperative work (CSCW) has been on Virtual Reality approaches that give the helper the freedom to independently explore a reconstructed version of the worker's environment using a virtual camera (see Schäfer et al. [79] for a review). These reconstructed workspaces can afford a high level of immersion and viewpoint flexibility, but they lack the dynamically changing details that are necessary for real-time collaboration. Other approaches involve cameras that stream directly from the real world, providing dynamic information from the task environment. However, these cameras are often limited to fixed viewpoints or viewpoints controlled solely by the worker (e.g., a head-worn camera), which can impede collaborative processes such as monitoring task status, observing worker's actions and comprehension, establishing joint attention, and formulating messages [21, 22]. One potential solution that combines a high level of viewpoint flexibility and real-time, dynamic information through a live stream is the use of *robotic cameras*.

Modern *collaborative robot*, or *cobot*, platforms, augmented with cameras, can move with many degrees of freedom (DoF), supporting precise camera control for complex tasks and environments while maintaining safety for co-located human interaction. Despite their potential, such robots with high kinematic capabilities have rarely been utilized in robotic camera systems that support remote collaboration [17]. Giving direct control of a high-DoF robotic camera to the helper presents challenges related to designing control schemes that meaningfully link the user's inputs to robot movements. Controlling a low-DoF camera, such as a pan-tilt camera, is relatively simple with 2D controls that are directly mapped to the camera's movement. However, applying such methods to controlling a high-DoF camera in order to obtain precise views, such as looking into a drawer, is not straightforward to implement, as it requires mapping the helper's view intent to the camera's full 6-DoF pose (position and orientation). On the other hand, *autonomous* camera control, particularly determining what the robot should be looking at at any given time during the collaboration, is an open question. In this work, we address the challenge of designing direct and autonomous camera control that enables the use of high-DoF robotic cameras for remote collaboration.

Prior literature suggests that both the helper and the worker may require control of the camera view at different points of the collaboration process, such as to provide guidance or ask questions [49, 59]. Therefore, a robotic system for remote collaboration must permit both the helper and the worker to modify the camera view. However, moving the camera is only a secondary activity for the helper and the worker, whose primary goal is to complete a collaborative physical task. Offloading some of the camera control to an autonomous robot can allow collaborators to devote more of their attention to the primary goal. Thus, the system should allow the robot to assume part of the workload of camera control by making autonomous adjustments to the camera view as needed while also allowing control of the view by the helper and the worker. We call this approach *shared camera control* (based on a robot control paradigm called *shared control* [53]) and investigate how robotic camera systems can leverage this approach to offer new capabilities to CSCW through the design, development, and evaluation of a prototype system called *Periscope* (see Figure 1).

The *Periscope* system supports a worker in completing physical tasks with remote guidance from a helper who observes the workspace through a robot-mounted camera. The camera view is displayed on a screen interface for both the worker and the helper, enabling them to share task-relevant visual information and develop a mutual understanding during the collaboration process. We design camera controls to empower both the helper and the worker to independently control the view depending on the needs of the task, but also allow the robot to assist and reduce their effort. Our system is centered around five design goals: (1) *versatility* to support camera views for various task activities; (2) *intuitivity* to simplify camera control for users through intuitive mappings and autonomous behaviors; (3) *dual-user interactivity* to allow both the helper and the worker to modify the camera view; (4) *congruity* to arbitrate user interactions and autonomous behaviors to reach consensus; and (5) *usability* to support general communication and functional requirements. To balance these five design goals, we designed three modes that uniquely distribute camera control among the worker, the helper, and the autonomous robot. These modes serve as an initial point of inquiry for understanding the promise of shared camera control for facilitating remote collaboration. Through shared camera control, we tackle the challenge of simplifying the control of a high-DoF robotic camera and providing users with diverse, informative, and task-relevant views.

We conducted a preliminary evaluation of the *Periscope* system with 12 dyads in a lab study to understand how the system supports remote collaboration. During a 2-hour session, each dyad collaboratively worked on assembly tasks while physically located in separate rooms. From our analysis of recorded video data of the collaboration, we present use patterns for the system's features that illustrate the individual value of each mode and the rich interactions enabled by transitioning between the modes. Based on these results, we present reflections on our design goals and design implications for future robotic camera systems. Our work makes key contributions in four categories, *Design (§3.1)*, *System (§3.4, §3.5)*, *Data (§5, §6.1)*, and *Recommendations (§6.2)*:

(1) *Design* — the shared camera control approach and a set of design goals to realize this approach.
(2) *System* — *Periscope*, a robotic camera system that is an instantiation of shared camera control.
(3) *Data* — empirical observations on system use and their contribution to the design goals.
(4) *Recommendations* — design implications for robotic camera-based CSCW systems.

2 RELATED WORK

In this section, we discuss prior research that identifies how *shared visual context* is essential for successful collaboration. Then, we review systems that provide *technological support for remote collaboration*, including robotic systems. Finally, we discuss existing *control frameworks for cameras and robots* that we use to develop our shared-camera-control system.

2.1 Shared Visual Context

During synchronous collaboration (both co-located and remote), verbal communication is the primary medium through which information is exchanged [20, 45]. *Shared visual context* [21] or task-relevant visual information that the collaborators have in common augments verbal communication and improves collaborative outcomes. Findings from studies [15, 20, 45, 85] suggest that people use the shared visual context for two coordination processes: *situation awareness* and *conversational grounding*. According to situation awareness theory by Endsley [18], shared visual information helps people to establish an up-to-date mental model of the state of the task, the environment, and their partner, which can help the pair to plan future actions. According to conversational grounding theory by Clark and Marshall [13], shared visual information supports verbal communication by providing an alternative and rich source of information that contributes to the development of a mutual understanding between collaborators, resulting in more efficient conversation. When collaborating on physical artifacts, shared visual information can particularly help the pair achieve *joint attention* [7], where they have a shared focus on an object.

An example of remote collaboration from Kraut et al. [45] illustrates the use of shared visual context for situation awareness and conversational grounding. In this example, a helper guides a worker in adjusting the inclination of a bicycle seat during a repair task. The helper uses the shared visual context to gain situation awareness about the current state of the worker, the task, and the environment, allowing them to acknowledge the state (e.g., "*Cool*") and plan next steps (e.g., "*next go on* and adjust it" and "angle the nose up *a little bit more*"). The shared visual information also supports conversational grounding and joint attention, as both the helper and the worker use definite articles (e.g., "*the* bar" and "*the* nose") and deixis [51] (e.g., "*this* bar *here*") that require contextual information to be fully understood. The worker's verbal responses (e.g., "*Is that good?*") and actions (e.g., *Adjusts seat*) indicate their understanding of the helper's instructions and further contribute to the grounding process.

Helper: *Uh- next go on and adjust it so it's parallel to the bar- the top*
Worker: *This bar here? Is that good?*
Helper: *Uh- angle the nose up a little bit more.*
Worker: *[Adjusts seat]*
Helper: *Cool.*

2.2 Technological Support for Remote Collaboration

Systems that support remote collaboration facilitate the sharing of visual context to enable effective cooperation and communication between users (see Druta et al. [17] for a review). Our work draws from design choices made in other systems that support two remote users seeking to accomplish synchronous collaboration over physical artifacts. We divide the review of prior work into four categories — (1) technologies for visual information capture, (2) technologies for visual information display, (3) technologies for communication cues, and (4) robotic systems for collaboration — and provide examples of systems that fall under each category. We discuss relevant opportunities and challenges of different technologies for providing a shared visual context between collaborators.

2.2.1 Technologies for Visual Information Capture: Prior systems have used fixed-view cameras [21, 25, 42], head-mounted cameras [22, 31, 36], shoulder-mounted cameras [46, 66], hand-held cameras [26, 58, 82], multiple cameras [22, 29, 75], pan-tilt-zoom (PTZ) cameras [64, 74], 360° cameras [38, 66] or depth cameras [3, 87] to capture visual information about the workspace. These sensors and additional eye-tracking or head-tracking technology may also be used for capturing information about the worker or helper [31, 86, 91] (see Xiao et al. [92] for a review).

Early remote collaboration systems mostly relied on views from fixed cameras or worker-worn cameras (e.g., head-mounted or hand-held cameras), which the helper could not modify independently. These approaches can disrupt collaboration because the helper has to repeatedly interrupt the worker while they are performing task-related activities and direct them to change the view. Recent research focuses on enabling the helper to independently view the workspace via remote control of physical cameras (e.g., PTZ cameras) or virtual cameras (e.g., in 3D reconstructed workspaces). Although this approach increases the system's complexity, granting the helper control over the view enables them to have diverse and independent views of the workspace.

2.2.2 Technologies for Visual Information Display: Prior systems have used 2D view [21, 25, 42, 74], 3D view [3, 27], 360° view [38, 50], Virtual Reality (VR) [86], Augmented Reality (AR) [31, 36, 82], Mixed Reality (MR) [5, 61, 66, 87, 88], and projected AR [32, 56, 83] for the display of shared visual information (see [79] for a review on VR, AR, and MR systems). AR and projected AR are typically used for situated information display to the worker who handles the physical artifacts.

While VR, AR, and MR solutions provide users with a highly immersive experience, high-quality virtual reconstructions can be difficult to update in real-time, require significant bandwidth, and may lack the fine and dynamically changing details that are necessary for many physical tasks. In such scenarios, live 2D or 360° video may be superior. Additionally, these approaches can be mixed together [87] to leverage the benefits and reduce the drawbacks of each approach.

2.2.3 Technologies for Communication Cues: The primary communication channels in remote collaboration systems are typically visual and verbal. Additionally, Fussell et al. [25] recommend that gestures used by helpers should be captured by collaboration systems to support referential communication. These gestures may be captured through vision-based or IMU-based hand tracking [5, 86, 87] or specified through annotations [24, 27, 41]. The gestures are then relayed to the worker by overlaying graphics on the shared view. This includes 2D graphic overlays on 2D views, 3D graphic overlays in MR, and projections onto the physical world in projected AR. Prior works have found improved collaborative outcomes when gestures are combined with visualizations of the helper's eye gaze [4, 5], the worker's eye gaze [31, 78], or viewing direction [33, 63]. Other interesting communication cues include virtual replicas of task objects [61] or human avatars to provide non-verbal cues in MR [65].

2.2.4 Robotic Systems for Collaboration: Prior work has explored how robots can facilitate remote collaboration. These works mostly focus on enabling the helper to control a robot-mounted camera in low-DoF settings and do not fully explore the possibility of robot autonomy, control of the robot by the worker, or how to manage the complexity of sharing camera control among the helper, the worker, and the robot. Thus, they do not leverage the full potential of a robotic platform for the formation of a co-constructed visual context. We address this gap in our work.

Early work by Kuzuoka et al. [47] demonstrated that granting the helper independent control of a 3-DoF robotic camera enabled the helper to explore 3D workspaces and examine physical artifacts from various angles. More recently, Feick et al. [19] used a robotic arm to reproduce orientation manipulations on a proxy object at a remote site. While this solution improves spatial understanding of the object, it is hard to scale beyond one object. Gurevich et al. [32] and Machino et al. [56]

designed systems that used a robot-mounted camera and projector (to capture the workspace and project on top of it), and showed that the mobility of the system improved collaborative outcomes. These systems allowed the helper to control the robot, but there was no exploration of robot autonomy or worker control of the robot. Sirkin and Ju [81] and Onishi et al. [62] explored the use of a robotic arm to display gestures such as pointing to and touching remote objects but not to capture any visual information about the workspace.

Telepresence robots make up a special case of robots designed to support collaboration by emulating face-to-face communication in a remote setting. The prototypical telepresence robot is a screen on wheels that is roughly human-sized in height with a camera and microphone. An interface will typically allow the remote user (in rare cases such as [67], the local user) control over the movement of the telepresence robot and the positioning of the cameras. These robots improve collaborative outcomes through the provision of a physical embodiment [48, 67] that enhances the feeling of presence or “being there” for the remote user, and improves the local user’s sense of the remote user’s presence [10]. There is a rich literature (e.g., [37, 43, 44, 68, 69, 84, 89]) on how telepresence robots and interfaces should be designed to support communication between remote users. However, these design choices are constrained by the anthropomorphic treatment of the robot as the remote user’s surrogate. Non-anthropomorphic form factors, such as a robotic arm, offer a different design space of interaction techniques that can leverage robot autonomy and control by the co-located user to support remote collaborative work.

Researchers have also explored other form factors for telepresence robots, such as drones [76, 93] or tabletop robots [2, 77]. While these systems are typically designed for interpersonal communication, some recent works [52, 90] have addressed the use of tabletop robots for supporting collaboration in remote physical tasks. Villanueva et al. [90] designed a tabletop robot that can be controlled by a remote instructor to provide in-situ advice on basic electrical circuitry to students. Li et al. [52] used a swarm of tabletop robots with cameras to allow several remote persons to view physical skill demonstrations by an instructor. The remote audience members can view the workspace through automatic and manual navigation of the robots and the instructor can physically move the robots for camera repositioning. These systems are advancing the possibilities of robotic platforms. However, the workspaces in these prior works are relatively level and uncluttered where low-DoF tabletop robots are adequate. Our work leverages high-DoF robot arms that can be used for precise camera control to support scenarios with clutter and complex geometry and allow remote users to achieve specific views such as the interior of a drawer or the top of an object. We draw from state-of-the-art camera and robot control frameworks (discussed next in §2.3) to limit the effort needed for the control of high-DoF robots in 3D environments.

2.3 Control Frameworks for Cameras and Robots

Relevant to the goals of this paper is prior literature on state-of-the-art camera control methods and control mechanisms for robots when humans and robots work in a collaborative ecosystem, which can inform the design of a robotic camera to support remote collaboration between individuals.

2.3.1 Camera Control: Christie et al. [12] describe various challenges associated with camera control. Designing control schemes for direct control of the camera by the user is challenging because users can find it difficult to deal simultaneously with all of the camera’s degrees of freedom. Consequently, control schemes must provide mappings that meaningfully link the user’s actions to the camera parameters. On the other hand, it is also challenging to partially or fully automate camera movement because the geometric specification of the camera pose needs to result in a semantically meaningful view for the user. Thus, our work draws from various manual and automated camera control techniques such as visual servoing [9, 34], through-the-lens camera control [30], assisted camera control in virtual environments [12], and automatic cinematography [11] to make it easier for the helper and the worker to influence the shared view.

Visual servoing [34] is a robot control method using features extracted from vision data (from a camera) to define a target pose for the robot and determine how the robot should move. Through-the-lens camera control [30] is a technique where a camera view is specified through controls in the image plane, essentially mapping visual goals to camera movements. Methods that provide assisted camera control in virtual environments [12] use knowledge of the environment to assist the user with camera control. For example, if the camera maintains a fixed distance around an object when it is being inspected, it results in the camera orbiting around the object in response to user inputs. Techniques for automatic cinematography [11] enable automatic tracking of a person (or their face or hands) to keep them in view. This has been utilized both in research prototypes of remote collaboration systems, for instance, hand tracking in Ranjan et al. [74], and commercial video conferencing products such as Apple Center Stage[1] and Lumens Auto Tracking Camera.[2]

In his paper on remote collaboration systems, Gaver [28] asserts that unless the cost of gaining additional information is low enough, it will not seem worth the additional effort for users. Our work is guided by this idea of allowing both the helper and the worker to move the camera with low cognitive and physical costs.

2.3.2 Shared Control: Shared control is a robot control paradigm where robot behavior is determined by multiple different agents (agents may be human or robotic) working together to achieve a common goal [16, 53]. This paradigm is also referred to as collaborative control [55] or mixed-initiative human-robot interaction [35]. One key aspect of shared control systems is the design of *arbitration* or the division of control among agents when completing a task. Losey et al. [53] suggest that agents assume different roles during task execution. For example, the human agent controls larger robot motions while the robotic agent controls finer robot positioning. Additionally, these roles can shift over time. Thus, arbitration in shared control should allow all agents to contribute and change the type of contribution they make over time. This idea of dynamic roles is central to the arbitration mechanisms we design for our shared camera control system.

Some prior works in the robotics literature [1, 60, 72, 80] use shared control-based methods for control of a robot-mounted camera to give the remote user a view of another robotic arm used for remote manipulation. There is no local worker in such scenarios, and hence these solutions do not consider the needs of a collaboration setting. In our work, we use an optimization-based shared control method similar to Rakita et al. [72, 73] with adaptations for remote human collaboration where the robot augmented with a camera is co-located with a worker completing manual tasks.

3 THE PERISCOPE SYSTEM

In this section, we introduce the design and implementation of the *Periscope* system, which supports remote collaboration by leveraging shared camera control. Our approach provides collaborators with a low-effort means of shaping the shared visual context using a robot-mounted camera.

3.1 Design Goals

Based on prior literature and early feasibility studies, we identified five high-level design goals that guided our design process for developing a robotic camera system to support remote collaboration. The first four design goals are related to the core functionality of camera control: *versatility*, *intuitivity*, *dual-user interactivity*, and *congruity*. The final design goal, *usability*, is related to system functionality that is peripheral (but crucial) to camera control.

[1] https://support.apple.com/en-us/HT212315

[2] https://www.mylumens.com/en/Products/12/Auto-Tracking-Camera

3.1.1 ***Versatility:*** *Support camera views for various task activities.* The visual information that users need to maintain awareness and ground their conversation varies depending on task activities (e.g., searching, assembling, inspecting, or correcting). Hence, the system should support these dynamic needs and provide the helper with access to diverse sources of visual information in the workspace (including the worker's face and actions, task objects, and the environment) from various angles and in varying levels of detail. This information should be shared with the worker, so that the pair can use the shared visual context to monitor comprehension, plan future actions, achieve joint attention, and communicate efficiently.

3.1.2 ***Intuitivity:*** *Simplify camera control for users through intuitive mappings and autonomous behaviors.* Camera movement in response to user input should be clear and familiar. The usage of autonomous behaviors should facilitate the user's ability to provide high-level specifications while the robot handles the low-level details of how to achieve those specifications. Autonomous behaviors should also be used without requiring human input for aspects of robot control that may be difficult and non-intuitive for users. Camera control should be as non-intrusive as possible (i.e., not interrupt the collaboration process).

3.1.3 ***Dual-user Interactivity:*** *Allow both the helper and the worker to modify the camera view.* Both the helper and the worker require control of the camera view at different points of the collaboration process to gather or exchange information. Hence, they should be able to independently control the camera. The camera control functionality should consider the specific modalities supported by the users' locations (the helper is remote, the worker is co-located with the robot).

3.1.4 ***Congruity:*** *Arbitrate user interactions and autonomous behaviors to reach consensus.* The camera's movement can be controlled by three sources of input with potentially conflicting interests: the helper, the worker, and the autonomous robot. Hence, there is a need for arbitration of control authority between the three entities in order to determine which input has priority at what times and to prevent any conflicts. Arbitration should allow all agents (human and robotic) to contribute and change the type of contribution they make over time.

3.1.5 ***Usability:*** *Support general communication and functional requirements.* The system should support verbal communication since it is a key medium through which information is exchanged during collaboration. Additionally, the system should try to support non-verbal communication (e.g., gestures, visual annotations), especially to facilitate deictic referencing. Finally, users should be informed of the system's internal state in a non-intrusive manner as necessary.

3.2 System Overview

We developed a prototype system based on the design goals stated in §3.1. As shown in Figure 2, the *Periscope* system consists of three components: (1) *user interfaces* for the helper and the worker, (2) a set of *helper interactions*, *worker interactions*, and *autonomous robot behaviors* to support establishing a shared visual context, and (3) *system modes* that arbitrate user interactions and autonomous behaviors in real-time, resulting in camera motion. To maintain brevity, we include technical details of the implementation in Appendix **??**, and present high-level descriptions of the system below.

3.3 User Interfaces

We designed interfaces for the helper and the worker based on the goals of *versatility*, *dual-user interactivity*, and *usability*. In our remote collaboration setup, the worker is co-located with a robot arm augmented with an RGB-D (color + depth) camera, which is used to capture information about the worker and the workspace. The robot arm has six degrees of freedom, which is the minimum required for reaching any position and orientation in a 3D workspace. The helper is in a remote

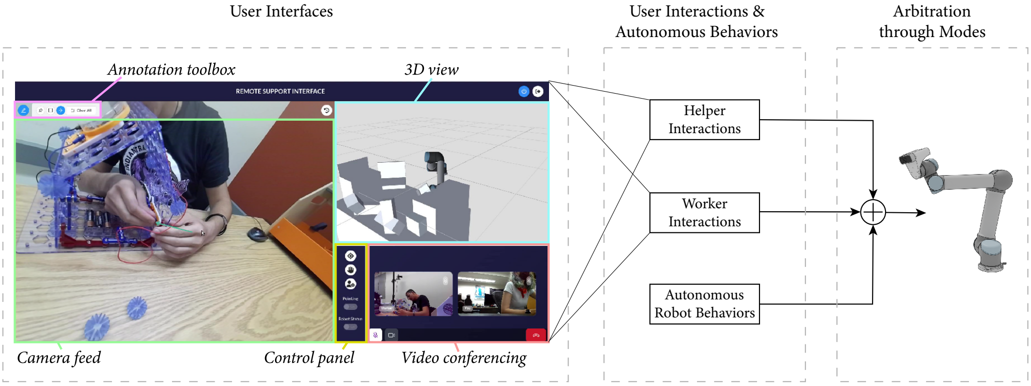

Fig. 2. The *Periscope* system consists of three components: (1) user interfaces for the helper and the worker, (2) a set of helper interactions, worker interactions and autonomous robot behaviors to support establishing a shared visual context, and (3) system modes that arbitrate user interactions and autonomous behaviors in real-time, resulting in camera motion. Each user interface consists of a *camera feed* that displays the live video feed from the robot-mounted camera, accepts mouse input commands, and can be annotated using the annotation toolbox; a *3D view* that shows a visualization of the robot and its surroundings; *video conferencing* for verbal communication between the helper and the worker; and a *control panel* for mode selection.

location and views the workspace on a 2D screen interface[3] through a live video from the RGB camera and a simulated 3D view. The worker can view the visual information shared with the helper on a 2D screen interface that is similar to the helper's interface.

The screen interface consists of four panels. The *camera feed* panel shows the live video feed from the robot-mounted camera. The camera feed accepts input commands (through mouse clicks and drags) that can be used for camera control. Additionally, the camera feed can be annotated (with a pin, a rectangle, or an arrow) using the annotation toolbox to support referential communication. Overlays on the camera feed provide visual feedback for input commands and annotations. The *3D view* panel shows a simulated visualization of the robot and its surrounding objects, and updates their states in real-time. The *video conferencing* panel allows verbal and visual communication between the helper and the worker. The *control panel* provides options related to camera control (including mode selection), in addition to those accessible through the camera feed.

3.4 User Interactions and Autonomous Behaviors

We designed interactions for the helper and the worker that are augmented by autonomous robot behaviors based on the goals of *versatility*, *intuitivity*, and *dual-user interactivity*. Below, we describe helper interactions, worker interactions, and autonomous behaviors afforded by the *Periscope* system (see Figure 3 for illustrations).

3.4.1 Helper Interactions.

Helpers use the screen interface to interact with the system through mouse input commands on the camera feed or the control panel.

Target: The helper can change the viewing direction of the camera by setting a target through a mouse right-click on the camera feed. The camera will point to the specified target such that the

[3] Although a head-mounted display is a viable option, its interplay with robotic technology for collaboration is unclear and we chose a more established display technology for this work.

target is positioned near the center of the camera's field of view. Visual feedback is displayed on the camera feed in the form of a dot corresponding to the target.

Adjust: The helper can move the camera in one direction based on a directional input in order to make adjustments to the view. Through mouse scroll, the helper can move the camera forward or backward in the direction that the camera is currently pointing at, allowing them to see more detail or context depending on task needs. Other directional inputs (mouse left-click + drag up/down/left/right) will result in different behavior depending on whether *Target* was engaged prior to *Adjust*. If the camera is pointing at a target, then it will orbitally rotate around the target point. If there is no target, the camera will linearly move in the direction specified by the helper. We will refer to the three behaviors as zoom (move forward/backward), orbit (orbital rotation), and shift (linear movement) in the remainder of the paper. Visual feedback is displayed on the camera feed in the form of arrow overlays.

The *Target* + *Adjust* interactions attempt to replicate the behavior of orbital cameras, which are widely used in virtual environments and suitable for object-focused applications.

Reset: The helper can move the camera from its current state to a pre-defined configuration by clicking a button on the GUI. The pre-defined configuration is identical to the initial configuration that the system enters at startup.

Annotate: The helper can overlay annotations on the camera feed to support referential communication between the collaborators. The helper can drop a pin to indicate a point, draw a rectangle to indicate an object, or place an arrow to indicate a direction. When the helper engages *Annotate*, the robot motion is automatically stopped to freeze the scene during the interaction.

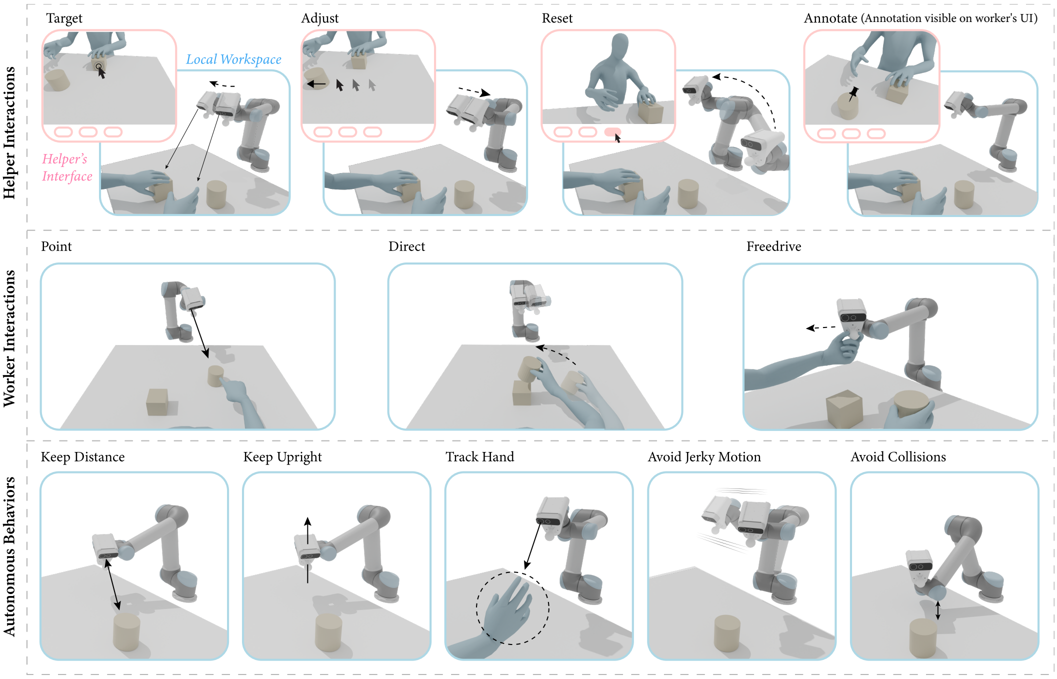

Fig. 3. The *Periscope* system supports a variety of interactions for the helper and the worker, assisted by autonomous robot behaviors.

3.4.2 Worker Interactions. Workers move the camera by engaging directly with the robot arm using physical contact and gestures recognized by the camera. These interactions leverage the worker's proximity to the robot.

Point: The worker can specify the target that the camera should look at using a pointing gesture. The camera will point to the target indicated by the worker's index finger. Additionally, the camera moves to a predetermined distance from the target (40 cm in our system) so that it is visible in adequate detail in the view. The *Point* interaction is intended to be a discrete input from the worker in contrast with the next interaction, *Direct*, which is intended to be a continuous input.

Direct: The worker can continuously influence the camera's viewing direction by moving their hand, which can be set as the camera's target. This interaction is augmented by the *Track hand* autonomous behavior, allowing the worker to influence the view without touching the robot.

Freedrive: The worker can manually move the robot-mounted camera into desired poses by manipulating the robot joints. The robot arm senses the forces applied to it and moves in the direction of the force as though being pushed or pulled by the worker.

3.4.3 Autonomous Behaviors. Autonomous robot behaviors augment helper and worker interactions by supporting the aspects of camera control that are difficult and non-intuitive for users. These behaviors are typically related to geometric (rather than semantic) qualities of the view, which are challenging for humans but feasible for robots to achieve.

Keep distance: The robot keeps the camera at a specific distance from the target point. This augments the *Adjust* interaction to enable orbital motions and *Point* interaction to keep the target visible in adequate detail. For the *Adjust* interaction, the distance is determined as the distance between the camera and the target at the time the helper engages adjustment through orbit.

Keep upright: The robot maintains the camera in an upright direction and prevents any roll (i.e., rotation along the front-to-back axis of the camera). This is typically done during assisted control of virtual cameras to avoid users from being disoriented.

Track hand: The robot detects the worker's hand (implemented using MediaPipe [54]) and automatically points the camera at the hand. This augments the worker's *Direct* interaction.

Avoid jerky motion: The robot avoids large and jittery camera motions, and promotes safe operation of the robot by maintaining the robot's range of motion within the limits of the joints. This is essential because the view needs to be stable and not disorienting for viewers.

Avoid collisions: The robot automatically avoids collisions with itself and objects in the environment, including the worker. This can be particularly beneficial for the helper, as they may face challenges in avoiding collisions when controlling the robot. Helpers have limited awareness of potential collisions as they only see the workspace from the camera's point of view and may not be aware of the placement of the robot arm's joints and obstacles outside the camera's field of view.

3.5 System Modes

We developed system modes that arbitrate the user interactions and the autonomous behaviors described in §3.4 based on the design goal of *congruity*. To achieve effective arbitration, these interactions and behaviors should work in harmony to generate camera motion. Additionally, there is a trade-off between the degree of control users desire and the amount of effort they are willing to put in. Ideally, users should have high control over the view with low effort, but this is difficult to achieve. Through an iterative design process, we developed three modes that we believe offer varying degrees of control to both users for low effort. Users can select from the three available modes via the control panel to support their current needs. The three modes are: *Helper-led Mode*, *Robot-led Mode*, and *Worker-led Mode*. Each mode is led by one of the three agents, while the other two exert less influence. This leader-follower approach makes the arbitration of control authority more tractable. After arbitration, a motion generation algorithm (detailed in Appendix A) moves the robot's joints to achieve the desired camera pose.

3.5.1 Helper-led Mode. This mode is led by the helper who can specify the camera's viewing direction by setting a target and adjusting the view through zoom and orbit. The worker has some influence over the camera's viewing direction via a pointing gesture that can be accepted by the helper. Meanwhile, the robot assists to ensure safe and high-quality camera control by keeping the camera at a constant distance during orbit, keeping the camera upright, avoiding jerky camera motions, and avoiding robot collisions. This mode gives the helper substantial control of the camera. The helper can freely move the camera to observe the workspace, and the worker can participate by pointing to a location of interest.

3.5.2 Robot-led Mode. This mode is led by the robot which tracks the worker's hand while the helper can adjust the view through zoom and orbit. Similar to the *helper-led mode*, the robot also assists by ensuring safe and high-quality camera control. This mode is designed to reduce the workload of camera control for both the helper and the worker. In this mode, the worker can focus on completing the physical task, while the robot captures the worker's activity in the workspace and maintains the worker's hand in the camera view. This mode allows the helper to focus on providing guidance without the need to control the camera to monitor the worker's behaviors.

3.5.3 Worker-led Mode. This mode is led by the worker who can set the camera's pose through freedrive (manually moving the robot) while the helper can adjust the view through zoom and shift (not orbit, since no target is set prior to adjust). This mode gives the worker substantial control of the camera. In fact, robot assistance for safe and high-quality camera control is disabled when the worker moves the camera. We wanted to include a mode in which autonomous behaviors exert less influence, giving more control to the co-located worker to handle these aspects of camera control. However, when the helper adjusts the view, the robot provides moderate assistance by avoiding jerky camera motions and robot collisions. The worker can use this mode to present visual information to the helper, and the helper can adjust the camera pose for a better viewpoint.

4 USER STUDY

We conducted an evaluation of the *Periscope* system in a lab study with 12 dyads to understand the utility of our shared camera control approach for remote collaboration.

4.1 Study Design

We recruited dyads to participate in our user study. One participant was assigned the role of *worker* and had access to the physical workspace but no instructions on how to carry out the assembly. The other participant was assigned the role of *helper* and was tasked with guiding the worker using the instruction manuals that we provided. During the study, participants collaboratively worked on a training task and a main task, which were both assembly tasks from scientific play kits. These kits were sufficiently complex to make completion without instructions challenging, and their components were sturdy enough to withstand frequent handling by participants.

The training required for participants to be able to successfully interact with the system was unclear initially. Thus, we iteratively developed a training protocol based on early participant observations and feedback. In our final training protocol, one experimenter guided both participants

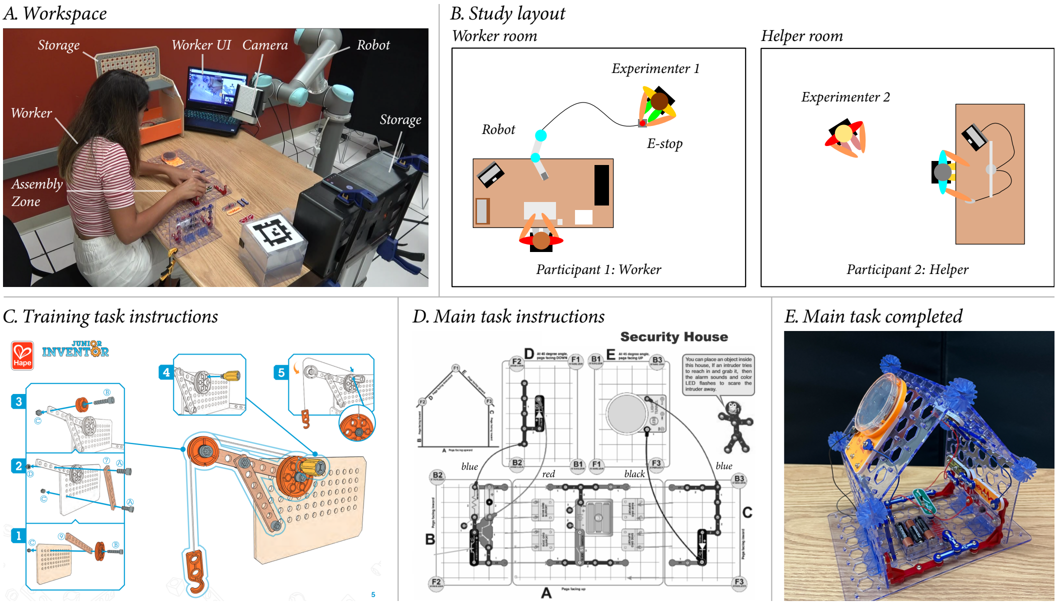

Fig. 4. A. One participant was located in the same physical space as a robot arm. B. The study took place in two rooms with accompanying experimenters. C. Instructions for the training task. D. Instructions for the main task. E. The completed structure that participant dyads were tasked with building collaboratively.

simultaneously through completion of a training task for around an hour. The training protocol consisted of ~70 steps that introduced all the functionalities available in the *Periscope* system and allowed dyads to try them out. Experimenters solicited feedback throughout the training process to encourage participants to reflect on their use of the system's functionalities. We also made adjustments to the main task protocol based on participant feedback. Below, we describe the final protocol that we developed and clarify the variations of the protocol followed by each dyad in §4.5.

4.2 Tasks

The training task was to construct a pulley system from a toy workbench kit[4] (see Figure 4C). The helper was provided with the instruction manual that came with the kit. The workbench comprised of a peg board for assembling the pulley system and a toolbox with storage space. The workbench was clamped to the table to be immobile. The components required for the task were distributed between the toolbox and another storage unit located away from the workbench.

The main task was to build a 3D illumination circuit project[5] (see Figure 4E). The helper was provided with a black and white copy of the instruction manual that came with the kit (see Figure 4D). Some visual features on the manual were deliberately blurred to ensure sufficient task complexity. Participants were tasked with building 3D circuits for a lighting and alarm system in a security house, which consisted of a base grid, two wall grids, and two roof grids. When participants began the task, the house was partially built, with one wall grid connected to the base grid and completed circuitry on the roof grids. Participants had to evaluate the partially assembled house, attach missing components to the existing wall grid, attach and build circuitry on the other wall grid and base grid, attach the roof grids, and finish the wiring.

[4] Workbench Kit: https://a.co/d/2zLeQoV
[5] SnapCircuit Kit: https://a.co/d/34trhAd

4.3 Study Setup

The study took place in two rooms: the worker room and the helper room (see Figure 4B). The participant who was assigned the role of the *worker* was located in the same physical space as a robot arm and Experimenter 1 (see Figure 4A). The worker sat behind a desk, facing the robot that was within arm's reach. The experimenter was nearby, observing the room and had access to the robot's emergency stop button. The worker viewed the screen interface on a laptop and could provide inputs to the interface using a mouse or directly interacting with the robot arm. A workbench kit (from the training task) was adjacent to the laptop. A large immobile organizer and a small movable organizer on the opposite side of the desk provided storage for various task components. The components for the training and main tasks were stored together. The participant used the laptop's camera and microphone for video-conferencing through the interface.

The participant who was assigned the role of the *helper* was located in a different room than the worker, accompanied by Experimenter 2. The helper sat behind a desk with access to a laptop, a monitor, and a mouse for interacting with the interface. The participant used the laptop's camera and microphone for video-conferencing through the interface.

4.4 Procedure

This protocol was approved by a university's Institutional Review Board (IRB). We conducted the study in two separate rooms in a university laboratory. Each session lasted approximately two hours. The first author (Experimenter 1) facilitated the study along with another experimenter (Experimenter 2). Both experimenters individually described the study to the participant and obtained written consent. Experimenter 1 introduced the interface and the physical robot to the worker before connecting to the video conference. In parallel, Experimenter 2 provided the same introduction for the interface and described the virtual robot in the 3D view of the interface to the helper before joining the video conference. Experimenter 1 guided both participants simultaneously through the training protocol. The experimenter familiarized participants with the workspace, outlined the task flow, and initiated test interactions in each mode. Participants were then asked to use their cheat-sheet, which listed all system features, to summarize what they had learned.

During the training task, the helper was encouraged to locate the necessary component, ask the worker to pick it up, and provide assembly instructions to the worker. Participants were asked to gather the required components for each step (steps are listed in the manual shown in Figure 4C) using a certain mode, and then assemble the components using an alternate mode. They were then asked to reflect on their experiences. We repeated this procedure for all the modes, allowing participants to gain experience with each mode for different task activities. We allowed participants to complete the final step of the task using any combination of modes they preferred. Participants were finally asked to reflect on their overall experience in all modes. If a participant avoided using a feature or used it wrongly, the experimenter reminded or corrected them regarding the system's functions. The training task took approximately 60 minutes.

The video conferencing link was disabled before Experimenter 1 went to the helper room and explained the procedure and goals for the main task to the helper. The helper was shown a completed model of the security house and had the opportunity to interact with it. Then, Experimenter 1 partially disassembled the house and set it up on the worker's table. The video conference was then resumed, and participants were given high-level directions on which panel to assemble. Participants were given the flexibility to use any (or none) of the system's modes and other features they found suitable for completing the task. We used this approach because we wanted to gain insights into how people utilized the system in a relatively realistic setting. Participants had 45–60 minutes to collaboratively work on the main task.

4.5 Participants

For the user study, we did not target any particular user group, as the scientific play kits did not require specific expertise and the system was designed for use by individuals unfamiliar with robots. We recruited 24 participants from a university campus. Demographic information for one dyad was not collected. The remaining participants (8 female, 14 male) were aged between 18 and 69 years (M = 26.32, SD = 10.48). Participants had various educational backgrounds, including urban design, business, physics, engineering, and computer science. Two participants reported prior participation in robotics studies, and one dyad consisted of individuals who knew each other prior to the experiment.

The first four out of the twelve dyads underwent a less rigorous training protocol and performed a different (but similar) task from the kit. While these four dyads were important for establishing the final protocol, we excluded them from our dataset for analysis as they followed a different procedure compared to the other eight dyads. The next two dyads followed the procedure described in §4.4, with the only difference being that the helpers were not shown the completed model of the security house before starting the main task. The remaining six dyads strictly followed the procedure described in §4.4. To ensure consistency and comparability within the dataset for the analysis described below, we used data from the last eight dyads that followed a similar procedure.

4.6 Analysis

During the study, we screen-recorded the helper's interface and recorded the workspace (including the worker and the robot), resulting in ~36 hours of video recordings (12 dyads*2 users*~1.5 hours). The dataset for our analysis consists of ~12 hours of video recordings from eight dyads during the main task (8 dyads*2 users*~0.75 hours). This is rich multi-user, multi-modal data containing dialogue, interactions with the system, worker actions, and camera motions.

We analyzed the videos using a deductive thematic analysis approach [6]. The first author and a study team member were familiarized with the data through conducting all study sessions and transcribing participant conversations. Both the first author and the study team member coded all helper videos (screen-recordings) to identify relevant conversations and patterns, conducted meetings to discuss their codes and resolve any conflict, and distilled the codes in a codebook. The first author then coded all worker videos and refined the codes. Resulting themes were refined by the first author and reported after discussions with the remaining authors. ELAN[6] [14] was used for video coding, and the collaborative whiteboard app Miro[7] was used to refine thematic findings.

5 RESULTS

Overall, we observed that dyads frequently utilized the *Periscope* system's modes and other features to establish a shared visual context that enhanced their verbal communication. All our results are summarized in Tables 1 and 2. Table 1 provides a list of use patterns for modes and other features of the system. These use patterns are nuanced interpretations of the rich multi-modal data that we analyzed. Thus, we include examples in Appendix B and provide references to them in this section to provide the context of the rich interactions from which they were interpreted. Table 2 provides an overview of the frequency and duration of use of the modes and other features. This table also includes a ranking based on the degree to which each dyad succeeded in completing the main task. We did not expect all dyads to reach completion because we deliberately designed the task to be challenging to prevent dyads from succeeding purely through verbal communication. We did not

[6] https://archive.mpi.nl/tla/elan
Max Planck Institute for Psycholinguistics, The Language Archive, Nijmegen, The Netherlands
[7] https://miro.com/

Table 1. Summary of use patterns identified from the analysis of video recordings of eight dyads who participated in a user study. Column 2 provides references in §5 to details about each pattern.

Feature	#	**Use Pattern**
Helper-led mode	5.1-1	The helper gains awareness of the workspace.
	5.1-2	The helper provides the worker with task instructions.
	5.1-3	The helper searches for something.
	5.1-4	The helper attempts to move the camera before asking the worker do it instead.
Robot-led mode	5.2-1	The dyad gathers components for the build.
	5.2-2	The helper tracks the worker's movement.
Worker-led mode	5.3-1	The worker wants to share some information with the helper.
	5.3-2	The worker anticipates the helper's need for a different view.
	5.3-3	The worker offers to move the camera on behalf of the helper.
	5.3-4	The helper attempts and fails to move the camera on their own.
	5.3-5	The helper is already aware from an earlier attempt that a particular view is difficult to achieve.
	5.3-6	The helper requests repositioning the camera that the worker had previously set up.
	5.3-7	The helper does not know where to position the camera.
Point	5.4.1-1	The helper asks the worker for a specific view.
	5.4.1-2	The worker refers to something in the workspace.
Reset	5.4.2-1	The reset pose serves as a bookmarked pose that provides a sufficient view of the workspace with minimal effort.
	5.4.2-2	The reset pose serves as an intermediate pose when transitioning from one sub-task to the next.
	5.4.2-3	The reset pose is a comfortable starting configuration for the *helper-led mode.*
	5.4.2-4	The system does not respond as expected.
Annotate	5.4.3	The helper refers to something in the workspace.

Table 2. Frequency and duration of use of *Periscope's* modes and other features. Dyads are ranked based on how much of the task they completed (#1 being best).

Dyad | Rank	***Helper-led Mode***	***Robot-led Mode***	***Worker-led Mode***	***Point***	***Reset***	***Annotate***
D1 | #8	9, 19_M, 25_{SD}	3, 70_M, 20_{SD}	7, 36_M, 23_{SD}	0	5	52
D2 | #1	8, 65_M, 76_{SD}	2, 13_M, 4_{SD}	1, 18_M, NA_{SD}	0	7	23
D3 | #3	7, 20_M, 13_{SD}	6, 126_M, 122_{SD}	8, 24_M, 7_{SD}	6	9	24
D4 | #5	15, 46_M, 34_{SD}	3, 18_M, 17_{SD}	4, 25_M, 14_{SD}	2	12	29
D5 | #7	2, 6_M, 7_{SD}	0, NA_M, NA_{SD}	15, 13_M, 16_{SD}	1	0	14
D6 | #4	21, 44_M, 39_{SD}	2, 18_M, 20_{SD}	5, 22_M, 15_{SD}	1	12	38
D7 | #2	12, 42_M, 48_{SD}	1, 20_M, NA_{SD}	8, 25_M, 12_{SD}	1	7	36
D8 | #6	8, 17_M, 8_{SD}	3, 106_M, 87_{SD}	10, 27_M, 19_{SD}	0	5	41
Total	82, 38_M, 41_{SD}	20, 71_M, 85_{SD}	58, 23_M, 16_{SD}	11	57	257

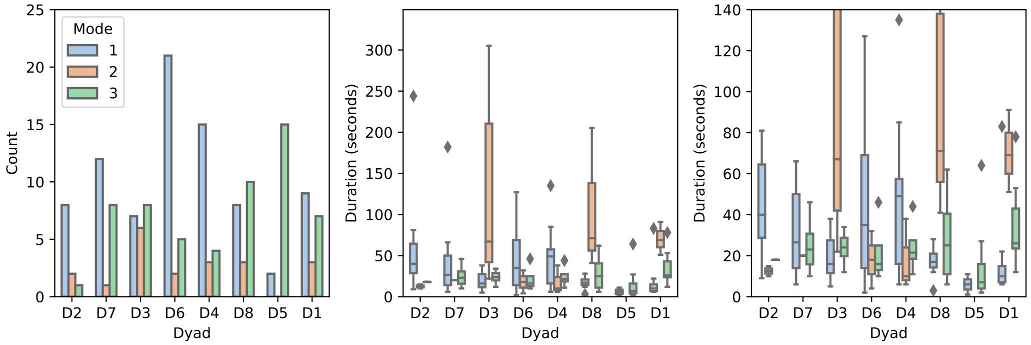

Fig. 5. Visualization of the frequency and duration of use of *Periscope's* modes. Dyads are arranged in descending order based on most to least completion of the task. (Left) Count plot depicting the frequency of use of the three modes in the data. (Center) Box plot depicting the duration of use of the three modes in the data. (Right) Zoomed-in view of the box plot depicting the duration of use of the three modes in the data with y-scale from 0 to 140.

expect all dyads to reach completion because we deliberately designed the task to be challenging to prevent dyads from succeeding purely through verbal communication. A visual representation of the frequency and duration data for the modes from Table 2 can be found in Figure 5. In the rest of the section, we provide a detailed breakdown of use patterns. In §6.1, we elaborate on the significance of these results to our design goals.

Note: Dyads are referred to as *D1, D2, D3, D4, D5, D6, D7, D8*. Dyad references and frequency of use patterns are included in parentheses.

5.1 Helper-led Mode Use Patterns

We observed that this mode was used 82 times (excluding the use of this mode when the worker used the pointing gesture which we discuss separately in §5.4.1). The average duration of each use was 38 seconds (*SD* = 41 seconds). We observed that helpers used the mode in the following ways: targeting only *(21/82)*, targeting with zoom adjustment *(27/82)*, and targeting with orbit adjustment *(34/82)*. This mode was the first mode that majority of the helpers used during the session *(6/8)*. The remaining dyads, *D3* and *D5*, used the *worker-led mode* as their first mode. This mode was exited when helpers opened the annotation toolbox *(46/82)*, reset the camera *(16/82)*, switched to the *worker-led mode (13/82)*, or switched to the *robot-led mode (5/82)*. This mode was not exited in the remaining cases *(2/82)*. Instead, it was either immediately followed by another use of the same mode *(1/2)* or the session ended *(1/2)*.

We observed four distinct use patterns, with occasional overlaps *(13/82)*:

(1) *The helper gains awareness of the workspace (36/82):* The helper inspected various objects in the workspace in order to assess the situation. For example in *D4*, as the worker attached a component onto a grid, the helper wanted to *"double check that it [the component] is facing the correct way"* and moved the camera for a better view of the grid (EB.1). This category is distinct because it involves the helper gaining information from the remote workspace.

(2) *The helper provides the worker with task instructions (30/82):* The helper provided guidance to the worker to make progress on the task. For example in *D6*, the helper moved the camera to look at the components that the worker had recently attached and instructed the worker to

make modifications, *"This part right here [a base support]...Okay, so you have to flip it"* (EB.2). This category is distinct because it involves the helper providing information to the worker.

(3) *The helper searches for something (17/82):* When the helper and the worker searched for something together, they typically utilized the *robot-led mode* (§5.2-1), but if the helper needed to search for something independently, they used the *helper-led mode*. For example in *D1*, the helper explicitly switched from *robot-led mode* to *helper-led mode* while looking for a component, which may have been prompted by the worker not following their instructions correctly (EB.5). We consider *searching* to be a distinct category in which the helper mostly used targeting only or targeting with zoom adjustment *(16/17)*.

(4) *The helper attempts to move the camera before asking the worker do it instead (13/82):* If the initial attempt with the helper-led mode was not sufficient to get the desired view, the helper asked the worker to move the camera using the worker-led mode (see EB.6). We will revisit this reason in §5.3-4 when discussing the use of the *worker-led mode*.

5.2 Robot-led Mode Use Patterns

This mode was used 20 times (excluding the use of this mode when the worker used the pointing gesture which we discuss separately in §5.4.1). The average duration of each use was 71 seconds (SD = 85 seconds). We observed some view adjustment by the helper *(Adjust: zoom (4/20), Adjust: orbit (2/20))*. There were three instances of the robot completely losing track of the worker's hand requiring the helper to reset the robot *(2/3)* or engage the *helper-led mode (1/3)*. This mode was exited when helpers opened the annotation toolbox *(8/20)*, switched to the *helper-led mode (8/20)*, reset the camera *(3/20)*, or switched to the *worker-led mode (1/20)*.

We observed two distinct use patterns, with one overlap *(1/20)*:

(1) *The dyad gathers components for the build (17/20):* The *robot-led mode* was mostly employed to locate the required components in the organizer (see EB.7). In the majority of these cases, the helper explicitly informed the worker that the tracking mode was on and that their hand was being tracked *(16/17)*. In one instance *(D4)*, the worker held their hand visible to the camera as if to direct the robot, prompting the helper to switch from the *helper-led mode* to the *robot-led mode*. In all cases, we observed that workers explicitly directed the camera by moving their hand to relevant locations *(17/17)*. Additionally, when waiting for the helper to give further instructions, workers often rested their hand on the table to maintain a steady view of the relevant area *(13/17)*. The *robot-led mode* was most frequently used by dyads *D1 (3/17)*, *D3 (6/17)*, and *D8 (3/17)* to find components.

(2) *The helper tracks the worker's movement (4/20):* The helper used the *robot-led mode* to maintain the worker's hand in view as the worker moved their hand to demonstrate or put something together (see EB.8). In these instances, the worker did not explicitly direct the camera.

5.3 Worker-led Mode Use Patterns

We observed that this mode was used 58 times in two distinct ways: *worker-initiated (22/58)* or *helper-initiated (36/58)*. This split-use may be due to the design of this mode, which can be engaged either by the worker or the helper. We distinguish between the mode's initiation and engagement; initiation relates to the individual who suggests using the mode, while engagement refers to actually clicking the button. The average duration of each use was 23 seconds (SD = 16 seconds). We observed some view adjustment by the helper *(6/58)*. Helpers often switched to the *helper-led mode* after attempting to adjust the view in this mode *(4/6)*. One instance of view adjustment *(D5)* required the system to be reset since the helper and the worker both attempted to move the camera at the same time, activating the robot's emergency brake. This mode was exited when helpers opened the annotation toolbox *(29/58)*, switched to the *helper-led mode (10/58)*, reset the camera *(6/58)*, or switched to the *robot-led mode (2/58)*. This mode was not exited in the remaining cases *(11/58)*. Instead, it was either immediately followed by another use of the same mode *(7/11)*, or the session ended *(4/11)*.

5.3.1 Worker-initiated: Workers initiated this mode either directly by engaging the mode and moving the camera *(12/22)* or indirectly through conversation *(10/22)*, such as *"Do you need me to move the camera again" (D4)*. The former behavior, in which the worker altered the view without notifying the helper, was most prevalent in dyads *D5* and *D8 (11/12)*. Workers initiating this mode mostly engaged the mode themselves *(16/22)* or the mode was already active from prior use *(3/22)*. Otherwise, they asked the helper to engage the mode on their behalf *(3/22)* with a phrase, such as *"Do you want to move to mode 3 [worker-led mode] and I can show it?" (D1)*.

We observed three main reasons for workers initiating the *worker-led mode*:

(1) *The worker wants to share some information with the helper (7/22):* The worker showed the helper something new in the workspace *(2/7)*, a view pertinent to a query or response that they had regarding the task *(4/7)* (see EB.9), or their progress on the task *(1/7)*. Workers may *(3/7)* or may not *(4/7)* let the helper know that they are changing the view.
(2) *The worker anticipates the helper's need for a different view (12/22):* When a helper acknowledged the end of the current step in the process or verbalized the next step in the process (see EB.10), some workers anticipated the helper's need for a different view and offered to move the camera *(4/12)* or proactively moved the camera without informing the helper *(8/12)*.
(3) *The worker offers to move the camera on behalf of the helper (3/22):* When a helper expressed frustration with camera positioning, for instance, by stating, *"Um...let me see if I can move the camera just a little bit" (D3)*, some workers offered to move the camera on the helper's behalf.

We observed the least amount of initiation of this mode by the worker in dyads *D2 (none)*, *D6 (once)*, and *D7 (none)*. Additionally, there were six instances of conflict in these dyads—*D2 (1/6)*, *D6 (2/6)*, *D7 (3/6)*—when the worker offered to move the camera or tried to proactively move the camera for any of the reasons mentioned above, but was overruled by the helper who used the *helper-led mode* to move the camera (see EB.11).

5.3.2 Helper-initiated: Helpers initiated this mode with a verbal request to the worker to move the camera. The request was ambiguous and context-specific, yet the worker typically understood it correctly *(31/36)*. For example, one helper requested, *"Could you move the camera so that I'm getting like a more of a bird's eye view" (D3)*. While the helper did not indicate which area or item should be visible, the worker showed the helper a view of the base grid based on an earlier conversation about where the base supports would link to on the base grid. If the worker was unable to decide which view to show, there was additional conversation to clarify the request *(5/36)*. When the worker moved the camera, the helper often acknowledged an adequate view *(20/36)* with a phrase such as *"Okay alright, that's enough" (D6)*.

We observed four main reasons for helpers initiating the *worker-led mode*:

(4) *The helper attempts and fails to move the camera on their own (13/36):* When the helper could not get the desired view using the *helper-led mode*, they asked the worker to move the camera. For example in *D7*, the helper made an unsuccessful attempt to inspect the roof panel and gave up, saying, *"I am not able to see the top panel...can you?...I need to look up to the panel"*.
(5) *The helper is already aware from an earlier attempt that a particular view is difficult to achieve (6/36):* The helper preemptively requested the worker to move the camera (see EB.13) because

they had previously made an effort to observe the same area but had either been successful after a protracted attempt *(2/6)* or had been unsuccessful and had relied on the worker *(4/6).*

(6) *The helper requests repositioning the camera that the worker had previously set up (14/36):* After the *worker-led mode* was used once, there were instances when helpers requested the view to be modified to show something that had become more pertinent (see EB.14).

(7) *The helper does not know where to position the camera (3/36):* Since the helper was remote, the worker was more familiar with the layout of the workspace. Thus, the first use of the *worker-led mode* by three helpers *(D3, D5, D6)* was for the worker to move the camera so they could look at something that was located in a place they were unfamiliar with (see EB.15).

Sometimes, helpers initiating the *worker-led mode* engaged the mode *(10/36)* or the mode was already active from prior use *(4/36)*. Otherwise, workers engaged the mode *(18/36)*. We observed four instances of conflict over mode engagement when both the helper and the worker engaged the mode, thereby canceling out each other's inputs. Additionally, there were two instances of conflict in dyad *D8* when the helper said, *"Can you show me..."*, and used the *helper-led mode* to move the camera. This statement was misunderstood by the worker as a request to engage the *worker-led mode* and move the camera, resulting in overriding the helper's mode selection.

5.4 Other Use Patterns

5.4.1 Point: We observed 11 instances where pointing was used for two reasons:

(1) *The helper asks the worker for a specific view (6/11):* Pointing was used explicitly by helpers in dyads *D3 (5/6)* and *D7 (1/6)* to request a view. For example, the helper in *D3* asked the worker, *"Could you point to the wall so that I can see inside it?"* (see EB.16).

(2) *The worker refers to something in the workspace (5/11):* Four workers—*D3 (1/5), D4 (2/5), D5 (1/5), D6 (1/5)*—used pointing to refer to something in the workspace (see EB.17).

Only one dyad *(D3)* successfully completed the interaction sequence as designed *(3/11)*: worker points, helper approves, and camera provides a close-up of the worker's target. In multiple cases, the helper was unable to approve the worker's target because of a bug in the system *(5/11)*. In these cases, the helper switched to the *robot-led mode* to track the worker's hand *(3/11)*, switched to the *worker-led mode (1/11)*, or did not take any action *(1/11)*. In the remaining cases, the helper never attempted to use the *helper-led mode* but directly used the *robot-led mode* when the worker pointed toward something *(3/11)*. Interestingly, some helpers *(D3, D7)* expected the camera to align with the direction of pointing. For example, the helper in *D3* stated, *"Oh, it's looking at your hand and not what I want it to be looking at,"* when the view did not match their expectation of the camera aligning with the direction of pointing (see EB.18).

5.4.2 Reset: We observed 57 instances where the helper used the *Reset* feature and identified four potential reasons for its use. However, due to insufficient context in the data to determine the intent behind each occurrence, we do not report the number of instances for each reason.

(1) *The reset pose serves as a bookmarked pose that provides a sufficient view of the workspace with minimal effort:* By simply clicking a button, the helper could easily obtain a reasonable view of most of the workspace (see EB.19).

(2) *The reset pose serves as an intermediate pose when transitioning from one sub-task to the next:* In many instances, the completion of a sub-task was marked by the helper using the *Reset* feature (see EB.20 and EB.21).

(3) *The reset pose is a comfortable starting configuration for the helper-led mode:* The robot would occasionally get into an odd configuration that the helper found challenging to modify. In

such cases, the helper relied on the reset feature to restore the robot to its initial configuration, with which they were familiar and comfortable working (see EB.22).

(4) *The system does not respond as expected:* Occasionally, there was a prohibitive lag between user commands and the corresponding robot motion, or the user was unable to move the camera because of issues with the robot's autonomous behaviors, such as being stuck in a collision state or losing track of the worker's hand (see EB.23). In response, helpers used the *Reset* feature as a way to restore the system to a functional state.

5.4.3 Annotate: We observed 257 instances where the helper added annotations to the view. These visual annotations were accompanied with one or more of the following words in the helper's speech: *this, that, these, those, it, other, here, there, where, looks similar/like, same, thing, next, last, one, another, both, right, way, direction, across, on, top, middle, bottom, horizontal, opposite.* While we see evidence of the system facilitating referential communication, a comprehensive conversation analysis on this topic is outside the scope of this work.

6 DISCUSSION

In this section, we discuss our system's effectiveness in supporting our design goals introduced in §3.1, present design implications for future robotic camera systems, address the limitations of our current work, and suggest possible directions for future research.

6.1 Reflection on Design Goals

Note: Whenever a statement is connected to a result in §5 or is illustrated by an example in Appendix B, the relevant reference is included in parentheses.

6.1.1 Versatility: The frequent use of the system's features (Table 2) and consistent use patterns across dyads (Table 1) is encouraging[8], especially since participants were not compelled to use any features to move the camera. The initial configuration (which is also the pre-defined pose for the *Reset* feature) offered a reasonable view of the workspace, and if the worker had brought everything into the static view or the dyads had relied mainly on verbal communication, it may have been possible to progress on the task (albeit inefficiently). However, we found that participants made use of the system's versatility to obtain diverse and context-specific views to support a variety of task activities, such as gaining awareness, providing instructions, searching and gathering components, assembling, sharing information, inspecting objects, and correcting errors.

Similar to prior work [32, 45], we saw evidence for the helper and the worker using our interface to establish a shared visual context in order to maintain awareness and ground their conversation. It should be noted that the following discussion about system *versatility* is inherently linked to system *usability*, which enabled effective communication between users. Annotation, in conjunction with the use of deixis (e.g., *this, here, across, now, next*; see §5.4.3), was the most apparent use of the shared context to achieve efficient and unambiguous communication. Additionally, dyads used the shared context to ground references of task objects (e.g., *"L-shaped stuff"* for the base support in EB.4 and *"the blue one"* for the snap-connector in EB.5), especially since they had no prior shared vocabulary for the objects. Finally, infrequent verbal communication related to some aspects of collaboration, such as monitoring comprehension, may suggest an effective use of visual information. Helpers could infer worker comprehension by watching worker actions immediately after receiving instructions, and then correct them if necessary (e.g., EB.1).

[8] The duration of mode use in Table 2 is more challenging to interpret than the frequency data, as it is possible that the user found obtaining the desired view difficult and therefore took longer, or that the user was actively accumulating information throughout the entire time. Better interpretations of duration and frequency data would be possible if we knew the quality of the information users acquired from every view.

We found some limitations in the system's versatility due to the particular robotic hardware that we used. There were some angles and locations that the robot could not be configured to show. Additionally, the system did not adequately support certain task activities such as debugging that required precise views and repeated view specifications (discussed in detail in §6.1.2). These findings provide concrete directions for enhancing the versatility of future systems.

6.1.2 Intuitivity: The frequent use of the system's features to move the camera (Table 2) may indicate that there were enough instances where users found it worthwhile to put in the effort to acquire information through camera control. Moreover, participants converged on particular patterns in their use of camera controls (Table 1), which could suggest that the controls had some degree of intuitivity. It is also promising that autonomous robot behaviors were generally invisible to participants. Occasionally, the robot lost track of the worker's hand and required guidance from the helper (§5.2) and rare robot collisions required experimenters to restart the system (§5.4.2-4). Otherwise, users did not have to intervene and take responsibility for the aspects of camera control that were handled by the robot. Overall, we believe that the discussion in §6.1.1 of participants using the system to achieve diverse, informative, and task-relevant views is supportive of the intuitiveness of our camera controls.

Conversation pauses and dialogue about camera control in our data raise concerns that participant efforts to move the camera interrupted their flow of collaboration (e.g., EB.1, EB.4, EB.9, EB.14, and EB.22). Nevertheless, helpers and workers took the time to do so in order to get a good view, after which interactions were smooth. This is illustrated in EB.3, where the verbose description, *"(it should attach on)...the inside of the triangle, like on the inside edge of the triangle that connects to the circle thing...Sorry...the thing...the clear thing with the circle on it"*, was replaced by the concise deictic expression, *"It should attach right...here"*, after the helper took the effort to obtain a good view. While we have taken steps in the right direction with our system design, we explain cases below where our system did not adequately meet this design goal.

Obtaining precise views: Helpers seemed comfortable with camera control when they used targeting only or targeting with zoom (e.g., during searching; see §5.1-3) to set three or four of the camera's DoF. In contrast, helpers had trouble with camera control when trying to obtain views that needed precise 6-DoF camera specification, such as viewing the bottom of the roof grid (EB.12).

Repeated view specifications: Helpers were frustrated with repeatedly specifying views when they had to move away to look for and collect components before returning to finish assembly (e.g., EB.13). Here, the reset pose was useful on occasion since it may be used as a transitional pose when switching between sub-tasks (§5.4.2-2), or as a quick way to get a sufficient view of the workspace without much effort (§5.4.2-1).

Lack of autonomous behaviors in Freedrive: We did not include any autonomous robot behaviors in our implementation of the *Freedrive* interaction for the worker. However, this may have resulted in workers having too many degrees of freedom to manipulate, causing them to sometimes struggle with physically posing the robot's joints. Workers had the most trouble with keeping the camera upright and the robot colliding with itself.

Non-intuitive pointing behavior: Some participants expected the camera to align with the direction of pointing and expressed frustration when this was not the case (e.g., EB.16 and EB.18).

6.1.3 Dual-user Interactivity: We begin with a discussion of how helpers and workers individually used their interactions. Helpers could have simply requested the worker to move the camera each time (as the helper in dyad *D5* did), but most helpers extensively used the interactions provided to them and independently explored the workspace without relying on the worker (§5.1). Additionally, this independence allowed parallel work in which the helper could move the camera as the worker was simultaneously carrying out a task (e.g., EB.3). The helper could also intervene based on their assessment of the state of the task without always needing to engage the worker in a dialogue about the status (e.g., EB.2). Workers used the interactions provided to them in two distinct ways. In the intended use, workers leveraged their familiarity and access to the workspace in order to share information with the helper (§5.3.1-1 and §5.3.2-7). However, more frequently, workers moved the camera on behalf of the helper when they were dissatisfied with their user experience (discussed later in this subsection). Overall, when participants had ownership of a part of the task or relevant information, they took ownership of the point of view. This finding is consistent with prior work [49, 59], but it merits further study to determine if there is a relation (and what its nature is) between the extent to which a user feels task or information ownership and the degree of camera control (e.g., 1-DoF vs 6-DoF) provided by an interaction.

An intriguing and novel outcome of participants having different degrees of camera control in each mode was the frequent transfer of control of the view between the helper and the worker both within and between modes (see mode exit details in §5.1, §5.2, and §5.3). Our analysis revealed that we must consider a user's influence over the view not only through the explicit use of a system feature but also through conversation, such as in the helper-initiated *worker-led mode* (§5.3.2). Influencing the view through conversation was unexpectedly frequent during the use of the *robot-led mode* for gathering components, in which the helper verbally directed the worker to move their hand to modify the view (§5.2-1). The worker was also mindful of this collaborative view control and exhibited unique behaviors, such as resting their hand on the table to maintain a steady view of the relevant area for the helper. In this scenario, the view is continuously, and sometimes implicitly, negotiated between the helper and the worker. Collaborative view control was also present, but infrequently and intermittently, within the *helper-led mode* and the *worker-led mode*. In the *helper-led mode*, workers could use pointing (although only dyad *D3* successfully used this feature; see §5.4.1-2) and in the *worker-led mode*, helpers could adjust the view themselves or ask the worker to adjust it instead (§5.3.2-6). The balance of view control in the *helper-led mode* and the *worker-led mode* may have been skewed disproportionately in favor of either the helper or the worker, making it less apparent than in the *robot-led mode* that view control could be shared.

There is an explicit transfer of view control when switching from one mode to another. Users may have changed modes due to the evolving needs of the task that necessitate more or less camera control (e.g., EB.5). Otherwise, users may exit a mode (in favor of another) when they were unable to acquire the desired view using the interactions provided in that mode. This was more typical with helpers requesting workers to move the camera on their behalf (§5.3.2-4, §5.3.2-5, §5.3.2-6), although there were also cases of the reverse (e.g., EB.22 and EB.23). Although this demonstrates the potential of dual-user interactivity to compensate for shortcomings in the system, future designs of the system should minimize this behavior.

6.1.4 Congruity: The frequent transfer of view control between the helper and the worker within and between modes, which we discuss in connection to dual-user interactivity in §6.1.3, is made possible through effective arbitration. We designed arbitration mechanisms within the system to ensure congruity, but interestingly, we observed that verbal negotiation between the helper and the worker during collaborative view control (discussed in §6.1.3) also helped to achieve congruity. Another facet of arbitration is the role of autonomous robot behaviors in camera control. Autonomous behaviors were generally unobtrusive to participants, as discussed in connection to intuitivity in §6.1.2, and thus contributed to effective arbitration.

The leader-follower approach (see §3.5) that we adopted to streamline arbitration seemed to be an effective strategy, as it may have helped to establish clear roles and ownership. This approach is also linked to the concept of information ownership leading to view ownership, as discussed in §6.1.3, where the leader drives the task forward based on information they possess, and the follower follows suit. However, we observed a few instances of conflict in the data, highlighting areas where arbitration could be more effective. There were disagreements between the helper and the worker on when to engage the *worker-led mode* and by whom (§5.3). This is due to both users having the option of engaging the mode. Another source of conflict in this mode was when the helper and the worker both tried to move the camera. Finally, there were issues with the arbitration of the worker's pointing interaction, which required approval by the helper to influence the view and hence diminished the worker's authority (§5.4.1). While it is promising that there were only a few instances of conflict, we recognize that we may have granted the helper excessive authority during arbitration. The worker had a diminished role in the arbitration process. This made achieving consensus more manageable, but it did not fully leverage the potential contributions that workers could make. Additionally, the robot could also play a more active role and take initiative, rather than just performing passive behaviors in support of helper and worker interactions.

6.1.5 Usability: The system facilitates rich interactions between the helper and the worker (illustrated through examples in Appendix B) and enables dyads to remotely collaborate on physical tasks. This is promising for the usability for the system. Below, we address usability issues that provide potential for improvement in future systems.

Latency and unresponsiveness: All helpers expressed frustration with the delay between their commands and corresponding robot motion. Furthermore, this latency varied during the session. This was especially problematic when the robot did not immediately respond to commands for adjusting the view (orbit, shift, zoom). Helpers then gave additional commands which caused the robot to overshoot the target location and necessitated correction.

Input sensitivity and direction: We had defined a standard amount and direction of robot movement in response to mouse input, but helpers may have different preferences based on their past experience with other systems and the task context.

Lack of transparency in certain state transitions: When the worker moved the camera in the *worker-led mode*, robot assistance through autonomous behaviors was designed to be inactive. However, this meant that the robot might be in a collision state and unable to move for safety reasons when helpers switched to the *helper-led mode* and attempted to move the camera. Since this information was not communicated to users, they assumed that the system was unresponsive and reset the robot's pose to resolve the issue.

Split-attention effect for the worker: The worker's interactions with the system were spatially distributed. Workers engaged *Mode 3* using the interface on the laptop and then moved the robot, which could be in a different part of the workspace than the laptop. While moving the robot, the worker had to simultaneously look at three spatially distributed areas: the task space, the robot (to avoid collisions), and the shared view on the laptop. The split-attention effect seemed less of a factor (although not eliminated) when the helper modified the shared view. The position of the robot-mounted camera changes whenever helpers modified the view, providing embodied cues about the helper's focus of attention to the worker. This could help the worker in achieving joint attention without requiring them to look at the interface on the laptop.

6.2 Design Implications

6.2.1 Modeless arbitration: Designing arbitration mechanisms that directly leverage the helper and worker interactions (e.g., target, point, freedrive), without the need for explicit modes, could improve the *intuitivity* and *congruity* of the system. For example, in the current prototype, setting the camera's target as an object versus the worker's hand requires disengaging from one mode and engaging in another. With an integrated interaction system, multiple specifications could be initiated using the same input, such as clicking on the hand in the camera feed to initiate hand tracking, and clicking on an object in the feed to set it as the camera's target.

6.2.2 Stronger worker-centered design: Designing the system with explicit support for workers could improve *dual-user interactivity*, particularly because our system, like many other prior works, was designed in a helper-centered manner. For views that are challenging for helpers to specify remotely, incorporating complementary interactions for workers could empower them to more efficiently shape the desired view on behalf of helpers. Additionally, in our current design, helpers have significant authority (e.g., to switch between modes). Designing the system to encourage variable authority between the helper and the worker could enhance the fluidity of collaboration. For instance, the system could automatically switch to freedrive when the worker makes physical contact with the robot, and switch to remote control when the helper provides mouse input.

6.2.3 Use pattern-based arbitration: Designing arbitration based on the use patterns presented in Table 1 has the potential to improve the *versatility* of the system. For example, in different contexts, users might require different sensitivity to their directional input when trying to adjust the view. The robot could adjust the amount of movement based on the perceived use pattern (inferred from the state of the environment and usage history). This approach could provide users with the responsiveness needed in one use case versus the precision required in another.

6.2.4 Expertise-based arbitration: Designing arbitration around expertise levels could improve the *intuitivity* and *congruity* of the system. For example, novices may benefit from simplified camera control and a more active robot agent. As users gain expertise, the system could provide them with increased control through new interactions or new ways to parameterize interactions.

6.2.5 System feedback: Providing more frequent and timely feedback to users (e.g., during state transitions) could enhance the *usability* of the system and promote efficient collaboration by reducing the need for dyads to discuss system status. The worker may also benefit from more embodied cues that inform about the state of the system.

6.3 Limitations

Our work has a number of limitations that primarily stem from the design of our evaluation study. Firstly, although we have envisioned *Periscope* to serve as an expert tool, our evaluation was conducted with novices. We attempted to overcome this discrepancy with extensive training until participants appeared fluent with the system. However, experts who frequently utilize video-based collaboration tools for physical tasks might provide more insight into the challenges they face day-to-day, use our system differently, and provide different feedback. Future studies could focus on expert users, explore other real-world scenarios that they might face, such as expert helpers assisting multiple workers, and apply these tools in more realistic tasks and conditions. Secondly, the setup of our study resulted in a stationary work environment where the robot arm only utilized about a third of its range of movement. Although these constraints afforded greater safety for the participant from collisions with the robot and minimized the potential for discomfort from large motions within close proximity, more research is needed to understand how our system might be used by collaborators in other workspace arrangements. Finally, as we allowed participants to freely interact with the system, determining the specific interaction capabilities of the system that contribute to user success is challenging. Future research that includes comparisons within the system (e.g., evaluating performance when only using one mode for the entire task) and with similar remote collaboration systems could provide a more comprehensive evaluation.

6.4 Future Work

We envision a number of potential extensions to our system that point to future research. First, in *Periscope*, modes arbitrated inputs from different sources in a deterministic fashion. While this approach was sufficient to realize an instance of robotic systems based on shared camera control, future systems that integrate more complex interactions and consider more nuanced circumstances for arbitration will require more sophisticated methods for arbitration. Planning-based approaches, program verification, and optimization-based scheduling are all promising directions that future work can consider. Second, our system assumed a very specific worker-helper setup, and other configurations, such as mutual collaboration, cross-training scenarios, and experts providing remote assistance to multiple workers, will require significant extensions to *Periscope*. These are interesting and challenging scenarios that make up an exciting research space for robotic camera systems.

We envision additional capabilities for *Periscope* that will require more research. For example, while our work considered how an autonomous agent, other than the helper and the worker, might participate in arbitration, further research is needed to understand how arbitration would apply to a more, or fully, autonomous agent that controls viewpoints by taking initiative. Similarly, *Periscope* can be extended to integrate a semi-autonomous agent with the ability to capture worker actions during periods of helper inattention (due to, e.g., distraction, interruption, assisting other workers) and to provide summaries of work completed. We also found the simulated 3D view of the workspace to be underutilized by collaborators and envision enhancing the capabilities of this view for input (e.g., receiving input directly through the simulation) and output (e.g., offering interactive capabilities through a head-mounted display). Finally, the robot's actions can go beyond supporting shared visual context and include also providing physical assistance to the worker, introducing telemanipulation and autonomous manipulation capabilities to *Periscope*. Manipulation actions by the robot will introduce questions around safety and arbitration, which also serve as interesting avenues for future research.

REFERENCES

[1] Firas Abi-Farraj, Nicolò Pedemonte, and Paolo Robuffo Giordano. 2016. A visual-based shared control architecture for remote telemanipulation. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 4266–4273.

[2] Sigurdur Orn Adalgeirsson and Cynthia Breazeal. 2010. MeBot: A robotic platform for socially embodied telepresence. In *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 15–22.

[3] Matt Adcock, Stuart Anderson, and Bruce Thomas. 2013. RemoteFusion: real time depth camera fusion for remote collaboration on physical tasks. In *Proceedings of the 12th ACM SIGGRAPH international conference on virtual-reality continuum and its applications in industry*. 235–242.

[4] Deepak Akkil, Jobin Mathew James, Poika Isokoski, and Jari Kangas. 2016. GazeTorch: enabling gaze awareness in collaborative physical tasks. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. 1151–1158.

[5] Huidong Bai, Prasanth Sasikumar, Jing Yang, and Mark Billinghurst. 2020. A user study on mixed reality remote collaboration with eye gaze and hand gesture sharing. In *Proceedings of the 2020 CHI conference on human factors in computing systems*. 1–13.

[6] Virginia Braun and Victoria Clarke. 2012. *Thematic analysis.* American Psychological Association.

[7] Jerome Bruner. 1995. From joint attention to the meeting of minds: An introduction. *Joint attention: Its origins and role in development* (1995), 1–14.

[8] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. 2019. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019).

[9] François Chaumette, Seth Hutchinson, and Peter Corke. 2016. Visual servoing. In *Springer Handbook of Robotics*. Springer, 841–866.

[10] Jung Ju Choi and Sonya S Kwak. 2017. Who is this?: Identity and presence in robot-mediated communication. *Cognitive Systems Research* 43 (2017), 174–189.

[11] David B Christianson, Sean E Anderson, Li-wei He, David H Salesin, Daniel S Weld, and Michael F Cohen. 1996. Declarative camera control for automatic cinematography. In *AAAI/IAAI, Vol. 1*. 148–155.

[12] Marc Christie, Patrick Olivier, and Jean-Marie Normand. 2008. Camera control in computer graphics. In *Computer Graphics Forum*, Vol. 27. Wiley Online Library, 2197–2218.

[13] Herbert H. Clark and Catherine R. Marshall. 1981. Definite Knowledge and Mutual Knowledge. In *Elements of Discourse Understanding*, Aravind K. Joshi, Bonnie L. Webber, and Ivan A. Sag (Eds.). Cambridge, UK: Cambridge University Press, 10–63.

[14] Onno Crasborn and Han Sloetjes. 2008. Enhanced ELAN functionality for sign language corpora. In *6th International Conference on Language Resources and Evaluation (LREC 2008)/3rd Workshop on the Representation and Processing of Sign Languages: Construction and Exploitation of Sign Language Corpora*. 39–43.

[15] Owen Daly-Jones, Andrew Monk, and Leon Watts. 1998. Some advantages of video conferencing over high-quality audio conferencing: fluency and awareness of attentional focus. *International Journal of Human-Computer Studies* 49, 1 (1998), 21–58.

[16] Anca D Dragan and Siddhartha S Srinivasa. 2013. A policy-blending formalism for shared control. *The International Journal of Robotics Research* 32, 7 (2013), 790–805.

[17] Romina Druta, Cristian Druta, Paul Negirla, and Ioan Silea. 2021. A review on methods and systems for remote collaboration. *Applied Sciences* 11, 21 (2021), 10035.

[18] Mica R Endsley. 1995. Measurement of situation awareness in dynamic systems. *Human factors* 37, 1 (1995), 65–84.

[19] Martin Feick, Terrance Mok, Anthony Tang, Lora Oehlberg, and Ehud Sharlin. 2018. Perspective on and re-orientation of physical proxies in object-focused remote collaboration. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–13.

[20] Nick V Flor. 1998. Side-by-side collaboration: A case study. *International Journal of Human-Computer Studies* 49, 3 (1998), 201–222.

[21] Susan R. Fussell, Robert E. Kraut, and Jane Siegel. 2000. Coordination of Communication: Effects of Shared Visual Context on Collaborative Work. In *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work* (Philadelphia, Pennsylvania, USA) *(CSCW '00)*. Association for Computing Machinery, New York, NY, USA, 21–30. https://doi.org/10.1145/358916.358947

[22] Susan R. Fussell, Leslie D. Setlock, and Robert E. Kraut. 2003. Effects of Head-Mounted and Scene-Oriented Video Systems on Remote Collaboration on Physical Tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Ft. Lauderdale, Florida, USA) *(CHI '03)*. Association for Computing Machinery, New York, NY, USA, 513–520. https://doi.org/10.1145/642611.642701

[23] Susan R Fussell, Leslie D Setlock, and Elizabeth M Parker. 2003. Where do helpers look? Gaze targets during collaborative physical tasks. In *CHI'03 extended abstracts on Human factors in computing systems*. 768–769.

[24] Susan R Fussell, Leslie D Setlock, Elizabeth M Parker, and Jie Yang. 2003. Assessing the value of a cursor pointing device for remote collaboration on physical tasks. In *CHI'03 Extended Abstracts on Human Factors in Computing Systems*. 788–789.

[25] Susan R Fussell, Leslie D Setlock, Jie Yang, Jiazhi Ou, Elizabeth Mauer, and Adam DI Kramer. 2004. Gestures over video streams to support remote collaboration on physical tasks. *Human-Computer Interaction* 19, 3 (2004), 273–309.

[26] Steffen Gauglitz, Cha Lee, Matthew Turk, and Tobias Höllerer. 2012. Integrating the Physical Environment into Mobile Remote Collaboration. In *Proceedings of the 14th International Conference on Human-Computer Interaction with Mobile Devices and Services* (San Francisco, California, USA) *(MobileHCI '12)*. Association for Computing Machinery, New York, NY, USA, 241–250. https://doi.org/10.1145/2371574.2371610

[27] Steffen Gauglitz, Benjamin Nuernberger, Matthew Turk, and Tobias Höllerer. 2014. In touch with the remote world: Remote collaboration with augmented reality drawings and virtual navigation. In *Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology*. 197–205.

[28] William W Gaver. 1992. The affordances of media spaces for collaboration. In *Proceedings of the 1992 ACM conference on Computer-supported cooperative work*. 17–24.

[29] William W. Gaver, Abigail Sellen, Christian Heath, and Paul Luff. 1993. One is Not Enough: Multiple Views in a Media Space. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems* (Amsterdam, The Netherlands) *(CHI '93)*. Association for Computing Machinery, New York, NY, USA, 335–341. https://doi.org/10.1145/169059.169268

[30] Michael Gleicher and Andrew Witkin. 1992. Through-the-lens camera control. In *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*. 331–340.

[31] Kunal Gupta, Gun A. Lee, and Mark Billinghurst. 2016. Do You See What I See? The Effect of Gaze Tracking on Task Space Remote Collaboration. *IEEE Transactions on Visualization and Computer Graphics* 22, 11 (2016), 2413–2422. https://doi.org/10.1109/TVCG.2016.2593778

[32] Pavel Gurevich, Joel Lanir, Benjamin Cohen, and Ran Stone. 2012. TeleAdvisor: a versatile augmented reality tool for remote assistance. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 619–622.

[33] Keita Higuch, Ryo Yonetani, and Yoichi Sato. 2016. Can eye help you? Effects of visualizing eye fixations on remote collaboration scenarios for physical tasks. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 5180–5190.

[34] Seth Hutchinson, Gregory D Hager, and Peter I Corke. 1996. A tutorial on visual servo control. *IEEE transactions on robotics and automation* 12, 5 (1996), 651–670.

[35] Shu Jiang and Ronald C. Arkin. 2015. Mixed-Initiative Human-Robot Interaction: Definition, Taxonomy, and Survey. In *2015 IEEE International Conference on Systems, Man, and Cybernetics*. 954–961. https://doi.org/10.1109/SMC.2015.174

[36] Steven Johnson, Madeleine Gibson, and Bilge Mutlu. 2015. Handheld or handsfree? Remote collaboration via lightweight head-mounted displays and handheld devices. In *Proceedings of the 18th ACM conference on computer supported cooperative work & social computing*. 1825–1836.

[37] Steven Johnson, Irene Rae, Bilge Mutlu, and Leila Takayama. 2015. Can you see me now? How field of view affects collaboration in robotic telepresence. In *Proceedings of the 33rd annual acm conference on human factors in computing systems*. 2397–2406.

[38] Shunichi Kasahara, Shohei Nagai, and Jun Rekimoto. 2014. LiveSphere: immersive experience sharing with 360 degrees head-mounted cameras. In *Proceedings of the adjunct publication of the 27th annual ACM symposium on User interface software and technology*. 61–62.

[39] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. 2017. Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In *Proceedings of the IEEE international conference on computer vision*. 1521–1529.

[40] Benjamin Kenwright. 2015. Generic convex collision detection using support mapping. *Technical report* (2015).

[41] Seungwon Kim, Gun A Lee, and Nobuchika Sakata. 2013. Comparing pointing and drawing for remote collaboration. In *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 1–6.

[42] David Kirk, Tom Rodden, and Danaë Stanton Fraser. 2007. Turn It This Way: Grounding Collaborative Action with Remote Gestures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '07)*. Association for Computing Machinery, New York, NY, USA, 1039–1048. https://doi.org/10.1145/1240624.1240782

[43] Andrey Kiselev, Annica Kristoffersson, and Amy Loutfi. 2014. The effect of field of view on social interaction in mobile robotic telepresence systems. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*. 214–215.

[44] Sven Kratz and Fred Rabelo Ferriera. 2016. Immersed remotely: Evaluating the use of head mounted devices for remote collaboration in robotic telepresence. In *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 638–645.

[45] Robert E Kraut, Susan R Fussell, and Jane Siegel. 2003. Visual information as a conversational resource in collaborative physical tasks. *Human–computer interaction* 18, 1-2 (2003), 13–49.

[46] T. Kurata, N. Sakata, M. Kourogi, H. Kuzuoka, and M. Billinghurst. 2004. Remote collaboration using a shoulder-worn active camera/laser. In *Eighth International Symposium on Wearable Computers*, Vol. 1. 62–69. https://doi.org/10.1109/ISWC.2004.37

[47] Hideaki Kuzuoka, Toshio Kosuge, and Masatomo Tanaka. 1994. GestureCam: A video communication system for sympathetic remote collaboration. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*. 35–43.

[48] Hideaki Kuzuoka, Shinya Oyama, Keiichi Yamazaki, Kenji Suzuki, and Mamoru Mitsuishi. 2000. GestureMan: A mobile robot that embodies a remote instructor's actions. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*. 155–162.

[49] Joel Lanir, Ran Stone, Benjamin Cohen, and Pavel Gurevich. 2013. Ownership and Control of Point of View in Remote Assistance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) *(CHI '13)*. Association for Computing Machinery, New York, NY, USA, 2243–2252. https://doi.org/10.1145/2470654.2481309

[50] Gun A Lee, Theophilus Teo, Seungwon Kim, and Mark Billinghurst. 2017. Mixed reality collaboration through sharing a live panorama. In *SIGGRAPH Asia 2017 Mobile Graphics & Interactive Applications*. Association for Computing Machinery, 1–4.

[51] Stephen C Levinson. 2004. Deixis. In *The handbook of pragmatics*. Blackwell, 97–121.

[52] Jiannan Li, Maurício Sousa, Chu Li, Jessie Liu, Yan Chen, Ravin Balakrishnan, and Tovi Grossman. 2022. ASTEROIDS: Exploring Swarms of Mini-Telepresence Robots for Physical Skill Demonstration. In *CHI Conference on Human Factors in Computing Systems*. 1–14.

[53] Dylan P Losey, Craig G McDonald, Edoardo Battaglia, and Marcia K O'Malley. 2018. A review of intent detection, arbitration, and communication aspects of shared control for physical human–robot interaction. *Applied Mechanics*

Reviews 70, 1 (2018).

[54] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, et al. 2019. Mediapipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172* (2019).

[55] Douglas G Macharet and Dinei A Florencio. 2012. A collaborative control system for telepresence robots. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 5105–5111.

[56] Tamotsu Machino, Satoshi Iwaki, Hiroaki Kawata, Yoshimasa Yanagihara, Yoshito Nanjo, and K-i Shimokura. 2006. Remote-collaboration system using mobile robot with camera and projector. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* IEEE, 4063–4068.

[57] Danylo Malyuta, Christian Brommer, Daniel Hentzen, Thomas Stastny, Roland Siegwart, and Roland Brockers. 2019. Long-duration fully autonomous operation of rotorcraft unmanned aerial systems for remote-sensing data acquisition. *Journal of Field Robotics* (Aug. 2019), arXiv:1908.06381. https://doi.org/10.1002/rob.21898

[58] Bernardo Marques, Samuel Silva, João Alves, António Rocha, Paulo Dias, and Beatriz Sousa Santos. 2022. Remote collaboration in maintenance contexts using augmented reality: Insights from a participatory process. *International Journal on Interactive Design and Manufacturing (IJIDeM)* 16, 1 (2022), 419–438.

[59] Helena M Mentis, Yuanyuan Feng, Azin Semsar, and Todd A Ponsky. 2020. Remotely Shaping the View in Surgical Telementoring. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–14.

[60] Davide Nicolis, Marco Palumbo, Andrea Maria Zanchettin, and Paolo Rocco. 2018. Occlusion-free visual servoing for the shared autonomy teleoperation of dual-arm robots. *IEEE Robotics and Automation Letters* 3, 2 (2018), 796–803.

[61] Ohan Oda, Carmine Elvezio, Mengu Sukan, Steven Feiner, and Barbara Tversky. 2015. Virtual replicas for remote assistance in virtual and augmented reality. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. 405–415.

[62] Yuya Onishi, Kazuaki Tanaka, and Hideyuki Nakanishi. 2016. Embodiment of video-mediated communication enhances social telepresence. In *Proceedings of the Fourth International Conference on Human Agent Interaction*. 171–178.

[63] Mai Otsuki, Keita Maruyama, Hideaki Kuzuoka, and Yusuke Suzuki. 2018. Effects of enhanced gaze presentation on gaze leading in remote collaborative physical tasks. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–11.

[64] Doug Palmer, Matt Adcock, Jocelyn Smith, Matthew Hutchins, Chris Gunn, Duncan Stevenson, and Ken Taylor. 2007. Annotating with Light for Remote Guidance. In *Proceedings of the 19th Australasian Conference on Computer-Human Interaction: Entertaining User Interfaces* (Adelaide, Australia) *(OZCHI '07)*. Association for Computing Machinery, New York, NY, USA, 103–110. https://doi.org/10.1145/1324892.1324911

[65] Thammathip Piumsomboon, Gun A Lee, Jonathon D Hart, Barrett Ens, Robert W Lindeman, Bruce H Thomas, and Mark Billinghurst. 2018. Mini-me: An adaptive avatar for mixed reality remote collaboration. In *Proceedings of the 2018 CHI conference on human factors in computing systems*. 1–13.

[66] Thammathip Piumsomboon, Gun A. Lee, Andrew Irlitti, Barrett Ens, Bruce H. Thomas, and Mark Billinghurst. 2019. On the Shoulder of the Giant: A Multi-Scale Mixed Reality Collaboration with 360 Video Sharing and Tangible Interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–17. https://doi.org/10.1145/3290605.3300458

[67] Irene Rae, Leila Takayama, and Bilge Mutlu. 2013. In-body experiences: embodiment, control, and trust in robot-mediated communication. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1921–1930.

[68] Irene Rae, Leila Takayama, and Bilge Mutlu. 2013. The influence of height in robot-mediated communication. In *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 1–8.

[69] Irene Rae, Gina Venolia, John C Tang, and David Molnar. 2015. A framework for understanding and designing telepresence. In *Proceedings of the 18th ACM conference on computer supported cooperative work & social computing*. 1552–1566.

[70] Daniel Rakita, Bilge Mutlu, and Michael Gleicher. 2017. A motion retargeting method for effective mimicry-based teleoperation of robot arms. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*. 361–370.

[71] Daniel Rakita, Bilge Mutlu, and Michael Gleicher. 2018. An autonomous dynamic camera method for effective remote teleoperation. In *2018 13th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 325–333.

[72] Daniel Rakita, Bilge Mutlu, and Michael Gleicher. 2019. Remote telemanipulation with adapting viewpoints in visually complex environments. *Robotics: Science and Systems XV* (2019).

[73] Daniel Rakita, Haochen Shi, Bilge Mutlu, and Michael Gleicher. 2021. Collisionik: A per-instant pose optimization method for generating robot motions with environment collision avoidance. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 9995–10001.

[74] Abhishek Ranjan, Jeremy P. Birnholtz, and Ravin Balakrishnan. 2007. Dynamic Shared Visual Spaces: Experimenting with Automatic Camera Control in a Remote Repair Task. In *Proceedings of the SIGCHI Conference on Human Factors in*

Computing Systems (San Jose, California, USA) *(CHI '07)*. Association for Computing Machinery, New York, NY, USA, 1177–1186. https://doi.org/10.1145/1240624.1240802

[75] Troels Ammitsbøl Rasmussen and Weidong Huang. 2019. SceneCam: Improving Multi-camera Remote Collaboration using Augmented Reality. In *2019 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*. 28–33. https://doi.org/10.1109/ISMAR-Adjunct.2019.00023

[76] Mehrnaz Sabet, Mania Orand, and David W. McDonald. 2021. Designing telepresence drones to support synchronous, mid-air remote collaboration: An exploratory study. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–17.

[77] Mose Sakashita, E Andy Ricci, Jatin Arora, and François Guimbretière. 2022. RemoteCoDe: Robotic Embodiment for Enhancing Peripheral Awareness in Remote Collaboration Tasks. *Proceedings of the ACM on Human-Computer Interaction* 6, CSCW1 (2022), 1–22.

[78] Prasanth Sasikumar, Lei Gao, Huidong Bai, and Mark Billinghurst. 2019. Wearable remotefusion: A mixed reality remote collaboration system with local eye gaze and remote hand gesture sharing. In *2019 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*. IEEE, 393–394.

[79] Alexander Schäfer, Gerd Reis, and Didier Stricker. 2021. A Survey on Synchronous Augmented, Virtual and Mixed Reality Remote Collaboration Systems. *ACM Computing Surveys (CSUR)* (2021).

[80] Emmanuel Senft, Michael Hagenow, Pragathi Praveena, Robert Radwin, Michael Zinn, Michael Gleicher, and Bilge Mutlu. 2022. A Method For Automated Drone Viewpoints to Support Remote Robot Manipulation. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 7704–7711. https://doi.org/10.1109/IROS47612.2022.9982063

[81] David Sirkin and Wendy Ju. 2012. Consistency in physical and on-screen action improves perceptions of telepresence robots. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*. 57–64.

[82] Rajinder S. Sodhi, Brett R. Jones, David Forsyth, Brian P. Bailey, and Giuliano Maciocci. 2013. BeThere: 3D Mobile Collaboration with Spatial Input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) *(CHI '13)*. Association for Computing Machinery, New York, NY, USA, 179–188. https://doi.org/10.1145/2470654.2470679

[83] Maximilian Speicher, Jingchen Cao, Ao Yu, Haihua Zhang, and Michael Nebeling. 2018. 360anywhere: Mobile ad-hoc collaboration in any environment using 360 video and augmented reality. *Proceedings of the ACM on Human-Computer Interaction* 2, EICS (2018), 1–20.

[84] Christoph Stahl, Dimitra Anastasiou, and Thibaud Latour. 2018. Social Telepresence Robots: The role of gesture for collaboration over a distance. In *Proceedings of the 11th PErvasive Technologies Related to Assistive Environments Conference*. 409–414.

[85] John C Tang. 1991. Findings from observational studies of collaborative work. *International Journal of Man-machine studies* 34, 2 (1991), 143–160.

[86] Franco Tecchia, Leila Alem, and Weidong Huang. 2012. 3D helping hands: a gesture based MR system for remote collaboration. In *Proceedings of the 11th ACM SIGGRAPH international conference on virtual-reality continuum and its applications in industry*. 323–328.

[87] Theophilus Teo, Louise Lawrence, Gun A. Lee, Mark Billinghurst, and Matt Adcock. 2019. Mixed Reality Remote Collaboration Combining 360 Video and 3D Reconstruction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3290605.3300431

[88] Balasaravanan Thoravi Kumaravel, Fraser Anderson, George Fitzmaurice, Bjoern Hartmann, and Tovi Grossman. 2019. Loki: Facilitating remote instruction of physical tasks using bi-directional mixed-reality telepresence. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. 161–174.

[89] Elina Vartiainen, Veronika Domova, and Marcus Englund. 2015. Expert on wheels: an approach to remote collaboration. In *Proceedings of the 3rd International Conference on Human-Agent Interaction*. 49–54.

[90] Ana M Villanueva, Ziyi Liu, Zhengzhe Zhu, Xin Du, Joey Huang, Kylie A Peppler, and Karthik Ramani. 2021. Robotar: An augmented reality compatible teleconsulting robotics toolkit for augmented makerspace experiences. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–13.

[91] Peng Wang, Shusheng Zhang, Xiaoliang Bai, Mark Billinghurst, Weiping He, Shuxia Wang, Xiaokun Zhang, Jiaxiang Du, and Yongxing Chen. 2019. Head pointer or eye gaze: Which helps more in mr remote collaboration?. In *2019 IEEE conference on virtual reality and 3D user interfaces (VR)*. IEEE, 1219–1220.

[92] Chun Xiao, Weidong Huang, and Mark Billinghurst. 2020. Usage and Effect of Eye Tracking in Remote Guidance. In *32nd Australian Conference on Human-Computer Interaction*. 622–628.

[93] Xujing Zhang, Sean Braley, Calvin Rubens, Timothy Merritt, and Roel Vertegaal. 2019. LightBee: A self-levitating light field display for hologrammatic telepresence. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–10.

A TECHNICAL DETAILS

A.1 System Overview

The system builds on Robot Operating System (ROS)[9], which enables communication between system components, and real-time control of the robot arm. In our prototype, we mount an Azure Kinect camera[10] on a Universal Robot UR5 collaborative robot arm.[11] The camera provides both color images (which the users view) and depth data for use in computer vision algorithms. The color and depth data have a resolution of 2048x1536 and 512x512, respectively.

Our front-end interface was built using the React framework[12] and it connects to the back-end ROS server using `roslibjs`[13]. Any visual feedback that is provided on the camera feed for input commands and annotations is implemented using React Conva[14]. The 3D view, built on `ros3djs`[15], shows a simulated visualization of the robot and its surrounding objects in `threejs`[16] and updates their states in real-time from the back-end ROS server. We use Dolby's API [17] for video conferencing services. The control panel consists of five buttons that interact with the ROS back-end.

A.2 Motion Generation

We cast this real-time motion generation problem in a constrained multiple-objective optimization structure. Most helper and worker interactions and autonomous behaviors are formulated as objectives.

$$\begin{aligned} \mathbf{q} = &\arg\min_{\mathbf{q}} \sum_{i=1}^{N} w_i * f(\chi_i(\mathbf{q})) \\ &s.t.\ \ l_i \le q_i \le u_i, \ \ \forall i \end{aligned} \tag{1}$$

Here, $\mathbf{q} \in \mathbb{R}^n$ is the configuration of a n-joint robot. l_i and r_i are the upper and lower bounds of the i-th robot joint. N is the total number of objectives and w_i is the weight of the i-th objective $\chi_i(\mathbf{q})$. f is the Groove function introduced by Rakita et al. [70] that normalizes objective values for multiple-objective accommodation.

Many objectives describe camera behaviors and use a forward kinematics function $\mathbf{\Psi}$ that calculates the camera pose given a joint configuration. Forward kinematics function $\mathbf{\Psi}_p(\mathbf{q})$, $\mathbf{\Psi}_R(\mathbf{q})$, $\mathbf{\Psi}_q(\mathbf{q})$ represents the position, rotation matrix, and quaternion of the camera at joint configuration $\mathbf{q}$, respectively.

The optimized joint configuration $\mathbf{q}$ is sent to the robot arm using its native programming language, URScript. URScript additionally has commands that directly support *Reset* and *Freedrive*.

A.3 Helper Interactions

A.3.1 Target. To point the camera towards a target, we adapt the "look-at task" from prior work [71].

$$\chi_{\text{set\_target}}(\mathbf{q}) = dist(\mathbf{t}, \mathbf{v}) \tag{2}$$

[9] https://www.ros.org/

[10] https://azure.microsoft.com/en-us/services/kinect-dk/

[11] https://www.universal-robots.com/products/ur5-robot/

[12] https://reactjs.org/

[13] http://wiki.ros.org/roslibjs

[14] https://konvajs.org/docs/react/index.html

[15] http://wiki.ros.org/ros3djs

[16] https://threejs.org/

[17] https://dolby.io/

Here, function $dist()$ returns the orthogonal distance between a target position $\mathbf{t} \in \mathbb{R}^3$ and a unit vector $\mathbf{v} \in \mathbb{R}^3$ that indicates the view direction.

A.3.2 Adjust. To move the camera according to directional inputs, the objective for view adjustment is:

$$\chi_{\text{adjust}}(\mathbf{q}) = ||\Psi_p(\mathbf{q}_{t-1}) + \Delta - \Psi_p(\mathbf{q}_t)||_2 \tag{3}$$

Here, $\mathbf{q}_t$ and $\mathbf{q}_{t-1}$ are the robot joint configuration at time t and $t-1$. $\Delta \in \mathbb{R}^3$ is an offset signal.

A.3.3 Reset. To move the camera to a pre-defined starting configuration, we send the pre-defined joint configuration to the robot arm via URScript.

A.3.4 Annotate. The front-end canvas accepts input signals and overlays pin/rectangle/arrow depending on user selection of shape and subsequent movement.

A.4 Worker Interactions

A.4.1 Point. Our pointing detection functionalities are built upon the open-source MediaPipe system [54][18], in which a hand pose is represented by 21 2D landmarks. To detect pointing gestures, an algorithm checks if the distance from the base of the worker's thumb to the worker's index fingertip is larger than the base of the thumb to all the other fingertips. With pointing being detected, the pointing slider in the control panel is enabled for the helper. If the helper chooses to turn it on, the target of the camera **t** is set to the position of the index fingertip in the robot frame.

A.4.2 Freedrive. The *worker-led mode* allows the robot-mounted camera to be manually moved by the worker into a desired pose. This mode switches the robot to freedrive directly via URScript. In freedrive, the robot arm senses the forces applied to it and moves in the direction of the force as if it is being pushed or pulled by the worker.

A.5 Autonomous Behaviors

A.5.1 Keep distance. To maintain a specified distance between the camera and a target point, we used an objective from prior work [71]:

$$\chi_{\text{dist}}(\mathbf{q}) = ||\mathbf{t} - \Psi_p(\mathbf{q})||_2 - d \tag{4}$$

Here, $\mathbf{t} \in \mathbb{R}^3$ is the target position and d is the certain distance.

A.5.2 Keep upright. We adapt an objective that keeps the camera upright from prior work [71].

$$\chi_{\text{lookat}}(\mathbf{q}) = (\Psi_R(\mathbf{q})[0,1,0]^\intercal) \cdot [0,0,1]^\intercal \tag{5}$$

To keep the camera upright, the camera's "left" axis (y axis in our system) should be orthogonal to the vertical axis $[0,0,1]$ in the world frame.

A.5.3 Track hand. As described in Sec. A.4.1, we detect landmarks of the worker's hand using MediaPipe. The landmarks are converted from the camera frame of reference to the robot's frame to be used by the system. We use the average position of 5 landmarks on the worker's right hand (wrist, base of all fingers) as the target.

[18] https://google.github.io/mediapipe/solutions/hands.html

A.5.4 Avoid jerky motion. To avoid large and jittery camera motions, both joint motion and camera motion smoothness objective are included in the optimization formulation. Prior work [71] assigns equal weights to all robot joints in the joint motion smoothness objectives. However, the joint that is closer to the robot's base leads to larger camera motion, so we apply higher penalty to these joints. Consequently, the robot has more tendency to make fine movements. In our notation, a joint that is closer to the robot's base has a lower index. In our system, the objectives that minimizes joint velocity, acceleration, and jerk are:

$$\chi_v(\mathbf{q}) = \sqrt{\sum_i^n (n-i+1)\dot{q_i}^2}, \quad \chi_a(\mathbf{q}) = \sqrt{\sum_i^n (n-i+1)\ddot{q_i}^2}, \quad \chi_j(\mathbf{q}) = \sqrt{\sum_i^n (n-i+1)\dddot{q_i}^2} \tag{6}$$

We use the same objective in prior work [71] to minimize the velocity of the camera.

$$\chi_{\text{ee\_vel}}(\mathbf{q}) = ||\Psi_p(\mathbf{q}_t) - \Psi_p(\mathbf{q}_{t-1})||_2 \tag{7}$$

Although joint limits are set as inequality constraints in our formulation (Equation 1), we also add an objective to keep solutions away from joint limits.

$$\chi_{\text{joint\_limits}}(\mathbf{q}) = \sum_{i=1}^{n} 0.05 \left(\frac{(q_i - l_i)/(u_i - l_i) - 0.5}{0.45} \right)^{50} \tag{8}$$

Here, q_i, l_i and u_i are the angle, lower, and upper limit of the i-th joint, respectively.

A.5.5 Avoid collisions. We use collision avoidance methods from prior work [73] to prevent collisions between the robot arm and the objects in the environment including the worker. These methods allow collision avoidance with both static objects as well as dynamic objects such as the worker. We use the same methods to prevent collisions between the links of the robot arm (self-collisions). In prior work [73], each robot link $\mathbf{l}_i$ and environment object $\mathbf{e} \in \mathcal{A}$ is wrapped in convex hull shapes. The distance between two convex hull shapes *dist*() is computed using a Support Mapping method [40].

$$\chi_{\text{self\_collision}}(\mathbf{q}) = \sum_{i=1}^{m-2} \sum_{j=i+2}^{m} \frac{(5\epsilon)^2}{dist\left(\mathbf{l}_i(\mathbf{q}), \mathbf{l}_j(\mathbf{q})\right)^2} \tag{9}$$

$$\chi_{\text{env\_collision}}(\mathbf{q}) = \sum_{\mathbf{e}\in\mathcal{A}} \sum_{i=1}^{m} \frac{(5\epsilon)^2}{dist\left(\mathbf{l}_i(\mathbf{q}), \mathbf{e}\right)^2} \tag{10}$$

Here, m is the total number of robot links and ϵ is a scalar value that signifies the cutoff distance between collision and non-collision. For both self- and environment collision, we set ϵ as 0.02.

To detect the worker's body positions for collision avoidance, we use the open-source OpenPose[8] system to detect human body poses from RGB images. Human body poses are represented in 25 key-points in the RGB image. We map the key-points to 3D using depth data. Since the depth data can be noisy, we use a median filter to get smooth and stable body key-points. With these stable 3D key-points, we wrap body part with convex hull spaces (e.g., spheres, cuboids) for robot collision avoidance. The body parts are also visualized in the 3D view panel in the front-end interface.

To detect dynamic object positions, we use AR tags [57] to detect the poses of dynamic objects in the environment. In future system, the AR tags can be replaced by some vision-based pose estimation technologies (e.g., SSD-6D [39]).

B EXAMPLES

Example B.1 (D4). When the worker attached a component on the grid, the helper said, *"Okay...so let me just double check that it [the component] is facing the correct way,"* and moved the camera to get a better view of the grid. While moving the camera, the helper continued the conversation, *"Is the arrow...[the worker indicates the direction of the arrow with their hand]...okay...if the arrow is pointing to the right, then it's in the correct spot."* Finally, the helper completed the camera movement to get a view of the component and confirmed, *"Yeah, that looks correct to me."*

Example B.2 (D6). While attaching the wall grid to the base grid, the worker asked the helper, *"Am I doing it correct so far?"* The helper replied, *"Yeah, you are doing it correct, yeah..."* while moving the camera to get a better view of the grid. However, after getting a better look at the grid and the recently the added components, the helper said, *"Wait, just hold on a minute now...,"* and instructed the worker to make modifications, *"This part right here [a base support]...Okay, so you have to flip it."*

Example B.3 (D3). The helper struggled with wiring instructions, *"Oh, um...it should attach on the...here [adds annotation]...as well as on the inside of the triangle, like on the inside edge of the triangle that connects to the circle thing...Sorry...the thing...the clear thing with the circle on it,"* and stated, *"I wish I could like look, but I don't think there's a way to get inside the house...maybe if I do this..."* The helper moved the camera and remarked, *"Okay, I see it...sort of...,"* and instructed the worker with an annotation, *"It should attach right...here [adds annotation]."*

Example B.4 (D2). The helper took time to set up the view and prefaced the process by stating, *"Sorry...I need to adjust the camera first. This is not a very comfortable viewing angle for me."* After moving the camera, the helper continued, *"Okay, this is nice [acknowledging the view]...So first you want to fix...this L-shaped stuff [base supports]...like here [adds annotation] and here [adds annotation]."*

Example B.5 (D1). In the *robot-led mode* where the robot was tracking the worker's hand, the worker asked, *"Which drawer do you want me to open up here?"* The worker moved toward a drawer, and the helper responded, *"We don't need the blue one...we need to find us more..."* The worker then moved their hand to a different location in the workspace, changing the view. Finally, the helper switched to the *helper-led mode* and said, *"Okay, hold on... I will open mode 1 [helper-led mode]...I almost find it,"* and instructed the worker to pick up the required component, *"We have to pick up the red one."*

Example B.6 (D8). When trying to view the wall grid, the helper first moved the camera with the *helper-led mode.* The helper was mostly successful in getting a good view of the grid but finally asked the worker, *"Can you move the camera a little bit closer in Mode 3 [worker-led mode] so that I can see it better?"*

Example B.7 (D3). The helper stated to the worker, *"I'm gonna have it [the robot] follow your hand, and you're going to start opening drawers again...[worker moves hand]...one above it...[worker moves hand]...nope not that...[worker moves hand]...one above it".*

Example B.8 (D4). The worker remarked, *"Oh, here it is"*, before picking up and presenting to the helper a storage box that the dyad was searching. The helper engaged the *robot-led mode* in response to the worker's remark to track the worker's movement.

Example B.9 (D6). The worker said, *"I also think that the phototransistor might be upside down... (Helper: Is it?) ...I can show you"*, before engaging the *worker-led mode* to show the helper the phototransistor component.

Example B.10 (D5). When the helper explained the next step with the sentence *"You can connect that to the second one"*, the worker proactively changed the view to the assembly area.

Example B.11 (D2). Worker: You can turn on Mode 3 [worker-led mode], and I'll help you adjust the camera; Helper: Um...I think I can adjust the camera myself.

Example B.12 (D7). The helper made an unsuccessful attempt to inspect the roof panel and gave up, saying, *"Actually, I am not able to see the top panel...can you?...I need to look up to the panel"*.

Example B.13 (D4). The helper requested the worker, *"Do you mind manually moving the camera? So kind of in the same spot that we had it before?"*.

Example B.14 (D5). The helper requested, *"The parts that I had you collect from the organizer...can you show me that"*, when they were ready to move to the next step in the process.

Example B.15 (D6). The helper requested, *"Could you just guide me towards the side of the workbench"*.

Example B.16 (D3). In order to view the inside of a wall grid, the helper asked the worker, *"Could you point to the wall so that I can see inside it?"*.

Example B.17 (D6). When the dyad was searching for storage areas where the required component may be located, the worker remarked, *"There are some drawers over here [pointing gesture]"*.

Example B.18 (D3). The helper stated, *"Oh, it's looking at your hand and not what I want it to be looking at"*, when the view did not match their expectation of the camera aligning with the direction of pointing.

Example B.19 (D6). The helper said, *"Could you just take me back...wait I'll just take myself back,"* and reset the camera to view the assembly area.

Example B.20 (D3). After gathering the necessary components, the helper said, *"So I'm going to reset the camera...and that's the two parts that are missing,"* and proceeded to give instructions to the worker for the assembly.

Example B.21 (D4). When the helper was instructed by the experimenter to begin the next step, the helper responded with, *"Okay, let me reset the camera,"* and proceeded with the planning for the next step.

Example B.22 (D8). The helper initiated the *worker-led mode* by asking the worker, *"Can you show me the board again?"* The worker moved the camera, but the view was not adequate. The helper remarked, *"Okay, let me reset and come back again,"* reset the camera, and then used the *helper-led mode* to view the grid (which the helper refers to as the board).

Example B.23 (D1). The following dialogue took place, *"**Helper:** Okay, we are in mode two now. Did the robot detect you? **Worker:** No. You might have to reset it...,"* before the helper reset the robot and engaged the *robot-led mode.* There were no further issues with the robot tracking the worker's hand.