Towards Effective Robotic Groupware

by

Pragathi Praveena

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Computer Sciences)

at the

UNIVERSITY OF WISCONSIN–MADISON

2024

Date of final oral examination: 02/13/2024

The dissertation is approved by the following members of the Final Oral Committee:
Bilge Mutlu, Professor, Computer Sciences
Michael Gleicher, Professor, Computer Sciences
Michael Zinn, Associate Professor, Mechanical Engineering
Robert Radwin, Professor, Industrial and Systems Engineering

For Ammi.

ACKNOWLEDGMENTS

My journey through graduate school unfolded in ways I could never have imagined. I experienced incredible growth and camaraderie throughout this time, and discovered a sense of purpose and an academic niche that resonated with me. I feel grateful for the rich and fulfilling experiences that I take with me as I move on to the next part of my academic journey.

I thank my committee members — Bilge Mutlu, Michael Gleicher, Michael Zinn, and Robert Radwin — for being part of this journey. I am especially grateful to Bilge for embracing my ideas with a "yes, and" spirit and for nurturing my potential by creating space for me to explore and shape my own path. I also want to acknowledge my academic mentors from my time at IIT Madras and Xerox Research Center India, before joining graduate school, who deeply influenced my thinking and whose belief in my potential played a pivotal role in my journey to this point — Anil Prabhakar, Namita Jacob, Prathosh AP, Sanjay Bharadwaj, Saurabh Srivastava, Geetha Manjunath, and Manish Gupta. I would like to acknowledge the funding from the NSF and NASA ULI, as well as our collaborators at Boeing, which made this dissertation work possible.

I had an absolute blast working with some of the best people. Yeping Wang, Haoming Meng, and Nathan White made the work in this dissertation not only better but also so much more enjoyable with their presence. Daniel Rakita, Emmanuel Senft, and David Porfirio generously lent an ear when I needed advice. Andrew Schoen, Arissa Sato, and Yaxin Hu broadened my horizons through our research conversations. Thank you, Lily Reback, Yuna Hwang, Zejun Zhou, Rainy Jin, Guru Subramani, and Luis Molina, for being part of my various projects. Thank you, Aditya Barve and Luke Swanson, for joining me on a wild ride with GEF. Thank you, Dakota Sullivan, Christine Lee, Hailey Johnson, and Amy Koike, for being my loudest cheerleaders. Thank you, Michael Hagenow, for being the best cohort buddy. And thank you, Bengisu Cagiltay, for being there through the best and worst parts of my journey. I would also like to thank the various individuals in the People and Robots Lab and the Graphics Lab with whom I did not work directly,

but who enriched my graduate school experience. I am profoundly thankful for my peer mentoring experiences in both labs, which brought light and joy into my life during challenging times.

No amount of words would be sufficient to thank my family and friends for supporting me through this journey in so many different ways. Thank you, Settlers of Madison and my Madison–Lassen friends and their friends, for enriching my Madison experience. Thank you, Varshaa, Prakruti, Shreya, Raghavi, Amala, Tejasvin, and Julia, for being a part of my life despite the distance and the busyness. I owe you my sanity. Thank you, Ashwini, for lending me your beautiful home to write this document. Thank you, Adarsh, for the time we shared. Finally, my family — Ammi, Appi, Ajju, and Jaju — I thank you for being my constant and for taking care of me. Thank you for always believing in my dreams, even when they took me farther away from you.

DECLARATION

The research presented in this dissertation was funded in part through a National Science Foundation (NSF) award 1830242 and in part by a NASA University Leadership Initiative (ULI) grant awarded to the UW-Madison and The Boeing Company (Cooperative Agreement # 80NSSC19M0124).

I was the lead researcher and primary contributor for the work presented in this dissertation. However, I collaborated with the following individuals on the ideation, implementation, evaluation, and preparation of manuscripts for this work: Emmanuel Senft, Yeping Wang, Haoming Meng, Lily Reback, Bengisu Cagiltay, Nathan White, Jill Streamer, Richard Gardner, Andrew Schoen, David Porfirio, Michael Gleicher, and Bilge Mutlu. I elaborate on substantial contributions by four student researchers below.

- Haoming Meng: Haoming contributed significantly to the work presented in Chapter 3, including implementing the front-end user interface and deploying the system to the Internet. Haoming is also the primary author of a demonstration paper for the system (Meng et al., 2023).
- Nathan White: Nathan and I closely collaborated throughout the entire research process for the work presented in Chapter 5, including designing and conducting the study, analyzing the data, and preparing the manuscript.
- Yeping Wang: Yeping contributed to the implementation of safety features for the system presented in Chapter 3 and the preparation of two papers, Praveena et al. (2023c) and Meng et al. (2023). Additionally, Yeping contributed significantly to the engineering efforts required to maintain the system.
- Lily Reback: Lily contributed to the work presented in Chapter 4 as an experimenter and a qualitative data coder.

In this dissertation, I use ’we’ for the description of our collective work.

The work presented in this dissertation is based on the following previously published papers: *Periscope* (Praveena et al., 2023c) and *Demonstrating Periscope* (Meng et al., 2023). The work presented in Chapter 5 is based on a manuscript in preparation (Praveena et al., 2024). The discussion in Chapter 6 includes research that was previously published in Praveena et al. (2023b).

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Collaboration is a cornerstone of human progress, with collective efforts leading to outcomes that surpass what could be achieved individually. The term *groupware*, frequently used in early research in computer-supported cooperative work (CSCW), refers to tools specifically designed to support collaborative activities. Historically, emerging technologies have been pivotal in advancing groupware research and facilitating novel approaches to collaboration. For example, over the past three decades, the Internet and related web technologies catalyzed a new era of *remote collaboration*. Remote collaboration tools were indispensable during the recent global pandemic, and they made remote work possible for a wide range of tasks. However, this period also highlighted a gap in the tools available for remote collaboration in jobs involving *physical work,* such as manufacturing and healthcare.

In parallel, advances in robotic technology have resulted in collaborative robots, or *cobots*, that are highly capable and designed for safe interaction with humans in shared spaces. Cobots present new opportunities for creating groupware, especially in physical work contexts. Compared to other emerging technologies being explored for similar purposes, such as augmented and virtual reality, cobots can uniquely leverage their physical form to extend human ability in remote spaces. Thus, my research explores this nascent and promising paradigm of *robotic groupware* through the development and evaluation of a robotic camera system called *Periscope* to support *remote* and *real-time* human collaboration. In this dissertation, I contextualize my research in scenarios where experts assist novices in manual assembly tasks. Using the *Periscope* system, a worker performs manual tasks with guidance from a remote expert who views the workspace through a camera mounted on a cobot arm co-located with the worker. The dynamic view provided by the robotic camera allows both collaborators to share task-relevant visual information and develop a mutual understanding during the collaboration process.

This dissertation describes the *design*, *usage*, and *application* of the *Periscope* system with the aim of characterizing a novel and promising point in the design space of robotic groupware. First, I describe the system's design with an emphasis on

the shared control of the camera by the worker, the expert, and the robot. Our approach is key to leveraging the advanced capabilities of cobot platforms without overwhelming users with the tool’s complexity and allowing them to maintain the desired level of control. Next, I describe a human-subjects study aimed at understanding the promise of this shared camera control approach and the system’s ability to support remote collaboration. Qualitative insights from the study, including patterns of use of the system’s features derived from users’ open-ended and natural exploration of the system, offer a valuable understanding of collaborative dynamics within this new paradigm of group work mediated by the *Periscope* system. Finally, I present a second human-subjects study that involved instructors and trainers of technical skills in order to explore the potential of the *Periscope* system for facilitating remote workforce training in manufacturing environments. The qualitative insights from this study help characterize real-world applications where a solution like *Periscope* may be useful and where it may face limitations.

The work presented in this dissertation translates concepts from the CSCW and robotics literature into an end-to-end, operational robotic groupware system and delivers valuable contributions to both communities. For the CSCW community, this research broadens the scope of work supported by groupware by leveraging the unique capabilities of cobot platforms. For the robotics community, it demonstrates the feasibility of a novel application of cobots and the potential for new paradigms of collaborative work.

1 INTRODUCTION

Collaboration is a cornerstone of human progress, with collective efforts leading to outcomes that surpass what could be achieved individually. In 1984, Irene Greif and Paul Cashman coined the term *computer-supported cooperative work* (CSCW) to describe a research area concerned with the design and use of technologies to support multiple individuals working together (Grudin, 1994). Greif, in her article on the origins of CSCW (Greif, 2019), emphasized the need for a research area distinct from human-computer interaction (HCI). This field of CSCW research extends its inquiry beyond individual interactions with computers, and considers the challenges and influences that technology introduces to human-human interactions within groups and organizations.

To understand some of these challenges and influences examined in CSCW, let us consider instant messaging through applications such as WhatsApp, Slack, or Discord. Instant messaging is a well-established paradigm that has *supported* and *shaped* modern interpersonal communication. *Awareness* (Schmidt, 2002) is a key concept in this context. In instant messaging, awareness includes an understanding of the activities, states, and intentions of others involved in a conversation. Cues such as online statuses, typing indicators, and read receipts inform users about the availability and responsiveness of others in the conversation. Another relevant concept is *conversational grounding* (Clark and Marshall, 1981), a process where individuals establish mutual understanding or common ground during communication. Acknowledgments are one example of grounding. In face-to-face conversations, people might nod or say “uh huh,” to signal that a phrase has been understood and that the conversation can proceed. In messaging, this role is fulfilled by digital cues such as a *thumbs-up* emoji. However, the *asynchronous* nature of messaging, where individuals in a conversation need not be concurrently active, can disrupt awareness and grounding. Ongoing research in CSCW aims to characterize such complexities, for example, Warner et al. (2021) explore what support for editing and deleting messages (asynchronous features) in instant messaging is necessary and how this impacts people’s awareness and grounding in communication.

Groupware is a term frequently used in early CSCW research to describe tools specifically designed to support collaborative activities (Greif, 1988). Historically, emerging technologies have been pivotal in advancing groupware research and facilitating novel approaches to collaboration. For example, over the past three decades, the Internet and related web technologies catalyzed a new era of *remote collaboration*. Remote collaboration tools, such as those for video conferencing and digital payments, were indispensable during the recent global pandemic and they made remote work possible for a wide range of tasks, especially in the information economy (Yang et al., 2022). However, this period also highlighted a gap in the tools available for remote collaboration in jobs involving *physical work,* such as manufacturing and healthcare (Sostero et al., 2020).

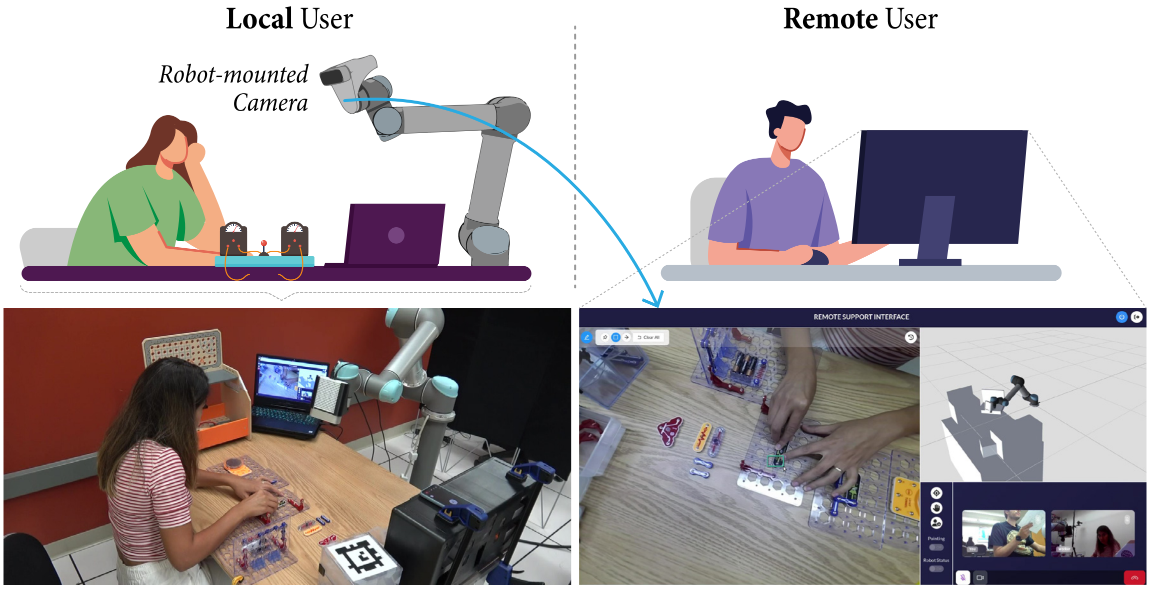
In parallel, advances in robotic technology have resulted in collaborative robots, or *cobots* (see the robot in Figure 1.1), that are highly capable and designed for safe interaction with humans in shared spaces. While cobots have predominantly found applications in automated task solutions (Michaelis et al., 2020), they present new opportunities for creating groupware, especially in physical work contexts. Compared to other emerging technologies being explored for similar purposes, such as augmented and virtual reality (see Ens et al. (2019) and Schäfer et al. (2022) for reviews), cobots can uniquely leverage their physical form to extend human ability in remote spaces. Harnessing cobots as a platform to develop groupware is particularly promising in the manufacturing domain, where their increasing prevalence and acceptance position cobots as a key enabling technology for new paradigms of work (Maddikunta et al., 2022).

In the context of enabling remote physical work, there is extensive research in robotics on control paradigms and user interfaces for *teleoperation* (described in detail in Chapter 3), where robots are operated at a distance to enable remote exploration and manipulation (Niemeyer et al., 2016). However, similar to the distinction between HCI and CSCW, developing *robotic groupware* for supporting collaborative work necessitates considerations beyond the interactions between a robot and an individual (*e.g.,* the teleoperator). The prevailing focus on using cobots (and other robot form factors) for automated solutions or individual interactions has resulted in a lack of established principles for integrating robots into groupware design, accompanied by a limited understanding of the dynamics within groups when robots are introduced (Sebo et al., 2020). Addressing this gap, our work draws from the significant advancements in robotic teleoperation, but its methods and evaluations distinctly prioritize the goal of supporting human-to-human collaboration.

The work in this dissertation is also informed by a CSCW application previously explored in human-robot interaction (HRI) — the use of mobile *telepresence* robots for facilitating remote interpersonal communication. This application is essentially video conferencing through a telepresence robot. An interface allows the remote user to control the movement of the robot and the positioning of its onboard camera(s) to explore and understand the local user's surroundings and communicate with them. These robots improve remote communication by providing a physical embodiment (Rae et al., 2013a; Kuzuoka et al., 2000) 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 (Choi and Kwak, 2017).

There is a rich literature (described in detail in Chapter 2) on how telepresence robots and interfaces should be designed to support communication between people. However, these design choices are dependent on the form factor and capabilities of the robot being used. For example, a prototypical telepresence robot is designed to emulate face-to-face communication and often takes the form of a screen on wheels that is roughly human-sized in height; the height may be adjustable to allow the remote user to interact at eye level with the local user, who may be seated or standing (Tsui and Yanco, 2013). Other robot form factors (*e.g.,* a cobot arm) and their unique capabilities (*e.g.,* increased range of motion, dexterity, precision, and repeatability) offer an underexplored design space of interaction techniques to support a wider range of remote collaborative work that includes physical tasks.

Expanding on existing work in teleoperation and telepresence, this dissertation explores a novel and promising point in the design space of robotic groupware. This is realized through the development and evaluation of a robotic camera system called *Periscope*, which leverages the capabilities of cobot techonology to facilitate *remote* and *real-time* human collaboration on physical work. *Periscope* enables a local



Local User's Workspace *Remote User's Interface*

Figure 1.1: With *Periscope*, a local worker completes an assembly task with guidance from a remote helper who views the workspace through a robot-mounted camera. The *Periscope* system facilitates remote collaboration by providing the worker and the helper with shared visual information that enhances their verbal communication and coordination processes.

user, engaged in physical work and co-located with a robotic arm, to collaborate with a remote user who views the local workspace through a camera mounted on the robot. The *Periscope* system facilitates collaboration by providing the worker and the helper with shared visual information that enhances their verbal communication and coordination processes (see Figure 1.1).

The *Periscope* system serves as a high-fidelity prototype to investigate an interesting point within the design space of robotic groupware. This dissertation describes the *design* process for creating the prototype, its *usage* in a user study that preserves key elements of a natural collaboration scenario within a lab setting, and its *application* to remote workforce training in manufacturing environments. Through the design, usage, and application of the *Periscope* system, this dissertation offers insights into a novel point in the design space of robotic groupware, specifically groupware built on cobot platforms and their potential for enabling remote collaborative work. While there is plenty of exploration to be done to deepen our understanding of this specific point and the broader design space, this dissertation begins to establish a vocabulary and framework for more general-purpose robot-supported collaborative work and lays the groundwork for how we can build effective robotic groupware.

The work presented in this dissertation translates concepts from the CSCW and robotics literature into an end-to-end, operational robotic groupware system and delivers valuable contributions to both communities. For the CSCW community, this research broadens the scope of work supported by groupware by leveraging the unique capabilities of cobot platforms. For the robotics community, it demonstrates the feasibility of a novel application of cobots and the potential for new paradigms of collaborative work.

1.1 Research Context

The design, usage, and application of groupware are inherently tied to the context in which it operates (Neale et al., 2004). Thus, this section outlines the research context that forms the foundation of the work presented in this dissertation. The described context aims to strike a balance: guiding the tool's design towards addressing the unique needs and challenges of its specific context while remaining broad enough for meaningful exploration of the tool's usage and application. The hope is that the methods and insights derived from this research context will have broader implications for other contexts, as discussed in Chapter 6, while remaining specific enough to guide the work in this dissertation.

Users

We focus on the simplest unit of collaboration, the *dyad*, involving two individuals: one *local* and one *remote*. In this setup, the experiences of the two users differ, a detail that will become more evident with the definitions of the local and remote users in §Scenario. We do not impose any requirements regarding prior acquaintance between the collaborators.

According to Lee and Paine (2015), increasing the number of collaborators, even with modular and easily extensible technologies, inherently introduces complex social dynamics and an increased cognitive burden (discussed further in the context of *collaboration load* in Chapter 3). Thus, the dyad serves as a good starting point for studying a new collaborative paradigm.

Task

Our focus on enabling physical work in manufacturing leads us to consider tasks involving the manipulation of physical artifacts, such as wiring and assembly tasks. In our setup, the *local* user engages in physical work and has access to the physical workspace, artifacts, and tools, while the *remote* user does not.

Scenario

In this work, we focus on the *remote expert* scenario. Here, the local user engaged in physical tasks has less expertise or knowledge compared to the remote user, who offers guidance to the local user. The *remote expert* scenario was central to early CSCW research (*e.g.,* Karsenty (1999); Fussell et al. (2000, 2003b); Kraut et al. (2003)) and continues to be a popular area of research, for instance, in applications of mixed reality (Ens et al., 2019). This provides us with a substantial body of existing research to draw upon. Additionally, this scenario is a prevalent focus in commercial applications such as Microsoft's Remote Assist[1] and Salesforce's Visual Remote Assistant[2] to enable technicians doing physical work to connect with remote experts. Based on this *remote expert* scenario, we can now define the roles of the local and remote users.

[1] `https://dynamics.microsoft.com/en-us/mixed-reality/remote-assist/`

[2] `https://www.salesforce.com/products/visual-remote-assistant/`

Local User: Situated in the physical workspace and performs work involving the manipulation of physical artifacts; also referred to as the *worker* or *student*.

Remote User: Situated in a different space without direct access to the physical workspace and provides assistance and instruction to the local user; also referred to as the *helper*, *expert*, or *instructor*.

Platform

This dissertation focuses on exploring the feasibility of using cobots as an enabling technology for groupware. Thus, central to this work is a 6-degree-of-freedom (6-DoF) cobot arm situated in the physical workspace of the local user. The use of a 6-DoF cobot is crucial (as opposed to a lower-DoF cobot) as this range of motion enables the cobot to reach a wide and continuous set of positions and orientations within a 3D workspace. In this work, the cobot is stationary. The cobot is mounted on caster wheels for manual repositioning, but its mobility is otherwise limited.

We augment the cobot with a camera that allows remote users to visually explore the local workspace and establish awareness and common ground (Kraut et al., 2003). While the inclusion of a gripper for remote manipulation was a viable option, concerns regarding its safety in a collaborative setting led to its exclusion in the initial exploration of this paradigm.

Time and Space

This work focuses on enabling *remote* (or *distributed*) and *real-time* (or *synchronous*) collaboration (Johansen, 1988). Similar to the *remote expert* scenario, there is a substantial body of existing research on real-time, distributed groupware in CSCW to draw upon (*e.g.,* Schäfer et al. (2022)). In addition, this context appears particularly well-suited for robots due to their capacity to extend people's abilities in remote spaces and we can leverage advances in teleoperation research to achieve real-time, remote control of robots.

Environment

The physical setup for this work is a tabletop workspace positioned approximately within the reach of a cobot arm. While this setup is primarily constrained by the robot's capabilities (reach and mobility), it reflects many realistic collaborative environments. We specifically focus on cluttered 3D workspaces, as the scale and complexity of these environments can effectively showcase the potential of a cobot platform. Within this environment, there are various sources of visual information: visible and manipulable objects utilized in task execution, and the worker's face, hands, and body language (Fussell et al., 2003b).

This research context reflects a variety of real-life applications such as a student in a lab course receiving assistance from a remote instructor, a technician on the factory floor seeking instruction from a remote expert to troubleshoot a machine issue, or a surgeon in a remote location providing guidance to a less experienced surgeon during a complex medical procedure.

1.2 Thesis Statement

This dissertation explores a novel and promising point in the design space of robotic groupware. My thesis statement is as follows: **Robotic camera systems, as a form of robotic groupware, can support human-to-human collaboration in remote settings that involve physical tasks.**

The *Periscope* system described in Chapter 3 serves as a novel and operational instantiation of the **robotic camera system** in the thesis statement. A key aspect of the design of the *Periscope* system is a *shared camera control* approach where the control of the camera is dynamically distributed between the local user, the remote user, and the robot depending on task needs. By adopting this approach, we can leverage the advanced capabilities of cobot platforms without overwhelming users with the complexity of the tool. Simultaneously, this approach enables users to maintain the desired level of control expected from a collaborative tool.

Chapter 4 provides qualitative insights derived from an exploratory evaluation of the *Periscope* system in a lab study. These insights serve as evidence of the promise of shared camera control and demonstrate the system's ability to **support human-to-human collaboration**. The study was designed to encourage users to engage in an open-ended exploration of the system's capabilities. This approach uncovered realistic insights about the collaboration dynamics within this new paradigm of group work mediated by the *Periscope* system.

Finally, in Chapter 5, qualitative insights derived from a second user study help to characterize real-world **remote settings that involve physical tasks** where a solution like *Periscope* may be effective and where it may face limitations. This user study focused on understanding the application of *Periscope* to remote workforce training in manufacturing environments and engaged potential participants of this application, such as engineering students, instructors from a technical college, and trainers from a manufacturing company. The insights from this study suggest the potential utility of robotic camera solutions for remote workforce training.

1.3 Research Methodology

At a high-level, this dissertation presents the design (Chapter 3), usage (Chapter 4), application (Chapter 5) of the *Periscope* system. In each chapter, we employed various techniques grounded in human-centered design.

Section 3.7 presents a discussion of the research methodology adopted for the design of the *Periscope* system. Here, I outline the key elements. In Chapter 3, we aimed to develop a Minimum Viable Product (MVP) (Ries, 2011) through a highly-iterative design process inspired from the *design sprint* methodology (Banfield et al., 2015; Knapp et al., 2016). This process involved iterating over five phases: understand, diverge, converge, build, and test. In the *understanding* phase, we defined our research context and identified needs by reviewing existing methods and systems in research literature. During the *diverging* phase, we developed various low-fidelity prototypes of a new feature to explore multiple solutions. During the *converging* phase, we selected the most promising solution based on rapid testing. The *building* phase saw the development of a higher-fidelity prototype, and the *testing* phase focused on integrating and validating this new feature within the existing system. We cycled through these stages multiple times to develop an MVP. Upon establishing an MVP, we shifted our focus to refining the entire system to make it operational.

Section 4.2 presents an in-depth motivation for the methodology that we adopted to understand system *usage*. Here, I outline the key elements. Chapter 4 describes a user study where participants were recruited to interact with our operational system. Our study design drew inspiration from the principles of *experimental simulations* (Neale et al., 2004) and aimed to maintain key elements of a natural collaboration scenario within a laboratory setting. Experimental simulations allow for repeated observations in a controlled environment while reflecting the context in which the system is intended to be used. The study's design also incorporated the concept of *formative-qualitative-opportunistic* evaluation (Twidale et al., 1994). *Formative* evaluations focus on assessing the interaction design (user experience while engaging with the system) rather than the system itself. *Opportunistic* evaluations allow users to explore the system in an open-ended and natural manner. Our empirical findings about system usage are rooted in *qualitative* data, such as conversations between collaborators and observed actions, gathered during our formative evaluation of users' opportunistic use of the *Periscope* system.

Section 5.2 presents the rationale behind the research methodology that we adopted to understand the *application* of a solution like the *Periscope* system in real-world settings. Here, I outline the key elements. Chapter 5 describes a user study in which the participants were potential real-world users of our proposed paradigm, such as engineering students, instructors from a technical college, and trainers from a manufacturing company. This type of study design is reminiscent of *participatory research*, which uses *"systematic inquiry in direct collaboration with those affected by an issue being studied for the purpose of action or change"* (Vaughn and Jacquez, 2020). We used critical reflections from participants on the utility of the system to characterize usage contexts related to remote workforce training in manufacturing environments where a robotic camera solution may be useful or face limitations. This process of characterization draws inspiration from *context of use analysis* (Maguire, 2001a). We employed the *Periscope* system as a *technology probe* (Hutchinson et al., 2003) to provide participants with hands-on experience and inspire discussion on the real-life opportunities and implications of our work.

1.4 Contributions

This dissertation makes the following contributions:

- *Survey* — a review of state of the art in HCI, CSCW, HRI, and robotics literature relevant to using collaborative robots as enabling technology for groupware (Chapter 2, Section 3.2).

- *Design* — the shared camera control approach and a set of design goals to realize this approach (Chapter 3).

- *System* — *Periscope*, an operational robotic camera system that is an instantiation of shared camera control with user interactions, autonomous behaviors, and arbitration policies designed for a specific context of use (Chapter 3).

- *Method* — an experimental paradigm to study the usage of a system while preserving key elements of a natural collaboration scenario (Chapter 4).

- *Data* — empirical observations on system use and their contribution to the design goals (Chapter 4).

- *Method* — an experimental paradigm to study the potential utility of a system by employing it as a technical probe and eliciting participatory critique (Chapter 5).

- *Data* — a reflective characterization of usage contexts related to remote workforce training in manufacturing environments where a robotic camera solution may be useful or face limitations (Chapter 5).

- *Dataset* — video recordings that exemplify human-human interaction mediated by a robotic camera system (Chapter 4, Chapter 5).

1.5 Overview of Dissertation

The remainder of this dissertation is organized into five chapters. Chapter 2 provides relevant background related to models and frameworks on collaboration, as well as technology that supports collaborative work. Chapter 3 presents the design and development of the *Periscope* system, addressing the central challenge of designing control mechanisms and policies that ensure effective utilization of a cobot within a robotic camera system. Both Chapters 4 and 5 present human-subjects research and are organized similarly. Chapter 4 presents an evaluation of the *Periscope* system in a lab study to understand the promise of the shared camera control approach in facilitating remote collaboration. Chapter 5 presents a user study exploring the potential application of a robotic camera solution in supporting remote workforce training in manufacturing environments. Finally, Chapter 6 provides a general discussion about the work in the previous chapters, including the significance and limitations of the work and directions for future research on robotic groupware.

2 BACKGROUND

This section provides background from the CSCW and HRI literature relevant to the concept of using robots as enabling technology for groupware. In Section 2.1, I discuss models and frameworks on collaboration and provide a brief review of collaborative systems that do not include robots. In Section 2.2, I discuss prior work that has explored the integration of robots into collaborative systems.

Chapters 3–5 provide further background and related work specific to the design, usage, and application of a prototype robotic groupware system.

2.1 Collaboration and Collaborative Systems

This section provides relevant background that defines and characterizes collaboration and groupware, highlights the need for shared visual information for successful collaboration, and offers a brief review on technology that supports collaboration.

Terminology

Although the terms *coordination*, *cooperation*, and *collaboration* are frequently used synonymously and are related concepts, they refer to distinct processes in the CSCW literature. Shah (2014), in his book on collaborative information seeking, proposed a model that distinguishes these terms. The model includes five processes: *communication*, *contribution*, *coordination*, *cooperation*, and *collaboration*, with each process being supported by its preceding processes. Below, I will present an example in the context of a group chat on a messaging application for each definition from Shah (2014) for the five processes.

Communication: This involves the process of *"sending or exchanging information."* For example, sending a message to inform the group about an upcoming event, which could lead to some coordination of activities in the future.

Contribution: This is a process by which *"individuals help each other in achieving their personal goals."* For example, if someone in the group chat asks about interesting events on the weekend, one could contribute by sending information about an upcoming event (communication) to fulfill the person's request for information.

Coordination: This process involves *"connecting different agents together for a harmonious action."* For example, within a group, an individual holding a ticket for an upcoming event, which they cannot attend, may choose to offer it to another person (contribution), and so the two individuals may coordinate a suitable time and place for the ticket handover.

Cooperation: In this process, *"different agents with similar interests take part in planning activities, negotiating roles, and sharing resources to achieve joint goals"* while adhering to *"some rules of interaction."* Central to cooperation is the pursuit of joint goals, which distinguishes it from previous processes. For example, a group cooperates to organize a birthday celebration. Each member is assigned a specific task related to decorations, food, or music and works relatively independently, but in harmony, towards the joint goal of a successful event.

Collaboration: This process involves *"creating a solution that is more than merely the sum of each party's contribution."* Collaboration often involves several cooperative acts. Collaboration is characterized by a high level of interdependence and shared ownership of the process and the outcomes. For example, when organizing the birthday celebration, collaboration (in contrast to cooperation) would involve a more collective approach where tasks are not just divided but also discussed, adjusted, and executed as a group.

The purpose of presenting these definitions is not solely to discuss their semantic distinctions but also to highlight that groupware can facilitate a spectrum of interdependent group work. Higher levels of interdependence in group work demand more complex tools that can support collective efforts towards a shared goal.

Frameworks

There are various frameworks in CSCW that are useful for describing collaborative situations and the systems that support them. Johansen (1988) introduced the *time-space* matrix to characterize groupware. This matrix categorizes groupware into four quadrants based on *time* (whether people work together at the same time or different times) and *space* (whether people work together in the same place or different places). The focus of the work in this dissertation lies in the quadrant of the matrix that corresponds to distributed (or remote; different places) and synchronous (or real-time; same time) collaboration.

However, time and space alone cannot fully capture the nuances of collaborative scenarios. Andriessen (2012) expanded the time-space matrix by incorporating the types of processes that the technology supports, including communication, coordination, and cooperation. Lee and Paine (2015) proposed a Model of Coordinated Action, encompassing seven dimensions including group size and whether the group work is intended to be short-term or long-term. Thoravi Kumaravel and Hartmann (2022) introduced the concept of an extended collaborative space to the time-space matrix to account for mixed reality environments where people may work in physical or virtual spaces. These more sophisticated models of collaborative activity help us understand the factors that affect the *design* and *impact* of groupware. The remainder of this subsection introduces two concepts critical to the design of groupware: *awareness* and *control*.

Awareness: *Situation awareness* theory by Endsley (1995) defined three levels of awareness: *perception* of relevant elements of the environment, *comprehension* of these perceived elements, and *projection* of the future states of these elements. *Workspace awareness* is a specialization of situation awareness tailored to real-time collaborative workspaces by Gutwin and Greenberg (2002). In collaborative settings, individuals create and sustain awareness to determine the current state of work and anticipate others' actions. Groupware design requires the determination of what awareness information is relevant to the collaborative work, how to obtain it, when and where it is required, and how to display it.

Control: Rodden and Blair (1991) highlighted the importance of *control* in groupware. Control refers to the management of collaborative activities and coordination among users. For example, Aditham et al. (1997) described several control policies addressing situations where a user seeks a resource currently held by a different user. One policy (called *explicit-implicit*) allows the user to explicitly request the resource, leading the system to automatically grant it by revoking it from the current holder. Another policy (called *implicit-explicit*) involves the system autonomously detecting the need for the resource and formally requesting it from the current holder. Explicit control grants users the ability to modify the coordination of group activities, whereas implicit control involves coordination by the system dictated by its rules. Rodden and Blair (1991) noted that collaboration on unstructured problems likely required more flexible control compared to routine, prescriptive tasks. He also suggested the potential for a dynamic distribution of control among participants and the system.

Chapter 3 provides additional background on awareness and control to motivate the design of awareness support and control mechanisms in the *Periscope* system.

Shared Visual Context

According to Ellis et al. (1991), groupware should provide an interface to a *shared environment* — a space accessible to all group members where they can interact and work together. Systems for distributed work can also create shared digital spaces for interaction. For instance, a shared document in a collaborative editor serves as a shared environment with the necessary features to facilitate collaboration, such as simultaneous editing, real-time updates, and version control. This document also provides users with a *shared visual context* by allowing them to access and view the same visual information simultaneously. Shared visual context is task-relevant visual information that collaborators have in common to augment their verbal communication[1] and improve collaborative outcomes (Fussell et al., 2000).

[1] In the activities within our research context (as described in Section 1.1), verbal communication is the primary medium through which information is exchanged (Flor, 1998; Kraut et al., 2003).

Previous research (Tang, 1991; Daly-Jones et al., 1998; Flor, 1998; Kraut et al., 2003) emphasizes the critical role of shared visual context for successful collaboration. The findings from these studies suggest that people use shared visual context for two coordination processes: *situation awareness* and *conversational grounding*. According to situation awareness theory by Endsley (1995), 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 (1981), 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* (Bruner, 1995), where they have a shared focus on an object.

Enabling Technologies

Systems that support remote collaboration facilitate the sharing of visual context to enable effective cooperation and communication between users (see Druta et al. (2021) for a review). This subsection presents a brief review of technologies that enable visual information capture, visual information display, and communication cues in *non-robotics groupware* that support two remote users seeking to accomplish synchronous collaboration over physical artifacts. The work in this dissertation draws from the design choices made in these systems and I discuss relevant opportunities and challenges associated with different technologies for providing a shared visual context between collaborators. Prior work that has explored how robotic technology can facilitate remote collaboration is discussed in Section 2.2.

Technologies for Visual Information Capture: Prior systems have used fixed-view cameras (Fussell et al., 2000, 2004; Kirk et al., 2007), head-mounted cameras (Fussell et al., 2003a; Johnson et al., 2015a; Gupta et al., 2016), shoulder-mounted cameras (Kurata et al., 2004; Piumsomboon et al., 2019), hand-held cameras (Gauglitz et al., 2012; Sodhi et al., 2013; Marques et al., 2022), multiple cameras (Gaver et al., 1993; Fussell et al., 2003a; Giusti et al., 2012; Rasmussen and Huang, 2019), pan-tilt-zoom (PTZ) cameras (Ranjan et al., 2007; Palmer et al., 2007), 360° cameras (Kasahara et al., 2014; Piumsomboon et al., 2019) or depth cameras (Adcock et al., 2013; Teo et al., 2019) 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 (Gupta et al. (2016); Tecchia et al. (2012); Wang et al. (2019); see Xiao et al. (2021) 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.

Technologies for Visual Information Display: Prior systems have used 2D views (Fussell et al., 2000, 2004; Kirk et al., 2007; Ranjan et al., 2007), 3D views (Adcock et al., 2013; Gauglitz et al., 2014), 360° views (Kasahara et al., 2014; Lee et al., 2017), Virtual Reality (VR) (Tecchia et al., 2012), Augmented Reality (AR) (Sodhi et al., 2013; Johnson et al., 2015a; Gupta et al., 2016), Mixed Reality (MR) (Oda et al., 2015; Thoravi Kumaravel et al., 2019; Piumsomboon et al., 2019; Teo et al., 2019; Bai et al., 2020), and projected AR (Gurevich et al., 2012; Machino et al., 2006; Speicher et al., 2018) for the display of shared visual information (see Ens et al. (2019) and Schäfer et al. (2022) for a reviews). AR and projected AR are typically used for situated information display to the worker who handles the physical artifacts.

VR, AR, and MR solutions provide users with a highly immersive experience. However, 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 (Teo et al., 2019) to leverage the benefits and reduce the drawbacks of each approach.

Technologies for Communication Cues: The primary communication channels in remote collaboration systems are typically verbal and visual. Additionally, Fussell et al. (2004) 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 (Bai et al., 2020; Tecchia et al., 2012; Teo et al., 2019) or specified through annotations (Fussell et al., 2003c; Kim et al., 2013; Gauglitz et al., 2014). 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 (Bai et al., 2020; Akkil et al., 2016), the worker's eye gaze (Gupta et al., 2016; Sasikumar et al., 2019), or viewing direction (Higuch et al., 2016; Otsuki et al., 2018). Other interesting communication cues include virtual replicas of task objects (Oda et al., 2015) or human avatars to provide non-verbal cues in MR (Piumsomboon et al., 2018).

2.2 Robots in Collaborative Systems

Prior work has explored how robots can help capture and display visual information and communication cues to facilitate remote collaboration. Early work by Kuzuoka et al. (1994) 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. (2018) used a robotic arm to reproduce orientation manipulations on a proxy object at a remote site. In this setup, when a remote user manipulated a physical object, the manipulations were mirrored with a proxy object held by a robot arm for the local user. While this solution improves the local user's spatial understanding of the object, the method is hard to scale beyond one object. Gurevich et al. (2012) and Machino et al. (2006) 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. Sirkin and Ju (2012) and Onishi et al. (2016) explored the use of a robotic arm to display gestures such as pointing to and touching remote objects but not to capture any 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 Rae et al. (2013a), 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 (Rae et al., 2013a; Kuzuoka et al., 2000) 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 (Choi and Kwak, 2017). Traditional telepresence work has typically resembled video conferencing through a telepresence robot, but more recent work is exploring the integration of extended reality (AR, VR, MR) with telepresence robots to create blended spaces that combine live video and virtual spaces for more immersive collaboration (Jones et al., 2021; Sakashita et al., 2023).

There is a rich literature (*e.g.*, Kratz and Rabelo Ferriera (2016); Kiselev et al. (2014); Rae et al. (2013b, 2015); Vartiainen et al. (2015); Johnson et al. (2015b); Stahl et al. (2018)) on how telepresence robots and interfaces should be designed to support communication between remote users. Typically, in this body of work, the robot's embodiment is strongly tied to the remote user and is designed to be their surrogate in the local workspace. For example, Tsui and Yanco (2013) recommend an adjustable height for the robot (or camera) to enable the remote user to interact at eye level with the local user, who may be seated or standing. Furthermore, Rae et al. (2013b) found that, similar to human-human communication, the robot's height can influence how much authority is exhibited by the local user.

Researchers have also explored other form factors for telepresence robots, such as drones (Sabet et al., 2021; Zhang et al., 2019) or tabletop robots (Adalgeirsson and Breazeal, 2010; Sakashita et al., 2022). While these systems are typically designed for interpersonal communication, some recent works (Villanueva et al., 2021; Li et al., 2022) have addressed the use of tabletop robots for supporting collaboration in remote physical tasks. Villanueva et al. (2021) 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. (2022) 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.

There is a growing area of research within HRI that studies robots functioning in groups. In their review of this work, Sebo et al. (2020) exclude the types of robotic systems discussed in this subsection and instead focus on scenarios where at least two locally present people are interacting simultaneously with an autonomous robot. The exclusion is based on the rationale that the robots in the systems discussed above primarily serve as tools rather than operating as autonomous agents. Nevertheless, this line of research can provide insights into how robots, when integrated into group settings, can influence human-to-human interactions.

2.3 Chapter Summary

Groupware can facilitate a spectrum of interdependent group work. Complex collaborative scenarios, such as remote expert guidance for physical tasks (as described in §1.1) demand sophisticated tools capable of supporting users' awareness needs in a shared environment. A critical aspect of this support is facilitating the sharing of visual context. Prior works have achieved this through various enabling technologies, including robots. These works provide valuable insights, including the observation that granting helpers control over the view enables them to have diverse and independent views of the workspace, and that collaborative systems should support referential communication.

However, many prior works focus on supporting interpersonal communication, which has simpler awareness needs, or they involve relatively uncluttered workspaces with short, structured tasks. These scenarios do not reflect the complex and dynamic real-world context of using groupware to support physical work. This gap highlights the potential for cobot platforms. In existing work, extended reality applications have emerged as some of the most promising solutions for scenarios similar to what we explore in this work. However, cobots, with their capability of leveraging their physical form to extend human ability in remote spaces, offer novel ways to facilitate collaborative efforts.

This dissertation contributes to the understanding of how and where a highly capable cobot platform can be constructively integrated into a collaborative system. The introduction of more sophisticated tools, including cobots, also introduces an increased complexity in user interactions. It is crucial to avoid overwhelming users with this complexity. Therefore, Chapter 3 focuses on the design of control mechanisms and policies for these cobots to ensure that they can be utilized effectively without overburdening users, while still providing them with the desired level of control. Chapter 4 provides insights into whether these control mechanisms successfully facilitate the sharing of visual context, thereby supporting the awareness needs of complex, interdependent group work. Finally, Chapter 5 characterizes real-world scenarios where cobot solutions may be particularly beneficial or face significant challenges.

3 DESIGN OF THE PERISCOPE SYSTEM

This chapter describes the design and development of the *Periscope* system. It addresses the central challenge of designing control mechanisms and policies to ensure that the cobot can be utilized effectively within a robotic camera system. In this chapter, I motivate this central challenge, review key prior work in CSCW and robotics that informed our design decisions, present design goals that guided our design process, provide details about our system (including the control mechanisms and policies that we developed), and reflect on the design process. This chapter includes research from previously published work in Praveena et al. (2023c) and Meng et al. (2023).

3.1 Motivation

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 (Gutwin and Greenberg, 2002; Kraut et al., 2003). Additionally, the requirements on these views may change over the course of the task (Fussell et al., 2003b). 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 at varying levels of detail, such as a close-up view to see fine details or a wide-angle view to see more context (Kuzuoka et al., 1994). 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 (Fussell et al., 2000).

The focus of recent research on remote collaboration in HCI and 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. (2022) 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 (Fussell et al., 2000, 2003a). 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 (Druta et al., 2021). Giving the helper direct control of a high-DoF robotic camera 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 (Mentis et al., 2020; Lanir et al., 2013). 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 (Rae et al., 2014). 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* (Losey et al., 2018)) 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 3.1).

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

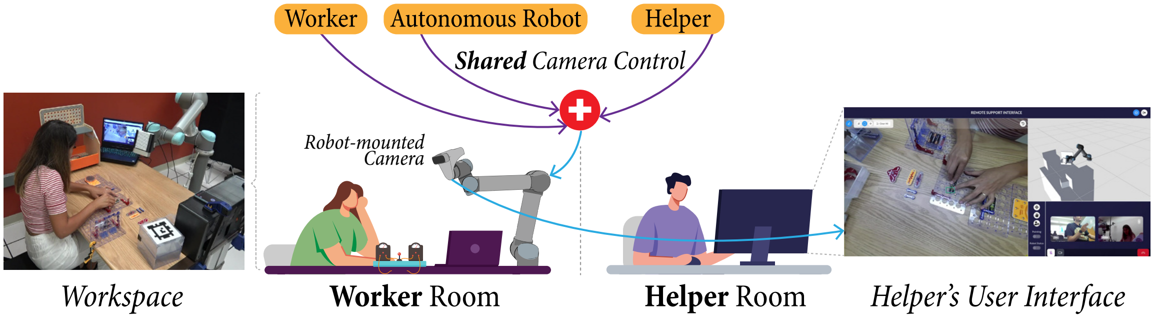


Figure 3.1: The design of the *Periscope* system incorporates a *shared camera control* approach in which the worker, the helper, and the autonomous robot all contribute to camera control depending on task needs. Through shared camera control, we tackle the challenge of simplifying control of a high-DoF robotic camera and providing users with diverse, informative, and task-relevant views.

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 control of a high-DoF robotic camera and providing users with diverse, informative, and task-relevant views.* The work in this chapter makes key contributions in two categories:

1. *Design* — the shared camera control approach and a set of design goals to realize this approach.

2. *System* — *Periscope*, an operational robotic camera system that is an instantiation of shared camera control with user interactions, autonomous behaviors, and arbitration policies designed for the context described in Section 1.1.

3.2 Background

The background information presented in Chapter 2 provides high-level guidelines for designing groupware systems. However, a key challenge we face is developing specific low-level control mechanisms and policies tailored for a robotic camera system that address the unique awareness needs of our research context. Consequently, this has guided us towards adopting a shared camera control approach. In this section, I discuss relevant background on *awareness* and *control*.

Awareness

The CSCW literature features extensive research on how technologies can facilitate *awareness* among collaborating individuals. This term, however, is multifaceted and sometimes employed in inconsistent ways. Schmidt (2002) discussed the intricacies of this concept in his remarks on "The Problem with Awareness." He noted that the term awareness has significance only when it denotes a person's awareness of something. Thus, in this section, I focus on the framework of *workspace awareness*, as described by Gutwin and Greenberg (2002). Workspace awareness is a specialization of the concept of situation awareness by Endsley (1995) and is tailored to real-time collaborative workspaces. In this context, the question of *"awareness of what?"* raised by Schmidt (2002) refers to the awareness of others' interactions in a shared environment. This framework provides a foundational understanding of the awareness needs that the robotic camera system should support.

Awareness Needs: Gutwin and Greenberg (2002) provide a set of basic elements as a starting point for thinking about awareness needs — elements that answer "who, what, where, when, and how" questions. In a remote dyad setup, understanding "who" is straightforward, but other aspects are critical for highly interdependent work such as the remote expert scenario.

- Awareness of *Actions*, *Intentions*: Understanding what a collaborator is doing.
- Awareness of *Artifacts*: Identifying objects a collaborator is engaging with.

- Awareness of *Location*, *Gaze*, *View*: Knowing where a collaborator is located, where they are looking, and what they can see.
- Awareness of *Reach*: Recognizing the area within the workspace where a collaborator can effect change.
- Awareness of the *Past*: Understanding how an action was performed or how an artifact reached its current state, including the timing of events, as well as the historical context of all the previously mentioned awareness types.

These are some of the elements that make up workspace awareness knowledge. In co-located group work, participants usually know these elements, either consciously or subconsciously. However, in designing remote, real-time groupware, decisions must be made regarding whether and how to support these elements (Gutwin and Greenberg, 2002).

Maintaining Awareness: According to the framework, awareness of an environment is maintained through the perception-action cycle (Neisser, 1976). When someone enters an environment for a specific task, they come with some basic understanding and expectations. As they gather information from the environment, they interpret new information using their existing knowledge to understand the current state and predict future changes. Since environments evolve, maintaining up-to-date awareness is crucial. This updating of awareness is achieved through exploration of the environment (Adams et al., 1995; Gaver, 1992). Gutwin and Greenberg (2002) noted that maintaining awareness is not the primary goal but a secondary one to aid in the completion of the collaborative task.

Control

After identifying the relevant sources of information for awareness, the next steps for groupware design involve understanding how to acquire this information, when and where it is required, and how to display it. In the unstructured and highly interdependent collaboration context we are focusing on, this information tends to be tacit, dynamic, and predicated on sequences of prior actions. This complexity makes it challenging to automate the support for awareness. Consequently, there is a need for flexible and intuitive control mechanisms and policies to empower users to acquire diverse, timely, and up-to-date information. Rodden and Blair (1991) distinguished a control mechanism and policy as *"the ability to move an object* (*the mechanism*) *and the decisions about when the object should be moved and to which site* (*the policy*)*."* For our context, the *object* can be considered as the view/camera/robot. This section focuses on the relevant background camera control methods and control mechanisms for robots when humans and robots work in a collaborative ecosystem that inform our design.

Camera Control: Christie et al. (2008) 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 (Chaumette et al., 2016; Hutchinson et al., 1996), through-the-lens camera control (Gleicher and Witkin, 1992), assisted camera control in virtual environments (Christie et al., 2008), and automatic cinematography (Christianson et al., 1996) to make it easier for the helper and the worker to influence the shared view.

Visual servoing (Hutchinson et al., 1996) 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 (Gleicher and Witkin, 1992) 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 (Christie et al., 2008) 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 (Christianson et al., 1996) 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 groupware, for instance, hand tracking in Ranjan et al. (2007), and commercial video conferencing products such as Apple Center Stage[1] and Lumens Auto Tracking Camera[2].

Teleoperation: There is a substantial body of research on teleoperation control interfaces (see Rea and Seo (2022) for a review), where awareness and control are critical aspects being studied, similar to research in groupware. These two aspects impose a significant cognitive load on teleoperators (Chen et al., 2007). This is particularly relevant in our context, where the remote user essentially teleoperates the robot. In this review, my focus will be solely on the control-related aspects from this body of work because awareness needs in a collaborative scenario differ from those in a teleoperation scenario. Regarding control, the level of autonomy of the operator's input is a key consideration.

Existing teleoperation interfaces span a range of levels of autonomy, where human input is combined with different levels of autonomous robot behaviors (Beer et al., 2014). Direct control can allow operators to execute their intentions with more precision and respond quickly to uncertainty, but it can become tedious because of the complexity inherent in controlling a high-DoF robot. Despite some advancements in direct control (*e.g.,* Rakita et al. (2017); Wang et al. (2023)), research has increasingly shifted towards incorporating more semi-autonomous behaviors. This shift facilitates the use of low-dimensional input devices for controlling high-dimensional robots (*e.g.,* Herlant et al. (2016); Karamcheti et al. (2021)), controlling robots at the action level rather than through direct motion control (*e.g.,* Schmaus et al. (2019); Senft et al. (2021b)), and employing flexible controls and intermittent input (*e.g.,* Bohren et al. (2013); Hagenow et al. (2021)).

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

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

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 (Dragan and Srinivasa, 2013; Losey et al., 2018). This paradigm is also referred to as collaborative control (Macharet and Florencio, 2012) or mixed-initiative human-robot interaction (Jiang and Arkin, 2015). 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. (2018) 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 (Abi-Farraj et al., 2016; Nicolis et al., 2018; Rakita et al., 2019; Senft et al., 2022) 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 by Rakita et al. (2017, 2018, 2019, 2021) with adaptations for remote human collaboration where the robot augmented with a camera is co-located with a worker completing manual tasks.

3.3 Design Goals

A key takeaway from the background presented in Chapter 2 and Section 3.2 is the concept of roles and contributions, which is present in both the CSCW and shared control literature. Therefore, adopting a shared control approach in designing control policies closely aligns with CSCW theories related to collaborative work. In robotics research, shared control typically involves one human and one robot. However, in our context, the design of shared control needs to accommodate both human collaborators.

In our design process, we must carefully consider balancing user control and effort. In his paper on remote collaboration systems, Gaver (1992) asserts that unless the cost of gaining additional information (*e.g.,* through camera control) is low enough, it will not seem worth the additional effort for users. Our work is guided by this idea, aiming to empower both the helper and the worker to move the camera with low cognitive and physical costs. To achieve this, various control mechanisms are possible and we prioritize those we believe will be most effective in our research context. Our control mechanisms are described in Section 3.5 and our control (or arbitration) policy is described in Section 3.6.

To guide our design process, we identified five high-level design goals based on prior literature and early feasibility studies. 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.

Versatility

Support camera views for various task activities
The visual information necessary for users 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, 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.

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).

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).

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.

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.

3.4 System Overview

We developed the *Periscope* system based on the design goals stated in Section 3.3. As shown in Figure 3.2, 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.

I include technical details of the implementation in Appendix A, and present high-level descriptions of the system in Sections 3.4–3.6.

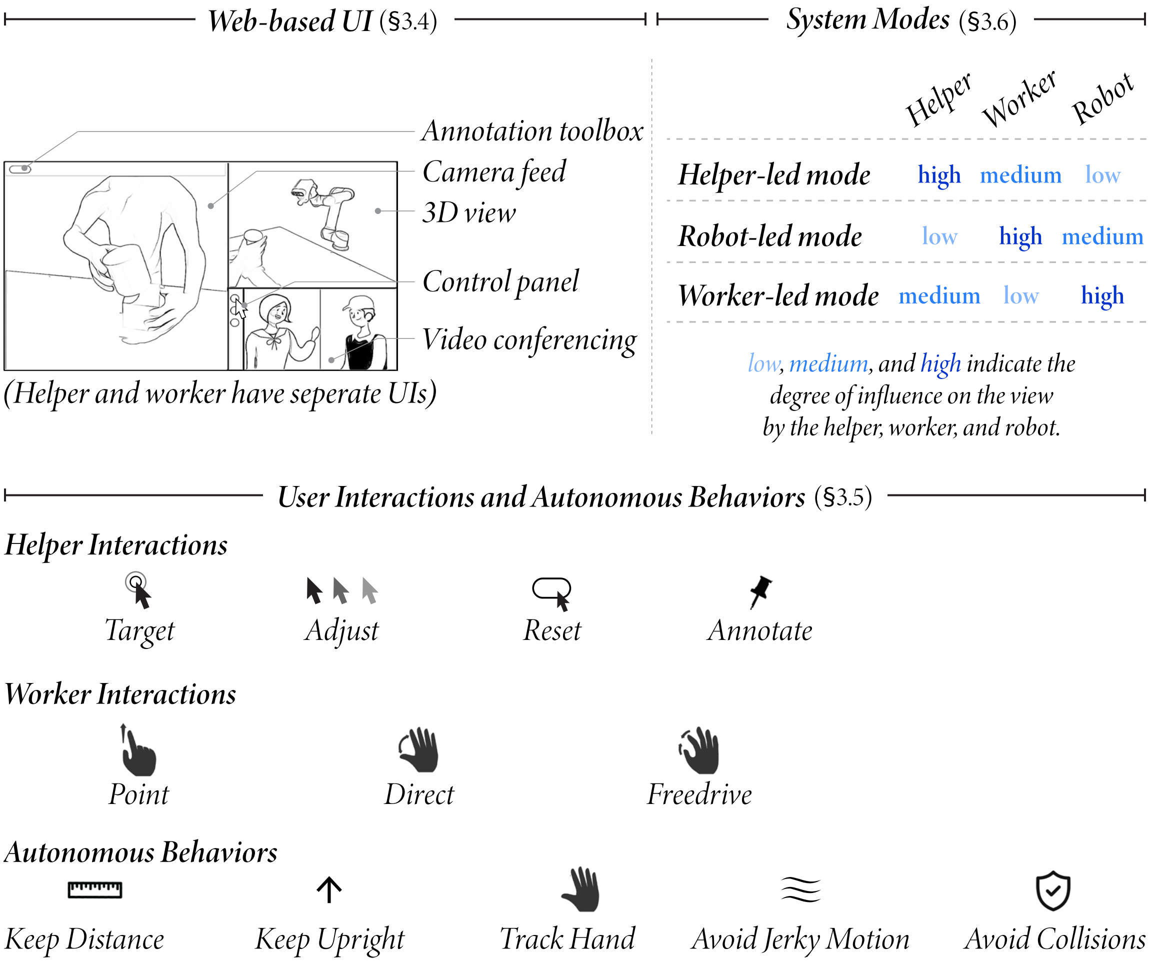
User Interfaces

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 a wide and continuous set of positions and orientations within a 3D workspace. The helper is in a remote 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

[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.

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.



	Helper	Worker	Robot
Helper-led mode	high	medium	low
Robot-led mode	low	high	medium
Worker-led mode	medium	low	high

low, medium, and high indicate the degree of influence on the view by the helper, worker, and robot.

Figure 3.2: The *Periscope* system consists of three components: (*Top-left*) user interfaces for the helper and the worker, (*Bottom*) a set of helper interactions, worker interactions, and autonomous robot behaviors to support establishing a shared visual context, and (*Top-right*) system modes that arbitrate user interactions and autonomous behaviors in real-time, resulting in camera motion.

3.5 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, I describe helper and worker interactions, and autonomous behaviors afforded by the *Periscope* system.

Helper Interactions

Helpers use the screen interface to interact with the system via mouse input commands on the camera feed or the control panel (see Figure 3.3 for illustrations).

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 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 a specific direction based on directional inputs, 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 behaviors depending on whether *Target* was engaged prior to *Adjust*. If the camera is pointing at a target, then it will perform orbital rotations 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 graphics on the camera feed for referential communication with the worker. The helper can drop a pin to indicate a point, draw a rectangle to indicate an object or an area, 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.

Worker Interactions

Workers move the camera by engaging directly with the robot arm using physical contact and gestures recognized by the camera (see Figure 3.3 for illustrations). 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 the target 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.

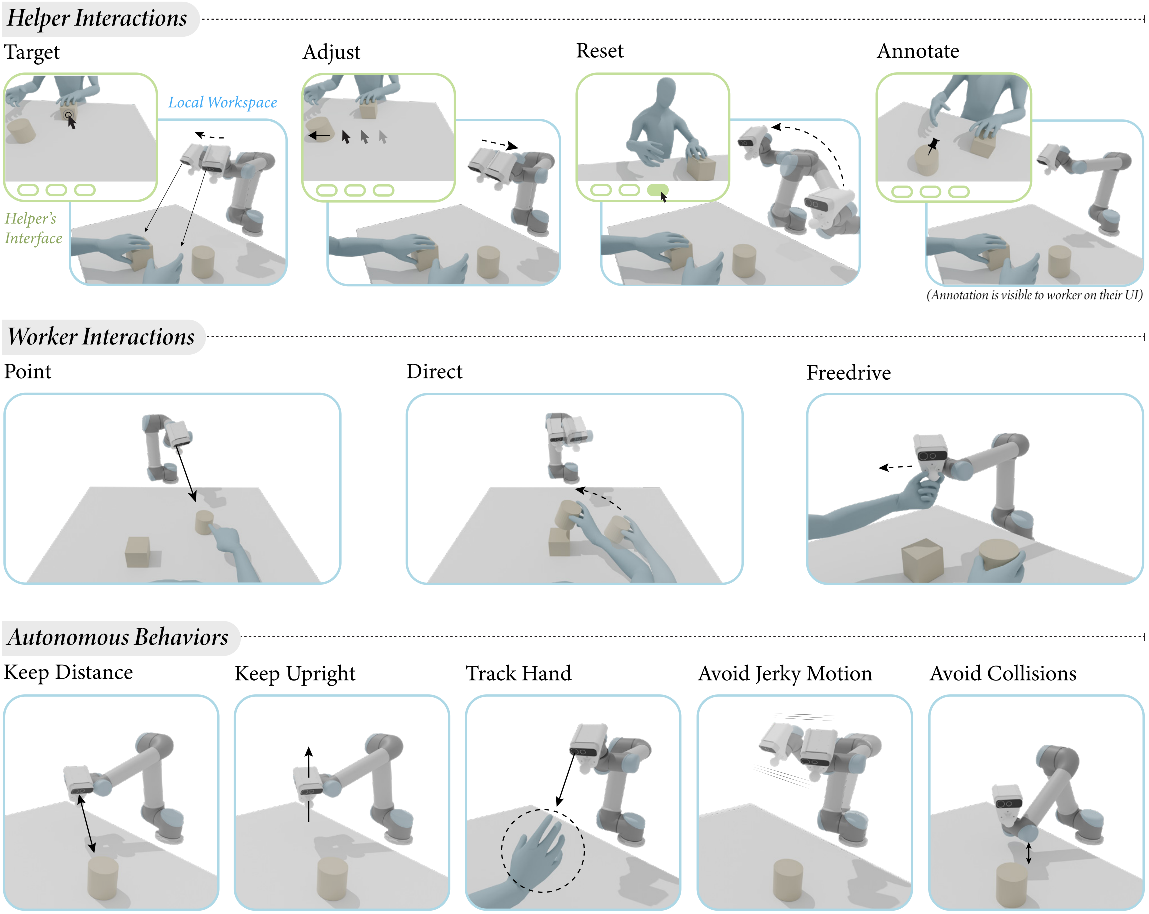


Figure 3.3: The *Periscope* system supports a variety of interactions for the helper and the worker, assisted by autonomous robot behaviors. Images with a blue border represent the local workspace, where both the robot and the worker are present. Images in the top row with a green border depict the remote helper's UI which enables the helper to interact with the system through mouse input on the camera feed or the control panel. Worker interactions leverage the worker's proximity to the robot, allowing them to move the camera directly using gestures and physical contact with the robot arm. Autonomous behaviors focus on geometric (rather than semantic) qualities of the view, which are challenging for humans but feasible for robots to achieve.

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 guide 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 responds to applied forces, moving in the direction of push or pull from the worker.

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 (see Figure 3.3 for illustrations).

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 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 its 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.

3.6 Arbitration

We developed system modes that arbitrate the user interactions and the autonomous behaviors described in Section 3.5 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.

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.

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.

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.

3.7 Discussion

This discussion centers on the development process of the *Periscope* system, provides a reflection on the system's functionality, and characterizes the type of systems contribution made by *Periscope*.

Development Process

The design and development of the *Periscope* system was driven by the objective to create a Minimum Viable Product (MVP) that was operational, usable, and useful. Ries (2011) defined an MVP as *"that version of a new product which allows a team to collect the maximum amount of validated learning about customers with the least effort."* This approach emphasizes the balance between realizing as much functionality as needed and maintaining simplicity (Schloesser et al., 2017). My approach to achieving the MVP involved a highly-iterative design process inspired from the *design sprint* methodology (Banfield et al., 2015; Knapp et al., 2016).

This process involved five phases: understand, diverge, converge, build, and test. In the *understanding* phase, we defined our research context and identified needs by reviewing existing methods and systems in research literature. During the *diverging* phase, we developed various low-fidelity prototypes of a new feature to explore multiple solutions. During the *converging* phase, we selected the most promising solution based on rapid testing. The *building* phase saw the development of a higher-fidelity prototype, and the testing phase focused on integrating and validating this new feature within the existing system. We cycled through these stages multiple times to develop an MVP. With each added feature, we carefully considered its potential to enable users to effectively accomplish tasks and tailored it to the intended context in which the system would be used.

This early prototyping was not bound by a strict plan but was flexible to address specific emerging questions (Schloesser et al., 2017). Upon establishing an MVP, we shifted our focus to refining the entire system. This refinement aimed to achieve what Hokkanen et al. (2016) described as the Minimum Viable User Experience (MVUX). My goal was to provide a satisfactory user experience that effectively communicates the envisioned value of this paradigm of using cobots in order to gather meaningful feedback and generate interest to support further research (*e.g.*, securing personnel and funding).

Systems Contribution

Appendix A provides implementation details for the *Periscope* system. This content was intentionally not included in the main chapter, as it primarily involves engineering aspects and the adaptation of existing work. This raises the question: Does the *Periscope* system make a significant systems contribution (apart from the design contribution), considering it largely adapts known techniques?

In response, based on UIST guidelines[4], I argue that *Periscope* achieves novel functionality leveraging known techniques. Although individual components of the system, such as shared control, robot motion generation, and web-based teleoperation interfaces, are well-studied in robotics, the system as a whole transcends these individual elements. Holistically, *Periscope* introduces novel functionality that pushes the boundaries of current work paradigms. *Periscope* is a synergistic integration of various methods and techniques from HCI, CSCW, HRI, and robotics that demonstrates the feasibility of building an operational groupware system using cobots and thereby unlocks new applications.

[4] `https://medium.com/acm-uist/a-note-from-the-uist-2021-pc-chairs-6a30df14f33b`

3.8 Chapter Summary

This chapter presents the design and development of an operational robotic camera system called the *Periscope* system, which facilitates remote collaboration on physical tasks. A key aspect of the design of the *Periscope* system is a shared camera control approach where the control of the camera is dynamically distributed between the local user, the remote user, and the robot depending on task needs. *Periscope* is an instantiation of shared camera control with user interactions, autonomous behaviors, and arbitration policies designed for the research context described in Section 1.1. We propose that: (1) this approach enables us to leverage the advanced capabilities of cobot platforms without overwhelming users with the complexity of the tool, and (2) it allows users to maintain the desired level of control expected from a collaborative tool. The user study described in the subsequent chapter explores this proposition.

4 USAGE OF THE PERISCOPE SYSTEM

This chapter presents our evaluation of the *Periscope* system with 12 dyads in a lab study to understand the promise of our shared camera control approach for facilitating remote collaboration. In this chapter, I provide the motivation for this work and our approach, present our evaluation methodology, describe the user study design, present empirical observations on system usage, and discuss the system's ability to support the design goals introduced in Section 3.3. This chapter includes research that was previously published in Praveena et al. (2023c).

4.1 Motivation

In Chapter 2, I discussed that systems that support remote collaboration facilitate the sharing of visual context to enable effective cooperation and communication between users. This shared visual context is used for maintaining awareness and conversational grounding. An example from Kraut et al. (2003) illustrates this. In this example, a helper guides a worker in adjusting the inclination of a bicycle seat during a repair task.

> ***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.*

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 (Levinson,

2006) (*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. This rich interaction between the helper and the worker demonstrates the efficacy of the mediating system. Our goal for the evaluation in this chapter is to design a user study that allows us to observe such interactions in order to gain insights into the usability of the *Periscope* system.

The evaluation of CSCW systems presents unique challenges due to the complex nature of group interactions (Grudin, 1994). The success of collaboration outcomes is influenced not only by the effectiveness of the groupware system in facilitating collaboration but also by various group and organizational dynamics. For example, factors such as the similarity among group members, familiarity with one another, and the presence of hierarchical structures within the group can exert a substantial impact on collaborative outcomes (Harris et al., 2019). Further, these social dynamics may evolve over time (Twidale et al., 1994) and may be reshaped by the use of the groupware (Neale et al., 2004).

In groupware evaluations, relying solely on performance measures (*e.g.,* task completion rate or completion time) may be insufficient or even misleading, and quantitative metrics to capture the nuances of the group dynamics described earlier have been elusive (Neale et al., 2004). Thus, evaluation methodologies in CSCW are continuously evolving to address these challenges (Pinelle and Gutwin, 2000; Wallace et al., 2017). In this work, we adopt an evaluation approach that incorporates the concepts of *formative-qualitative-opportunistic* evaluation (Twidale et al., 1994), *experimental simulations* (Neale et al., 2004), and the trade-off between precision, generalizability, and realism (McGrath, 1984). I discuss this further in Section 4.2. The work in this chapter makes key contributions in two categories:

1. *Method* — an experimental paradigm to study the usability of a system while preserving key elements of a natural collaboration scenario.

2. *Data* — empirical observations on system use and their contribution to the design goals described in Section 3.3.

4.2 Research Methodology

Twidale et al. (1994) suggested that during the initial phases of evaluating novel groupware systems, adopting a *formative-qualitative-opportunistic* style of evaluation is valuable. This is in contrast to a *summative-quantitative-controlled* evaluation style, which may be better suited for mature systems. In the following paragraphs, I will elucidate the terms used in the two evaluation styles.

Hix and Hartson (1993) distinguish between *formative* and *summative* evaluation, highlighting that formative evaluations focus on assessing the *interaction design*—what the user sees, hears, and does while engaging with the system—rather than the system itself, which is the focus in summative evaluations. They further emphasize that formative evaluation should not be dismissed as informal or lacking scientific rigor; formative evaluation has an *"explicit and well-defined procedure and does result in quantitative data, but is not intended to provide statistical significance."* In the context of the *Periscope* system, formative evaluation is particularly valuable for gaining a broad understanding of the paradigm, rather than assessing specific features, graphics, or devices within the system. Our findings are rooted in *qualitative* data, such as conversations between collaborators and observed actions, enabling us to explain system features in the context of their use and provide descriptions of user processes to aid in the design of future robotic groupware. While some *quantitative* results are reported, they are not intended to be statistically significant.

The *opportunistic–controlled* scale addresses the level of control over the user's interactions during a study. Opportunistic evaluations allow open-ended and natural exploration by users, whereas controlled evaluations involve a structured and regulated environment to isolate specific variables for analysis. For evaluating novel groupware systems like *Periscope*, opportunistic evaluations are valuable for exploring design opportunities, demonstrating successful tool use, and identifying specific variables that could be further studied in controlled settings in the future. An intermediate approach on this scale involves *experimental simulations*, as described by Neale et al. (2004). Experimental simulations are designed to stage situations in the laboratory that resemble operational contexts. Simulations enable repeated observations while maintaining a more naturalistic setting, allow the study of complex tasks over longer time periods than in controlled experiments, and allow the use *probes* or deliberate interventions to elicit specific user responses. This approach strikes a balance — providing a controlled environment while including the context of use of the system being evaluated.

In his review on research methods for understanding groups, such as field studies, laboratory experiments, surveys, and formal theories, McGrath (1984) underscores the challenge of simultaneously maximizing *precision* (control over measurements and extraneous factors), *generalizability* (applicability of findings to a wider population), and *realism* (faithfulness to the intended contextual setting). Experimental simulations offer some balance between precision and realism, albeit with low generalizability (Runkel and McGrath, 1972; McGrath, 1984). In our preliminary evaluation of the *Periscope* system, we try to maintain a balance between realism and precision. We aim to increase the generalizability of our findings with another user study discussed in Chapter 5.

To mimic real-world settings in this chapter's study, we incorporate a complex 3D physical task within a cluttered workspace, structure collaboration sessions to last 45–60 minutes, and do not dictate the use of specific system features or provide explicit steps for task completion. To introduce an element of control, participants undergo a rigorous training protocol to ensure a uniform baseline knowledge about the system, receive a rough task outline to ensure some meaningful points of comparison between participants, and encounter impediments in instructions such as blurring or removing color to encourage more real-world visual exploration.

Our study design described in Section 4.3 draws inspiration from the principles of experimental simulations and aims to maintain key elements of a natural collaboration scenario within a laboratory setting. Our empirical findings presented in Section 4.4 and the subsequent discussion in Section 4.5 are derived from a rigorous analysis of qualitative data gathered during our formative evaluation of users' opportunistic (rather than mandated) use of the *Periscope* system.

4.3 User Study

We recruited dyads to participate in our 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 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, I describe the final protocol that we developed and clarify the variations of the protocol followed by each dyad in §Participants.

Tasks

The training task was to construct a pulley system from a toy workbench kit[1] (see Figure 4.1C). 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.

[1] Workbench Kit: `https://a.co/d/2zLeQoV`

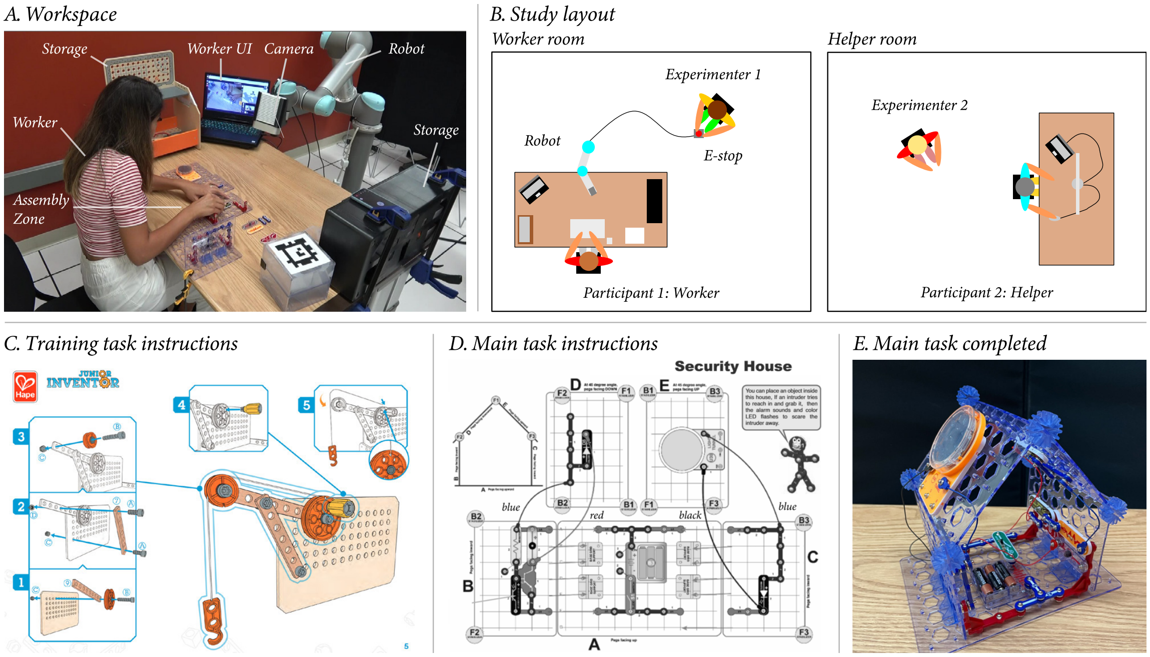


Figure 4.1: ***A.*** One participant (the worker) was located in the same physical space as a robot arm with access to the workspace but no instructions on how to carry out the assembly. The other participant (the helper) was tasked with guiding the worker remotely using the *Periscope* interface. ***B.*** 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.

The main task was to build a 3D illumination circuit project[2] (see Figure 4.1E). The helper was provided with a black and white copy of the instruction manual that came with the kit (see Figure 4.1D). 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

[2] SnapCircuit Kit: `https://a.co/d/34trhAd`

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.

Study Setup

The study took place in two rooms: the worker room and the helper room (see Figure 4.1B). 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 4.1A). 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.

Procedure

This protocol was approved by the Institutional Review Board of the University of Wisconsin–Madison. We conducted the study in two rooms in a university laboratory. Each study session lasted approximately two hours and was facilitated by two experimenters (I was Experimenter 1). 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 refer to a document that 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 4.1C) 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. Finally, participants completed a demographics survey, engaged in a brief interview about their experience, and received compensation for their time.

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 the University of Wisconsin–Madison's campus community. Demographic information for one dyad was not collected. The remaining participants (8 women, 14 men) 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 individuals in one dyad 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 §Procedure, 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 §Procedure. 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.

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 (Braun and Clarke, 2012). The first author[3] 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[4] (Crasborn and Sloetjes, 2008) was used for video coding, and the collaborative whiteboard app Miro[5] was used to refine thematic findings.

4.4 Findings

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. This section provides diverse examples of interactions facilitated by the *Periscope* system (see *§Example Interactions*), patterns of use of the system's features derived from nuanced interpretations of these examples and other interaction data (see *§Use Patterns*), and quantitative data for each dyad regarding their system use (see *§Dyadic Usage Metrics*). In Section 4.5, we elaborate on the significance of these findings to our design goals.

Note: In this section, dyads are enumerated as *D1, D2, D3, D4, D5, D6, D7, D8*.

[3] This is refering to published work (Praveena et al., 2023c): Pragathi Praveena, Yeping Wang, Emmanuel Senft, Michael Gleicher, and Bilge Mutlu. 2023. Periscope: A Robotic Camera System to Support Remote Physical Collaboration. Proc. ACM Hum.-Comput. Interact. 7, CSCW2, Article 350 (October 2023), 39 pages. https://doi.org/10.1145/3610199

[4] `https://archive.mpi.nl/tla/elan`
Max Planck Institute for Psycholinguistics, The Language Archive, Nijmegen, The Netherlands

[5] `https://miro.com/`

Example Interactions

This subsection presents various examples of interactions facilitated by the *Periscope* system. It features one long interaction and 23 short interactions. These interactions are later referenced in §Use Patterns and §Discussion.

Example 4.1 (D4)**.** Figure 4.2 shows an interaction spanning 10:16 minutes. During this interaction, the dyad locates required components and builds circuitry on the vacant wall grid. This interaction includes the following seven micro-interactions. If a micro-interaction was coded as a specific use pattern, the corresponding pattern is denoted in parentheses.

1. As the worker attaches the base support, the helper observes the worker's actions by independently moving the camera using the *helper-led mode* (UP-1).

2. The worker seeks confirmation if the attached component looks right, and the helper provides corrections by annotating the video feed (UP-2, UP-20). The annotation is accompanied by the use of deixis (*this*).

3. The dyad looks for another component (*phototransistor*) while the robot tracks the worker's hand in the *robot-led mode* (UP-5). Upon confirmation from the helper about the component (*that black piece*), the worker picks it up.

4. As the worker attaches the phototransistor, the helper attempts to verify the accuracy of the component's position using the *helper-led mode* (UP-1). The helper is unable to obtain a good view of the wall grid by the time the worker completes the attachment and subsequently seeks verbal clarification regarding the position of the component. The worker provides a clarification.

5. In addition to the clarification, the worker offers to move the camera using the *worker-led mode* to improve the helper's view and collectively resolve any ambiguity about the position of the phototransistor (UP-9). As the worker moves the camera, the helper acknowledges when the view is satisfactory.

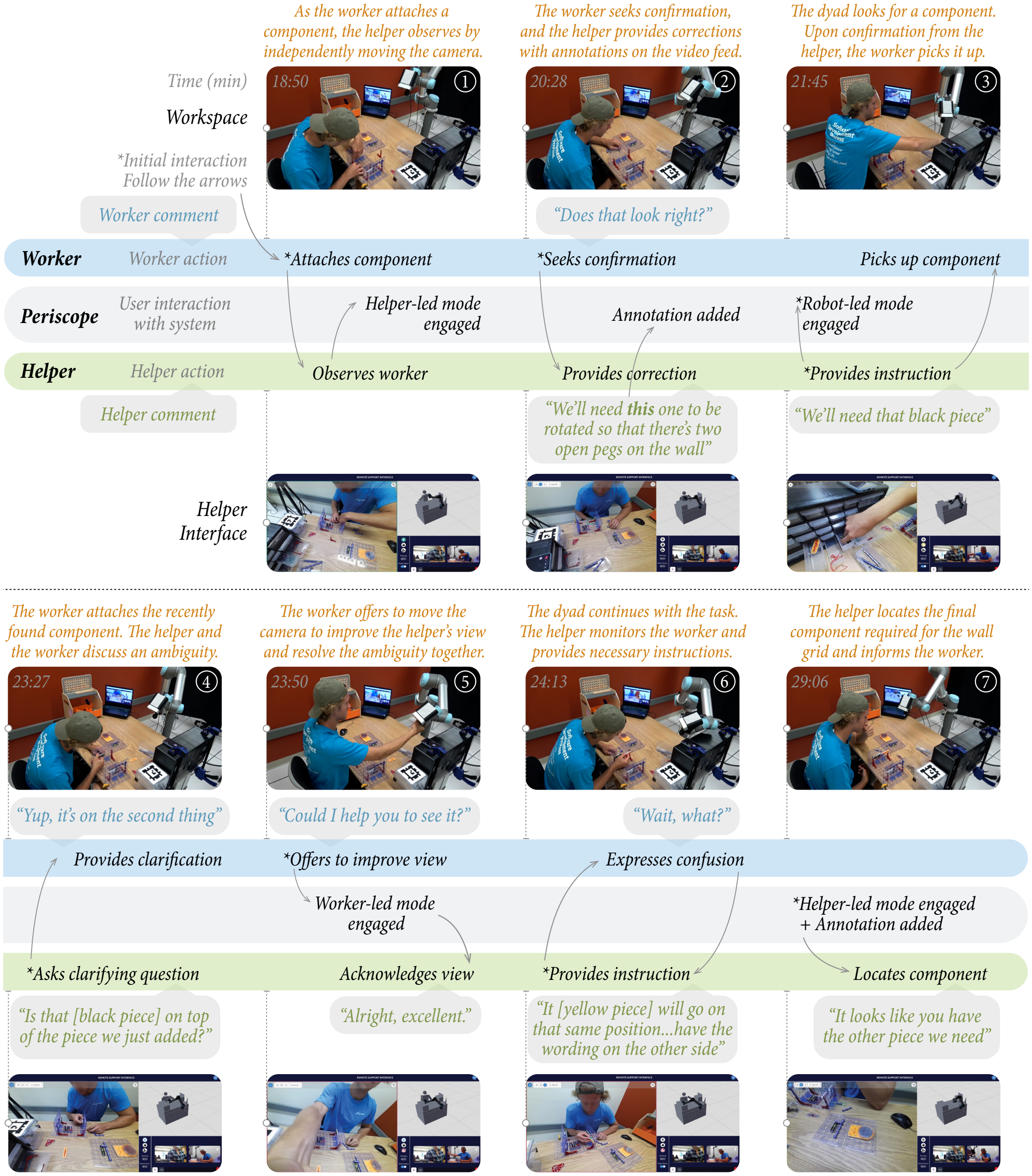

As the worker attaches a component, the helper observes by independently moving the camera.
The worker seeks confirmation, and the helper provides corrections with annotations on the video feed.
The dyad looks for a component. Upon confirmation from the helper, the worker picks it up.
Time (min)
Workspace
*Initial interaction
Follow the arrows
18:50
1
20:28
2
21:45
3
Worker comment
"Does that look right?"
Worker
Worker action
*Attaches component
*Seeks confirmation
Picks up component
Periscope
User interaction with system
Helper-led mode engaged
Annotation added
*Robot-led mode engaged
Helper
Helper action
Observes worker
Provides correction
*Provides instruction
Helper comment
"We'll need **this** one to be rotated so that there's two open pegs on the wall"
"We'll need that black piece"
Helper Interface
The worker attaches the recently found component. The helper and the worker discuss an ambiguity.
The worker offers to move the camera to improve the helper's view and resolve the ambiguity together.
The dyad continues with the task. The helper monitors the worker and provides necessary instructions.
The helper locates the final component required for the wall grid and informs the worker.
23:27
4
23:50
5
24:13
6
29:06
7
"Yup, it's on the second thing"
"Could I help you to see it?"
"Wait, what?"
Provides clarification
*Offers to improve view
Expresses confusion
Worker-led mode engaged
*Helper-led mode engaged + Annotation added
*Asks clarifying question
Acknowledges view
*Provides instruction
Locates component
"Is that [black piece] on top of the piece we just added?"
"Alright, excellent."
"It [yellow piece] will go on that same position...have the wording on the other side"
"It looks like you have the other piece we need"

Figure 4.2: This example interaction from dyad *D4* includes seven micro-interactions spanning 10:16 minutes, with three annotated tracks for worker actions (*"Worker"*), helper actions (*"Helper"*), and their interactions with the *Periscope* system (*"Periscope"*). Each micro-interaction has conversation snippets accompanying the *"Helper"* and *"Worker"* tracks, and images illustrating the worker's space and the helper's interface. Each micro-interaction starts with (*) and subsequent actions/interactions are indicated by arrows. Time and sequence numbers are annotated on the workspace image.

6. The dyad continues with the task. The helper monitors the worker using the view from the previous micro-interaction. The helper provides necessary instructions and uses visual information to clarify any confusion in the worker's understanding of the instructions.

7. The helper locates the final component (*blue connector*) required for the wall grid using the *helper-led mode* (UP-3) and indicates its location to the worker by annotating the feed (UP-20).

Example 4.2 (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 4.3 (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 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 4.4 (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 4.5 (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 for a few seconds, 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 4.6 (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 independently moved their hand to a different location in the workspace, changing the view unexpectedly for the helper. Frustrated, 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 4.7 (D8)**.** When trying to view the vertical 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 4.8 (D3)**.** During the search for a component, the helper instructed 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...Maybe take that out and show me?...[worker brings the component closer to the camera]...Yeah, that's what we're looking for."*

Example 4.9 (D4)**.** The worker remarked, *"Oh, here it is,"* before picking up and presenting to the helper a storage box that the dyad was looking for. The helper engaged the *robot-led mode* in response to the worker's remark to track the worker's hand movement.

Example 4.10 (D6)**.** While the helper was adjusting the camera view, the worker examined the circuit and noticed a potential error. 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 4.11 (D5). As the helper explained the next step with the instruction, *"You can connect that to the second one"*, the worker proactively changed the view to the assembly area, providing a clear view of the connections.

Example 4.12 (D2). Upon noticing the helper's difficulty in adjusting the view using the *helper-led mode*, the worker offered their assistance, saying, *"You can turn on mode 3 [worker-led mode], and I'll help you adjust the camera."* The helper turned down the offer, stating, *"Um...I think I can adjust the camera myself."*

Example 4.13 (D7). The helper attempted to move the camera to inspect the roof panel but faced challenges and eventually gave up, admitting, *"Actually, I am not able to see the top panel...can you?...I need to look up to the panel."* In response, the worker engaged the *worker-led mode* to move the camera and show the roof panel.

Example 4.14 (D4). The helper briefly attempted to use the *helper-led mode* to view the wall grid, a view that the helper had achieved previously after considerable effort. Following the brief attempt, 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 4.15 (D5). During the assembly process, the helper consistently relied on the worker to move the camera while looking at different aspects in the workspace, such as the organizer, components, and assembly area. When they were ready to progress to the next step in the process, the helper requested, *"The parts that I had you collect from the organizer...can you show me that?"*

Example 4.16 (D6). To access the instructions located on the side of the workbench, the helper asked the worker, *"Could you just guide me towards the side of the workbench?"*

Example 4.17 (D3). 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?"* The helper seemed to anticipate that the camera would align with the direction of the worker's pointing gesture.

Example 4.18 (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 4.19 (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 4.20 (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 4.21 (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 4.22 (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 4.23 (D8)**.** The helper initiated the *worker-led mode* by asking the worker, *"Can you show me the board [wall grid] 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.

Example 4.24 (D1)**.** The helper engaged the *robot-led mode* and stated, *"Okay, we are in mode 2 [robot-led mode] now. Did the robot detect you?"* The worker replied, *"No. You might have to reset it."* The helper proceeded to reset the robot and engaged the *robot-led mode* again. There were no further issues with the robot tracking the worker's hand.

Use Patterns

Table 4.1 summarizes a list of 20 use patterns for modes and other features of the system. The use patterns are nuanced interpretations of the rich multi-modal data that we analyzed, examples of which are provided in *§Example Interactions*. We cite references to them in this subsection to offer the context of the rich interactions from which they were interpreted. In the rest of the subsection, we provide a detailed breakdown of use patterns.

Table 4.1: 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 §Use Patterns to details about each pattern.

Feature	#	**Use Pattern**
Helper-led mode	UP-1	The helper gains awareness of the workspace.
	UP-2	The helper provides the worker with task instructions.
	UP-3	The helper searches for something.
	UP-4	The helper attempts to move the camera before asking the worker do it instead.
Robot-led mode	UP-5	The dyad gathers components for the build.
	UP-6	The helper tracks the worker's movement.
Worker-led mode	UP-7	The worker wants to share some information with the helper.
	UP-8	The worker anticipates the helper's need for a different view.
	UP-9	The worker offers to move the camera on behalf of the helper.
	UP-10	The helper attempts and fails to move the camera on their own.
	UP-11	The helper is already aware from an earlier attempt that a particular view is difficult to achieve.
	UP-12	The helper requests repositioning the camera that the worker had previously set up.
	UP-13	The helper does not know where to position the camera.
Point	UP-14	The helper asks the worker for a specific view.
	UP-15	The worker refers to something in the workspace.
Reset	UP-16	The reset pose serves as a bookmarked pose that provides a sufficient view of the workspace with minimal effort.
	UP-17	The reset pose serves as an intermediate pose when transitioning from one sub-task to the next.
	UP-18	The reset pose is a comfortable starting configuration for the *helper-led mode*.
	UP-19	The system does not respond as expected.
Annotate	UP-20	The helper refers to something in the workspace.

Note: Dyad references and counts of use patterns are included in parentheses.

Helper-led Mode Use Patterns

We observed that this mode was used 82 times (excluding its use when the worker used the pointing gesture which we discuss separately in *§Point*). The average duration of each use was 38 seconds ($SD = 41$ seconds). 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 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 (E4.2). 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"* (E4.3). 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*

(UP-5), 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 (E4.6). 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 E4.7). We will revisit this reason in UP-10 when discussing the use of the *worker-led mode*.

Robot-led Mode Use Patterns

This mode was used 20 times (excluding its use when the worker used the pointing gesture which we discuss separately in *§Point*). 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 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 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 E4.8). 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 E4.9). In these instances, the worker did not explicitly direct the camera.

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*).

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 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 E4.10), 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 E4.11), 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 E4.12).

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 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 reasons for helpers initiating the *worker-led mode*:

1. *The helper attempts and fails to move the camera on their own (13/36):* If the helper did not get the desired view using the *helper-led mode*, they asked the worker to move the camera. For example in *D7* (E4.13), 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."*

2. *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 E4.14) 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*).

3. *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 E4.15).

4. *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 E4.16).

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 and canceled out each other's inputs. In addition, 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 phrase was misunderstood by the worker as a request to engage the *worker-led mode* to move the camera, resulting in overriding the helper's mode selection.

Other Use Patterns

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 E4.17).

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 E4.18).

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 E4.19).

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 E4.20).

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 E4.21 and E4.22).

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 E4.23).

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 E4.24). In response, helpers used the *Reset* feature as a way to restore the system to a functional state.

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.

Dyadic Usage Metrics

Our quantitative data for each dyad regarding their system use is summarized in Table 4.2 and Figure 4.3. Table 4.2 provides the count and duration of use of the modes and other features, and Figure 4.3 specifically visualizes the data related to the modes. Table 4.2 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.

Comparing the task performance of dyads based on the quantitative data on feature usage is challenging, as it requires consideration of both the quality of information acquired by participants and the specific intentions and experiences associated with each instance of feature use. For instance, when considering the duration of feature use, it is possible that prolonged use indicates the participant encountered challenges in obtaining the desired view (*e.g.*, the helper in dyad *D4*), or alternatively, that the participant was consistently gathering information throughout the entire period (*e.g.*, the helper in dyad *D2*). On the other hand, a shorter duration might suggest that the participant gave up quickly (*e.g.*, the helper in dyad *D5*) or that they efficiently gathered information (*e.g.*, the helper in dyad *D3*). Note that these examples are based on a high-level qualitative assessment. It is challenging to quantitatively capture these nuanced details about the quality of information acquired by participants and their intentions and experiences associated with each instance of feature use. Additional research is required to systematically capture these details for a more insightful interpretation of the data on feature usage and task performance.

One key high-level insight that we can draw from the quantitative data is that the helper inadequately engaging in independent exploration using the *helper-*

Table 4.2: Count and duration of use of *Periscope's* features. Dyads are ranked by amount of task completion (#1 being best). Count represents the number of times a feature was entered and then exited (or the session ended). Duration indicates the average time in seconds spent within a mode per count (t = mean (stdev)).

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

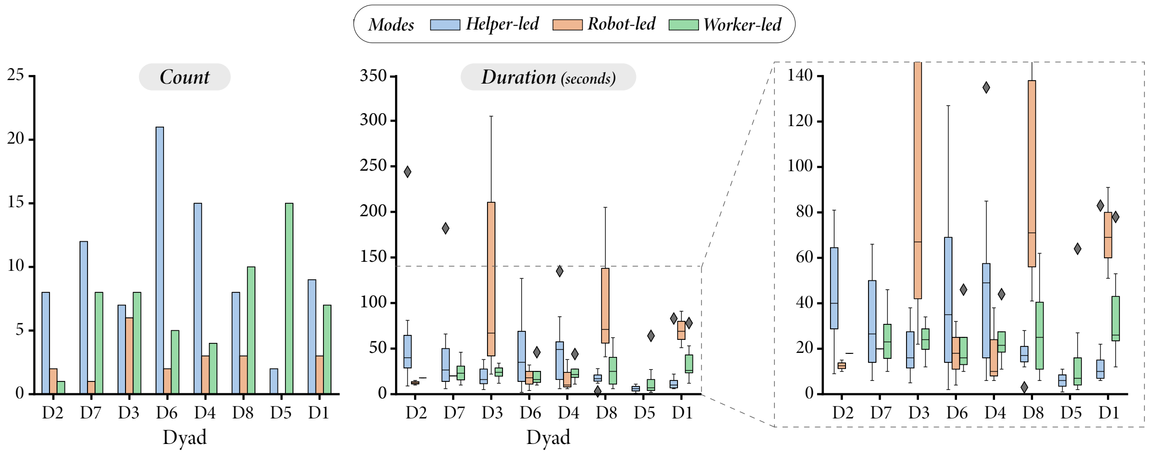


Figure 4.3: Visualization of the count and duration of use of *Periscope's* modes. Dyads are arranged in descending order based on most to least completion of the task. ***Left:*** Count plot depicting the number of times of use of the three modes in the data. ***Center:*** Box plot depicting the duration 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.

led mode and excessively relying on the worker to modify the view was typically associated with poorer performance (*e.g.,* dyad *D2* vs dyad *D5*). This trend was more pronounced in the duration data than in the count data. However, beyond this insight, it is challenging to identify any patterns regarding how the usage of features influences task performance.

4.5 Discussion

In this section, I discuss our system's ability to support our design goals introduced in Section 3.3 and present design implications for future systems.

Reflection on Design Goals

Table 4.3 summarizes our reflections on the design goals, highlighting both the abilities and limitations of the *Periscope* system in supporting them. In the rest of this section, I provide detailed reflections related to each design goal.

Note: Whenever a statement is connected to a result in Section 4.4, the relevant reference is included in parentheses.

Versatility: The frequent use of the system's features (Table 4.2) and consistent use patterns across dyads (Table 4.1) is encouraging, 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 (*e.g.,* E4.1).

Table 4.3: Summary of reflections on our design goals: "+" points indicate our system's ability to support the design goals; "–" indicate limitations identified in supporting the design goals.

Design Goal	Reflection
Versatility	+ Participants used the system's versatility to obtain diverse and context-specific views to support various task activities. The frequent use of the system's features and consistent use patterns across dyads is encouraging, especially since participants were not compelled to use any features.
	+ The shared context afforded by the system, including annotation and deixis, facilitated efficient and unambiguous communication.
	– Certain angles and locations were inaccessible due to the choice of robotic hardware, and the system did not adequately support repeated view specifications and precise views for certain task activities.
Intuitivity	+ The frequent use of camera controls suggests that users found it worthwhile to put in the effort to acquire information through camera control. The emergence of consistent patterns in camera control usage suggests an inherent intuitiveness associated with the controls.
	+ Autonomous robot behaviors were generally invisible to participants, contributing to the intuitiveness of camera controls.
	– Conversation pauses and dialogue related to camera control interrupted collaboration flow, but users took the time to obtain good views, after which interactions were smooth.
	– The system did not adequately support repeated view specifications and precise views. Additionally, the lack of autonomous behaviors in *Freedrive* affected camera control by the worker, and in the *Point* interaction, some users expected the camera to align with the direction of pointing.
Dual-user Interactivity	+ Helpers used the provided interactions to explore the workspace independently and in parallel with the worker's task execution. This allowed for efficient collaboration and timely interventions based on the helper's assessment of the task status without requiring constant dialogue with the worker.
	+ Participants took ownership of the point of view when they had ownership of a task or relevant information, resulting in frequent transfer of view control between the helper and the worker.
	+ There were two types of shared view control: (1) collaborative view control within a mode, where users jointly influenced the camera view through dialogue and the provided interactions, and (2) transfer of view control when switching between modes, where responsibility for camera control was shifted between the helper, the worker, and the robot.
	– Workers frequently moved the camera on behalf of the helper when helpers were dissatisfied with their user experience. Workers only occasionally leveraged their familiarity with the workspace to share information with the helper by using the provided interactions.
Congruity	+ Effective arbitration was facilitated by our leader-follower approach to designing the modes. Additionally, verbal negotiation between the helper and the worker during collaborative view control helped to achieve congruity.
	– There were instances of conflict in the data, particularly regarding the engagement of the worker-led mode, simultaneous attempts by the helper and the worker to move the camera, and a diminished role for the worker and robot during arbitration.
Usability	+ The system facilitated rich interactions and enabled remote collaboration on physical tasks.
	– Helpers expressed frustration with some latency and unresponsiveness in robot motion, especially when adjusting the view. Helpers may have different preferences for input sensitivity and direction based on their past experience and task context. Lack of transparency in some state transitions led to confusion when the robot became unresponsive in certain situations.
	– Workers split their attention between the task space, the robot, and the shared view on the laptop. The split-attention effect was mitigated to some extent by the embodied cues provided by the robot about the shared view and the helper's focus of attention.

Similar to prior work (Kraut et al., 2003; Gurevich et al., 2012), 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 *§Annotate*), was the most apparent use of the shared context to achieve efficient and unambiguous communication. Additionally, dyads used the shared visual context to ground references of task objects (*e.g., "L-shaped stuff"* for the base support in E4.5 and *"the blue one"* for the snap-connector in E4.6), 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 a proficient 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.,* E4.2).

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 *§Intuitivity*). These findings provide concrete directions for enhancing the versatility of future systems.

Intuitivity: The frequent use of the system's features (Table 4.2) to move the camera 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 4.1), 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 (*§Robot-led Mode Use Patterns*) and rare robot collisions required experimenters to restart the system (UP-19). 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 *§Versatility* 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.,* E4.2, E4.5, E4.10, E4.15, and E4.23). 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 E4.4, 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 UP-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 (E4.13).

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.,* E4.14). Here, the reset pose was useful on occasion since it may be used as a transitional pose when switching between subtasks (UP-17), or as a quick way to get a sufficient view of the workspace without much effort (UP-16).

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.,* E4.17 and E4.19).

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 (*§Helper-led Mode Use Patterns*). 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.,* E4.4). 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.,* E4.3). 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 (UP-7 and UP-13). 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 (Lanir et al., 2013; Mentis et al., 2020), 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 for each mode in *§Use Patterns*). 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* (Section 4.4). 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 (UP-5). 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 UP-15) and in the *worker-led mode*, helpers could adjust the view themselves or ask the worker to adjust it instead (UP-12). 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.,* E4.6). 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 (UP-10, UP-11, and UP-12), although there were also cases of the reverse (*e.g.,* E4.23 and E4.24). Although this demonstrates the potential of dual-user interactivity to compensate for system limitations, future designs of the system should minimize this behavior.

Congruity: The frequent transfer of view control between the helper and the worker within and between modes, which we discuss in *§Dual-user interactivity*, 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 *§Dual-user interactivity*) 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 *§Intuitivity*, and thus contributed to effective arbitration.

The leader-follower approach (see Section 3.6) 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 *§Dual-user interactivity*, 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 (*§Worker-led Mode Use Patterns*). 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 (*§Point*). 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.

Usability: The system facilitates rich interactions between the helper and the worker (illustrated through examples in Section 4.4) and enables dyads to remotely collaborate on physical tasks. This is promising for the system's usability. 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. This was addressed before running the user study in Chapter 5.

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.

Design Implications

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. We would like to acknowledge the utility of modes, as they explicitly distribute each user's and the robot's influence on the camera view, establishing clear leader and follower roles. However, exploring a modeless approach presents the possibility of introducing intermediate levels with varying degrees of influence, allowing for more nuanced interactions.

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.

Use pattern-based arbitration: Designing arbitration based on the use patterns presented in Table 4.1 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.

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.

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 system state.

4.6 Chapter Summary

This chapter presented insights into the *usage* of the *Periscope* system, as assessed through a formative evaluation involving 12 dyads in a semi-controlled lab study. Our study design drew inspiration from the principles of experimental simulations and maintained key elements of a natural collaboration scenario within a lab setting. This included considerations for task and workspace complexity, session duration, and opportunities for naturalistic use. During a 2-hour session, each dyad worked collaboratively on assembly tasks while physically located in separate rooms. Our empirical findings are derived from a rigorous analysis of qualitative data capturing participants' interactions with each other and the *Periscope* system. Our findings include diverse examples of interactions facilitated by the *Periscope* system, patterns of use of the system's features derived from analysis of these examples and other interaction data, and quantitative data for each dyad regarding their system use. The discussion elaborated on the significance of these findings to our design goals and their implications for the design of future systems. The study described in this chapter lays the groundwork for the upcoming chapter where we deploy the tool in tasks and conditions that more closely resemble real-world situations and focus on users who would employ such a tool more frequently in their routine activities.

5 APPLICATION OF THE PERISCOPE SYSTEM

This chapter presents a user study exploring the potential of the *Periscope* system in facilitating remote workforce training in manufacturing environments. In the study, we engaged instructors from a technical college and trainers from The Boeing Company to understand real-life opportunities and implications of remote training. In this chapter, I motivate our application scenario, present our research methodology, describe the user study design, provide a characterization of usage contexts based on findings from the study for modernizing workforce training by leveraging tools like the *Periscope* system, and discuss the validity of our findings. This chapter includes research from a manuscript in preparation (Praveena et al., 2024).

5.1 Motivation

In modern industrial environments, the increasing automation of routine tasks is leading to a shift in the nature of work. The tasks that require human intervention tend to be less frequent but require a higher level of skill and expertise. Traditional workforce training methods are inadequate in this evolving landscape and pose a number of challenges that we identified through discussions with our collaborators at Boeing. Firstly, these training sessions are typically designed around more common tasks and skills that might be automated away in the future and might not address the *high-skill requirements of non-routine tasks* (Autor et al., 2003; Nedelkoska and Quintini, 2018). Secondly, the time gap between training sessions and the actual occurrence of a task on the production line could lead to *skill decay*, *i.e.,* a decline in the retention of the necessary skills (Arthur Jr et al., 1998), and workers may not have the confidence or up-to-date knowledge to handle such tasks effectively. Thirdly, current training methods are *disruptive*, often requiring workers to leave their workstations and attend training sessions at separate facilities resulting in significant production downtime and loss of contextual conditions that exist in operation environments. Finally, these issues are compounded by an *aging workforce*;

experienced workers who possess advanced skills are retiring, creating knowledge transfer gaps as newer workers may not gain the same level of expertise through experience alone (DeLong, 2004; Burmeister and Deller, 2016). These challenges might leave workers unprepared to address critical, high-skill tasks and eventually result in operational inefficiencies and increase the risk of errors.

In this evolving industrial landscape, traditional workforce training methods can be augmented through new paradigms, such as *remote training*. Remote training solutions have the potential to enable workers to acquire or refresh necessary skills right at the moment they are needed, ideally in the actual work environment to ensure that the training is relevant, timely, and directly applicable to the task at hand. Remote training technology can *democratize expertise* (Brown, 2020) and connect experts with geographically dispersed individuals. This becomes particularly valuable for including an aging workforce and enabling them to remotely impart their knowledge and expertise to younger workers, even if they are not co-located or have physical limitations that make it challenging to engage in traditional on-site training and knowledge transfer.

In order to explore the opportunities and challenges presented by robotic camera systems for remote workforce training in manufacturing environments, we adopt an evaluation approach that incorporates the concepts of *participatory research* (Spinuzzi, 2005; Vaughn and Jacquez, 2020), *technology probes* (Hutchinson et al., 2003), and *context of use analysis* (Maguire, 2001a). I discuss this further in Section 5.2. The work in this chapter makes key contributions in two categories:

1. *Method* — an experimental paradigm to study the potential utility of a system by employing it as a technical probe and eliciting participatory critique.

2. *Data* — a reflective characterization of usage contexts related to remote workforce training in manufacturing environments where a robotic camera solution may be useful or face limitations.

5.2 Research Methodology

Our approach for understanding the utility of the *Periscope* system focuses on characterizing its potential usage contexts in real-world settings. Given the system's early developmental stage, it is impractical to deploy it directly in real-life use cases to assess its usefulness. Therefore, our methodology draws inspiration from human-centered design practices, particularly *usability context analysis* (Thomas and Bevan, 1996) or *context of use analysis* (Maguire, 2001a), which characterize anticipated circumstances of system use. We incorporated a subset of the factors outlined in these papers, considering the exploratory nature of our analysis and the nascent state of the paradigm under study.

Data for the analysis of context of use can be gathered using various methods, such as interviews, focus groups, surveys, field studies, contextual design, and artifact analysis (Maguire, 2001b). In our study, we engaged with potential real-world users of our proposed paradigm to elicit critical reflections on the utility of the system. This study design is aligned with the goals of *participatory research*, which involves *"systematic inquiry in direct collaboration with those affected by an issue being studied for the purpose of action or change"* (Vaughn and Jacquez, 2020). There are many variations of participatory practices and they are typically tailored to fit specific research contexts. For example, Nanavati et al. (2023) adopted a *community-based participatory research* approach in which the academic researchers engaged a person with permanent motor impairments as a community researcher. This individual contributed to creating design materials, conducting interviews, analyzing data, and co-authoring the paper. In another example, Spiel et al. (2017) conducted a *participatory evaluation* where autistic children, as intended users of the technology, were involved in defining the measures of success for the evaluation and interpreting the study results. In our approach, insights from participants not only provided data for characterizing potential usage contexts but also influenced the study design.

Hansen et al. (2020) emphasize the importance of hands-on experience with prototypes for spurring discussion and reflection by participants while examining how future technologies might alter current practices. In our study, we employed the *Periscope* system as a *technology probe* (Hutchinson et al., 2003) to provide participants with hands-on experience of our proposed paradigm. We made this choice because envisioning a work practice significantly different from the one they are familiar with can be challenging without tangible experience (Hansen et al., 2020).

Thus, we employ a multi-method approach for the work in this chapter. Our study design described in Section 5.3 draws inspiration from the principles of *participatory research* (Spinuzzi, 2005; Vaughn and Jacquez, 2020) and *technology probes* (Hutchinson et al., 2003) to inspire potential users of our proposed paradigm to share their perspectives on the real-life opportunities and implications of our work. Our findings presented in Section 5.4 are based on a process inspired from *context of use analysis* (Thomas and Bevan, 1996; Maguire, 2001a).

5.3 User Study

Early in the process, before designing the formal user study, we[1] engaged in preliminary discussions with a Boeing trainer who provided foundational, hands-on technical training for workers. We later recruited this trainer as a participant in the study described in this section. During the discussion, we introduced the *Periscope* system and presented two usage scenarios (see Figure 5.1) to stimulate conversation about training practices at Boeing and to explore where a solution like *Periscope* might be beneficial. This was also to gauge if these scenarios resembled any proprietary Boeing processes for which we would not be able to obtain pictures or documentation. Based on these usage scenarios, the trainer discussed two current training practices with us that may be promising for our application.

First, in a controlled setting such as a lab area in a training facility and under the guidance of the trainer, workers practice movements and procedures that they learn in a classroom setting. This training can last several days and include multiple sessions. The key elements of this type of scenario include (1) the trainer demonstrating techniques followed by the worker trying it hands-on, (2) the trainer

[1] This refers to Michael Hagenow, Jill Streamer, and myself.

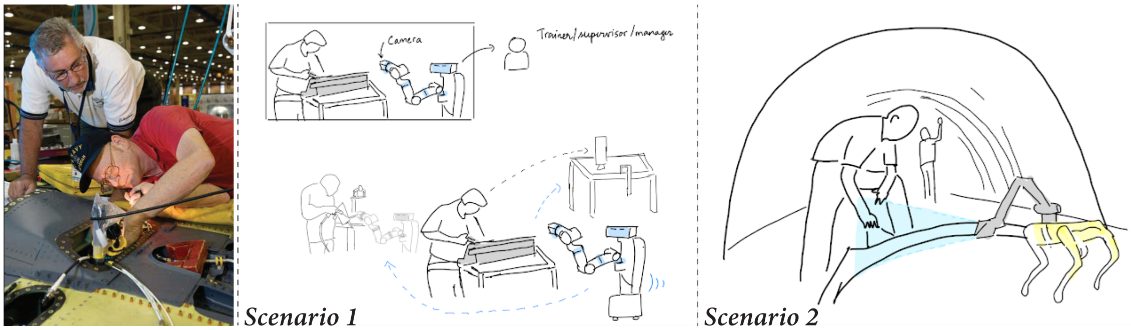


Figure 5.1: ***Left, © 2005 Boeing:*** A trainer/workplace coach, positioned on the left, works closely with a worker, positioned on the right, at a Boeing facility in St. Louis. ***Center:*** Scenario 1, used in the preliminary discussion with a Boeing trainer, illustrates training in a controlled setting such as a lab area in a training facility. ***Right:*** Scenario 2, used in the preliminary discussion with a Boeing trainer, illustrates on-the-job training in the production environment.

observing and evaluating the worker's performance and providing immediate feedback, and (3) the trainer maintaining a safe and supportive environment for workers when addressing any errors in their techniques. When evaluating the worker's performance to provide feedback (in this case, the task involves performing actions on a surface), the trainer employs a variety of methods. These include feeling the surface with their hands to sense certain material properties, using a flashlight to inspect the surface from a certain angle, listening to sounds produced by the tool as it makes contact with the surface, or using a measurement tool to objectively verify the accuracy of the task. Second, workers receive on-the-job training in the production environment following the initial training in the facility. Workers practice on parts closely resembling those in actual production, but the training is typically conducted in a separate area adjacent to the main production line. These settings are more chaotic, and the training period can extend over several months.

After this preliminary discussion with the trainer, subsequent conversations with our collaborators at Boeing highlighted that skill decay and the resulting disruption can impact production (these terms were introduced in Section 5.1). These discussions laid the foundation for the user study presented below.

Study Design

Our study was conducted in three phases. *Phase 1* involved discussions with Boeing trainers and subject matter experts to understand the characteristics of the task that would be suitable for the experimental simulation in *Phase 2*. *Phase 2* involved dyads consisting of instructors (remote) and students (local) using the *Periscope* system to collaborate on a car stereo installation task and then reflecting on their experiences. *Phase 3* involved reviewing video recordings[2] of a subset of interactions from *Phase 2* with Boeing trainers.

Participants

In *Phase 1*, we recruited two Boeing trainers experienced in composite materials and one subject matter expert (SME) who had experience with car stereo installation. In *Phase 2*, we recruited two instructors from a technical college (both men), one an automotive technician and the other an electrical engineer. We also recruited six participants (1 woman, 5 men; ages 20–64; mean age = 31.5) from the University of Wisconsin–Madison's campus community who had prior experience with manual work (*e.g.*, wiring, refinishing, equipment assembly, installation, and repair). In *Phase 3*, we recruited one Boeing trainer who had previously participated in *Phase 1*.

Procedure

Phase 1 — An experimenter described the study to participants and obtained written consent. After giving consent, the participants were introduced to the system, either through videos (for Boeing trainers) or through a guided demonstration of the system (for SMEs). The Boeing trainers were asked about their current training practices and and how they perceived the system being deployed and used within their facilities. The SMEs guided the local experimenter through the car stereo installation task. Once the ideal setup was completed, they were asked about

[2] https://youtube.com/playlist?list=PLilPy_VXz5rIfRaYtfRynyR3nBNWI3LLI&si=5iYLSg4RXnm863aL

potential issues that could arise in car stereo installation and how these could be simulated in the lab. Subsequently, the local experimenter tested some of these aspects and received feedback from the SMEs. The study lasted for one hour.

Phase 2 — In this phase, pairs of instructors and students were informed that they would be completing a wiring and routing task involving the connection of a car radio, amplifier, and speakers, as illustrated in Figure 5.2. Note that *student* refers to the role of the participant; not all participants in the student role were actual students in real life. The instructors participated in a total of four sessions: one for training with the system and understanding the study task, followed by three sessions to collaborate with three different students. In the initial training session, instructors were asked about potential issues that could arise in car stereo installation to solicit feedback on the task design. After the initial session, instructors received a detailed lesson plan for the task (included in Appendix B), which explained the process of wiring the system, guidelines for routing wires, and common troubleshooting techniques for a non-working system.

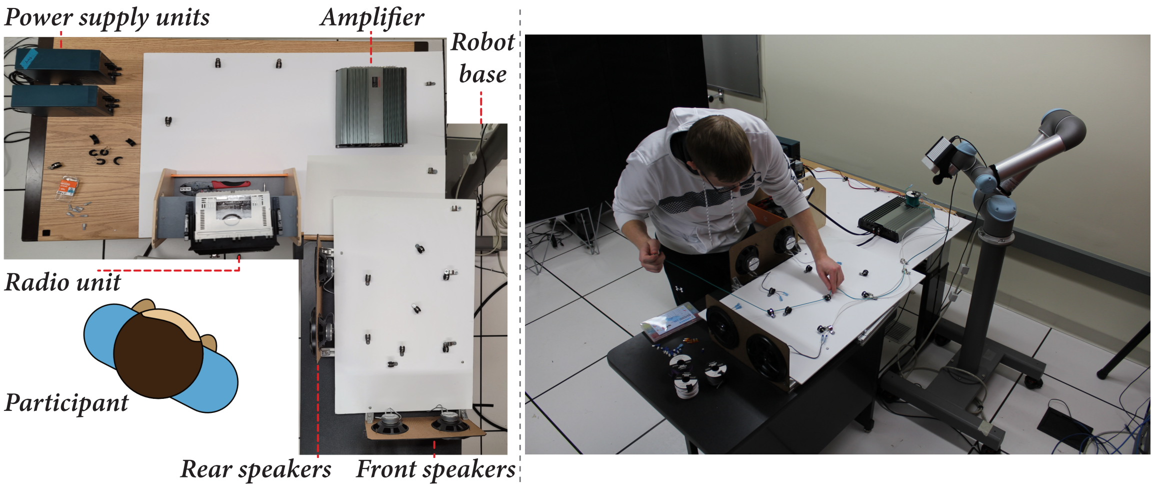


Figure 5.2: ***Left:*** A top-view of the setup for *Phase 2*. The local participant (student) completes a wiring and routing task involving the connection of a car radio, amplifier, and speakers. The remote participant (instructor) uses the robot to view the workspace and instruct the student. ***Right:*** A photo of the real setup of the student and the robotic camera that the instructor can access.

Students arrived 30 minutes before the instructors. An experimenter described the study to students and obtained written consent. After giving consent, students were introduced to the *Periscope* user interface. Students then received in-person instruction from the local experimenter for 15–20 minutes. Students were shown how to use each of the tools needed for the task, and given verbal instructions on best practices. Then students were walked through the process of connecting a speaker to the amplifier, which allowed them to get some initial practice using the tools. Similarly, students were given verbal instructions on how to route the wires and were advised to avoid wiring around hard corners. Students were given approximately 10 minutes to wire the speaker and route the wires.

After completing this initial setup, students were asked to wait outside the setup area, so that the local experimenter could complete the task setup for them to work through with the remote instructor. This break allowed the experimenters to introduce common problems into the setup to ensure that there would be issues for the students and the remote instructor to find and troubleshoot. After the setup was completed, the students were brought back into the room, and the remote instructor connected to the *Periscope* system from their home or office. This point was approximately the 30-minute mark of the study.

After obtaining consent from the remote instructor, they were introduced to the student and given a few minutes to greet one another. Both the instructor and the student were then informed about the collaborative task of connecting the remaining speakers to the amplifier and testing the system. They were asked to make as much progress as possible within 25 minutes. After this collaboration, the instructor and the student were separated to independently complete surveys and respond to semi-structured interview questions. Finally, both the instructor and the student received compensation for their participation. The total duration of the study was 90 minutes for the students and 60 minutes for the instructors.

Phase 3 — An experimenter described the study to the participant and obtained written consent. After giving consent, participants were shown videos of interactions from *Phase 2* of instructor-student dyads utilizing the *Periscope* system and collaborating on a task. During the review of these interactions, participants were asked semi-structured interview questions regarding their perceptions of the system and the interaction, as well as how they perceived it being deployed and used within their facilities. The study lasted for one hour.

Measures

In *Phase 1*, participants were asked semi-structured interview questions to understand their current practices and how they envision a system like *Periscope* fitting into those practices. We also solicited feedback on the suitability of our task design for its relevance to real-world contexts of use.

In *Phase 2*, instructors and students were given formal survey metrics. However, these metrics are not elaborated upon here, as they were not analyzed and do not have any impact on the discussions within this dissertation. Both instructors and students participated in semi-structured interviews aimed at understanding their experiences during the session and reflecting on their use of the Periscope system.

In *Phase 3*, participants were asked semi-structured interview questions to understand how they perceived the recorded interactions, their interpretations of the perceived system usage, and their opinions on how these might align with or differ from their own perceived usage.

Interviews in *Phase 1* and *Phase 3* were recorded, resulting in ~4 hours of data. During *Phase 2*, we screen-recorded the student's interface and recorded the workspace (including the student and the robot), resulting in ~3 hours of in-person instruction data and ~5 hours of remote instruction data. Interviews were also recorded, resulting in ~5 hours of data.

Analysis

We follow an interpretive approach (Walsham, 2006) for the preliminary analysis of the data collected in this study. Interpretive research acknowledges that the background and experience of the researchers is valuable in interpreting the data. Researchers can uncover meanings in an iterative way as they become more immersed in the data. Walsham (2006) recommend describing the researchers' role for interpretive research.

PP was the lead researcher in developing the *Periscope* system and conceptualized the potential application of the system for remote training. NW and PP both served as experimenters in every session across all phases. All experts (trainers, instructors, and SME) were familiar with one or both of the researchers through multiple interactions.

In phase two, NW served as the local experimenter, while PP served as the remote experimenter. NW conducted the in-person training, and PP demonstrated the system to both the instructor and the student. NW and PP exchanged field notes, both descriptive and reflective, during the collaboration sessions. NW interviewed the students, and PP interviewed the instructors. Towards the end of the study session, NW joined the instructor interviews after completing the student interviews. After every session, NW and PP debriefed to discuss key takeaways from their respective interviews and to compare the current session with any previous sessions. NW and PP also debriefed after interviews with other experts.

NW and PP chose a section of the interaction data for review, focusing on the instances when the robot was moving around the amplifier. This action was selected because it was a common element across all interactions and posed challenges due to the spatial constraints of the setup. NW reviewed the video interaction data and provided descriptions, which PP then confirmed and revised. Additionally, PP added interpretive findings on the task design and potential applications, which NW then confirmed and revised.

5.4 Findings

In this section, I present a subset of interactions observed during *Phase 2* of the study where instructor-student dyads collaborated using the *Periscope* system. Interpretations of these interactions, along with field notes, inform the characterization of usage contexts that I will present later in this section.

Example Interactions

We reviewed three sessions for each instructor (I1 and I2), focusing on the instances when the robot was moving around the amplifier. In the following description, we refer to the *helper-led mode* as *independent control*, *the robot-led mode* as *hand-tracking*, and the *worker-led mode* as *manual assistance*. Further discussion on this terminology is provided in Section 6.3. The following descriptions are "thick," a term used by Geertz (2008) to describe accounts of qualitative data that extend beyond factual recounting of events and provide the researchers' interpretations of the context surrounding the events.

Instructor 1

In the first session with P1, I1 struggled with robot control. I1 was on a wireless Internet connection that caused significant delays in the camera feed, necessitating significant reliance on verbal communication to monitor the student's progress. Throughout this session, whenever P1 worked with the amplifier, I1 switched to manual assistance to allow P1 to adjust the view. I1 did not modify any of the views, even when P1 caused the view to rotate by 90°.

In the second session with P4, I1 switched to a wired Internet connection and more actively attempted to control the robot. Initially, while P4 described the study setup, I1 tried to move the robot's camera using independent control to view the amplifier's connections to the radio. This led to challenging configurations, causing repeated low-speed collisions with the table and triggering safety stops. These stops required experimenter intervention to resume robot movement. Later in this session, I1 used a mix of independent control and manual assistance from P4 to better view the amplifier's connections to the speakers, even adjusting the viewing angle when necessary. Towards the end of the session, I1 primarily used hand-tracking as P4 repeatedly moved between the speakers and the amplifier. Additionally, I1 used the zoom and orbital adjust capabilities to modify the view while continuing to maintain hand-tracking.

In the final session with P6, I1 encountered several instances of camera cable disconnection due to the cable becoming tangled around the robot. This required experimenter intervention to reconnect the cables. I1 needed to refresh the browser for an updated camera feed. The cable entanglement resulted from the way I1 and P6 interacted using the camera system. I1 used independent control to follow P6 to the amplifier, and then had P6 manually assist in moving the camera. However, P6's assistance often led to awkward robot configurations which complicated independent control for I1. This resulted in I1 having to use the robot's reset button to achieve a better configuration and regain control over the robot's movements. Unfortunately, these resets sometimes caused cable disconnections and one near-collision with P6.

Overall, I1 demonstrated considerable improvement in system usage, becoming increasingly comfortable with various features for exploring the workspace. This improvement was particularly noticeable from the first to the second session, which was partly due to resolving connectivity issues by switching from a wireless to a wired connection.

Instructor 2

In the first session with P2, I2 solely used independent control, sometimes leading to sub-optimal viewing angles, such as only seeing the top of the amplifier and not the connections. In the second session with P3, I2 started with hand-tracking while P3 was explaining the study setup, but primarily used independent control for the remainder of the session.

In the final session with P5, I2 primarily used independent control, with a single instance of requiring manual assistance to view the amplifier connections. This need for manual assistance arose when P5 was connecting the speaker wire to the amplifier and the connector became detached. Due to challenges in verbally communicating the instructions, I2 had P5 manually adjust the camera for a clearer view of the connectors. Subsequently, I2 asked P5 to check the laptop to see the annotation I2 had added to the feed. This instance was the only time the annotation feature was used across all six sessions.

Overall, I2 mostly relied on independent control, preferring not to "waste" the student's time. This approach sometimes resulted in sub-optimal viewing angles, especially when examining the amplifier connections, but generally led to smoother interactions between the instructor and students.

Comparison

Comparing the instructors' approaches and system usage, I2's comfort and confidence with the system were notably higher from the outset. During *Phase 3* of the study, when we showed video clips of the interactions to a Boeing trainer, they also commented on the challenges that I1 appeared to be facing while using the system.

Usage Contexts

In this section, I first present an analysis of the task design. This is followed by a characterization of three potential scenarios related to remote workforce training: formal schooling, in-house foundational training, and workplace coaching.

Reflection on Task Design

Our goal with the task design was to create a task that allowed for meaningful progress using the system's current capabilities, while also reflecting potential contexts of use in the real-world. The feedback from participants provided us with insight into whether this goal was met. The Boeing trainers and the SME in *Phase 1* considered the task as an acceptable abstraction of potential contexts of use. In *Phase 2*, we observed that all instructor-student pairs made meaningful progress (*e.g.,* wires were connected, task-related information was exchanged, and problems were diagnosed) during their session. Occasionally, we observed that alongside task progress, instructors and students also engaged in small talk, indicating a level of comfort in the interaction. Thus, it seems that our task was adequate in achieving what we set out to do. Below, I discuss the characteristics of our task design to identify scenarios where a robotic camera solution might be beneficial or may encounter limitations.

Task type: There were several task types of interest to Boeing, including wiring, sanding, riveting, and sealant application. Wiring appeared to be the most practical for our task design for an exploratory evaluation as it did not require safety equipment, dangerous or specialized tools, and was likely a skill with which a sufficient number of engineering students on campus would be familiar. Boeing trainers, experienced in composite manufacturing, also felt that processes similar to working on composites were too tactile to be sufficiently supported by the system and concurred that wiring was a suitable candidate. The car stereo wiring was well-received by instructors; even though it was not precisely in their subject area, they were very familiar with it and the process. Likewise, Boeing trainers were well acquainted with the task from experiences outside of work.

Takeaway — Wiring appears to be a practical task where visual observations are likely adequate for making significant progress. The car stereo wiring task, though complex enough to require expertise, was accessible to a wide range of participants. This task type is not heavily constrained by its physical environment and is a low-risk activity, as it involves minimal issues related to noise, heat, vibration, health hazards, or the need for protective equipment.

Task process: Based on conversations with Boeing trainers in *Phase 1*, it appeared that that demonstration played a significant role in the instruction process, especially in the early stages of training. However, demonstration by the remote expert is not supported by the system (users would need to rely on the video conferencing view for this purpose). Consequently, in *Phase 2* we decided to conduct in-person training with the students for 15–20 minutes to provide them with some basic knowledge. Although we considered the possibility of students learning from videos, this approach did not perform well in the pilot. For the remote collaboration portion of the task that lasted 25 minutes, we selected two aspects where we believed the instructor's expertise would be particularly valuable: *skill enhancement* and *troubleshooting*. A Boeing trainer in *Phase 1* had identified these as part of their process on the shop floor.

Skill enhancement seemed to be quite effective in the study since our training was basic and the instructors possessed more expertise. They were able to assess the student’s technique and comfort through video (from the robot-mounted camera) and provide advice. This feedback from instructors was not part of the formal lesson plan, and there was no mandate to provide it. Instead, the feedback (such as a suggestion to tug on connectors to check for loose connections) was contextual and spontaneous, and a byproduct of instructors offering recommendations as they watched the student executing the task.

Troubleshooting presented a mixed outcome, primarily because not all aspects within the workspace were readily visible to the instructor. For instance, issues like a blown fuse or a cut wire were challenging to visually assess through the camera feed. Therefore, the success of troubleshooting largely hinged on the worker’s initiative. There were instances when the worker touched the cut wire but did not mention it to the expert, or they pointed out the cut wire, but the expert missed seeing and hearing it, leading to the problem remaining unaddressed.

Takeaway — A blended instructional approach worked reasonably well, consisting of in-person instruction to impart foundational knowledge, followed by skill enhancement and troubleshooting facilitated by the system. There appeared to be some qualitative differences between the characteristics of the in-person and remote portions of the task. During the initial in-person portion, there was a continuous flow of information from the expert (the experimenter) to the student. The training followed a routine and structured format, and the resulting interaction seemed fairly consistent across all students. In the later remote portion involving skill enhancement and troubleshooting, there were sporadic interventions by the expert as the worker had a basic knowledge of the task and could work independently for an extended period of time. These interactions were more open-ended, contextual, and personalized, and every session took a unique path, even with the same instructor.

External factors: All students were rarely seated and frequently moved around the workspace, especially between the speakers and the amplifiers by walking past the power supply. Conversely, the setup remained relatively static, with students not attempting or unable to reposition the radio, amplifier, speakers, and other equipment to provide a better view. This reflects real-world contexts where workers will likely exhibit movement during physical work. This consideration is not only for the sake of providing optimal views but also for safety reasons when workers approach the robot. During our study, we observed a near-collision incident when the robot was in autonomous motion, although, in general, students appeared comfortable working in close proximity to the robot. Additionally, it would also not be practical in real-world manufacturing settings to move components closer to the camera for visibility; instead, the camera would need to be adjusted to get a closer view of the component.

Due to the in-person training, students were familiar with relevant terminology and processes (*e.g.,* crimping, wire cutter, grommet), which allowed instructors to convey instructions verbally without the need for demonstrations or circumlocution (*e.g.,* "the round thing" or "the yellow tool to your left"). Additionally, throughout all sessions, annotation use was minimal and occurred only once. Instructors generally did not perceive an inherent need for annotation, perhaps due to the students' familiarity with the relevant terminology. However, during the interview, when the experimenter pointed out that an instance of communication breakdown could have been avoided with annotations, they expressed their hesitation to use this feature. This stemmed from the students' constant movement, as using annotations would have required the student to make a deliberate effort to access the laptop (which is what happened in the single instance of annotation use during the session subsequent to the interview).

Finally, concerning usage of system modes, instructors primarily changed the view using independent control, occasionally requesting manual assistance from the student to adjust the view. The instructor who appeared more at ease with the system believed that seeking manual assistance wasted the student's time. Hand-tracking served a dual purpose—it was utilized for monitoring of the worker during task execution and also provided a rapid but less precise method for assisting the instructor in adjusting the view. The instructor who was less comfortable with the robot control frequently initiated resets, either to establish an intermediate pose during transitions or to set a comfortable starting pose. Overall, both instructors expressed more comfort with the system with every subsequent session.

Takeaway — The behaviors displayed by dyads in *Phase 2* of this study exhibited similarities and differences when compared to the previous study in Chapter 4. The similarities are encouraging for generalizability. The differences are influenced by various contextual factors, including the larger workspace, increased worker mobility, shared vocabulary established prior to the initiation of the interaction, and the nature of the task components.

Scenario 1: Formal Schooling

A remote instruction scenario in a technical college or a vocational school closely resembles the design of our study. Instructors who participated in our study confirmed that even during in-person instruction, particularly during skill enhancement and troubleshooting work, they would likely not physically interact with anything and would only use visual information to guide the student. However, they acknowledged having more flexibility with viewpoints in-person compared to what our current system supports. A blended instructional approach could be effective in this context, with students receiving in-person instruction at a technical or engineering college while also accessing knowledge from remote experts elsewhere in the country.

One of our instructor participants connected us with Normandale Community College in Bloomington, Minnesota, USA. This college offers a certificate curriculum in vacuum technology that is fully accessible to remote learners, including hands-on learning activities, through a remotely accessible vacuum equipment trainer system. Instructors in this program simultaneously engage with local students and those joining via a telepresence connection. This work was funded by National Science Foundation (NSF) awards 1400408 and 1700624.

According to the college's website[3] and a conversation with individuals involved in remote training at Normandale, the program's motivation is to increase enrollments for courses that have high capital equipment costs but low enrollments when

[3] `https://www.normandale.edu/academics/degrees-certificates/vacuum-and-thin-film-technology/shaping-the-future-of-vacuum-technology-education.html`

limited to only locally accessible students. This quote from a video on the website[4] encapsulates their motivation:

> The proposal advanced an idea to test the delivery of vacuum technology classes in a telepresence classroom as a strategy to expand the audience for vacuum technology classes. We would enroll people from more distant locations who could not attend classes in person on campus. We were already aware of glass coating industries located in the state of Minnesota that used vacuum systems extensively. However, the locations of those companies made it difficult, if not impossible, for those workers to attend Normandale's Vac Tech classes in person.

While this setup is quite different from ours (they have remote learners, whereas we have remote experts), it is encouraging to observe real-world examples of modernizing instruction for manufacturing work. If we were to investigate this use case further, we would have to redesign the control mechanisms and policies for this particular context of use.

Scenario 2: In-house Foundational Training

Training effectiveness can vary between laboratory settings and operational contexts. This discrepancy was highlighted by a Boeing trainer, who noted differences in student comfort and confidence when working with "live parts" in operational contexts as opposed to mock-ups in laboratory settings.

> I think there's an advantage for me to see, watch, observe them work... especially with **live parts**...A lot of times students come to my facility off-site in the training lab and their demeanor is a little bit different because the parts, they aren't live...I read a lot of that in people's **body language**, their confidence, and all that. So, seeing them work on something in the offsite training facility — they may be even a lot more loose, they

[4] `https://www.youtube.com/watch?v=vk14iCWabhE`

> may be a lot more comfortable, they may actually perform better. And then get them in the real world on live parts and they might not, and I would be able to see that probably a whole lot better. And I can, I believe I can help people a lot more when I see those things...and I've done it before. I've been in the shop before...or on the live part with an individual who's suddenly **uncomfortable**, because now, yes, exactly, the stakes are higher, right?

For trainers, observing workers directly during on-the-job training, while potentially helpful for identifying learning challenges, requires visiting the shop floor, which might not be part of their regular routine. In practice, they depend on verbal feedback from the team to which the worker is assigned. Alternatively, a different type of trainer known as a workplace coach, who regularly visits the shop floor, can provide insights into the worker's learning challenges. The possibility for trainers to remotely view workers in the production environment could offer advantages by allowing for direct observation rather than relying on second-hand reports. A remote solution provides the opportunity for sporadic check-ins with workers over the months of on-the-job training and ensuring ongoing support for any emerging issues or further learning needs. Additionally, a remote solution might offer efficiency advantages by allowing a single trainer to simultaneously check in on multiple distributed workers.

Scenario 3: Workplace Coaching

The previous scenario mentioned the workplace coach who is able to support workers on the shop floor. An article about workplace coaches in Boeing[5] mentions the motivation for this role as: *"Since the tools and processes that employees work with are often complicated, detailed and fluid, having coaches available on the job often prevents sending an employee back to a classroom or spending time looking for answers."*

[5] https://web.archive.org/web/20190304111258/https://www.boeing.com/news/frontiers/archive/2005/december/ts_sf.html

This description of the role of a workplace coach aligns with our task characterization of skill enhancement and troubleshooting, suggesting it could be a good fit for a remote solution like *Periscope*. Regarding the future of coaching, the author of the article notes:

> Similarly, for the 787 program—a global partnership with a widely dispersed work force—a planned **virtual** Workplace Coaching System will allow coaches to be everywhere employees are located, maximizing limited resources. The future Workplace Coaching System "could ultimately become the hub of training and support beyond the classroom," said John Fisher, workplace coaching manager for the LTD organization. "It will provide a continuous learning environment for engineers around the clock and enable various Boeing programs to share ideas and best practices seamlessly across the system in real-time."

The advantages highlighted in the article include enabling coaches to be accessible wherever employees are located, fostering a continuous learning environment, and providing real-time support. These benefits align closely with our vision of solutions designed to meet the challenges of the evolving nature of work in an increasingly automated world.

To equip workers with the necessary technical skills for handling infrequent (or non-routine), yet critical and complex tasks, traditional workforce training methods can be enhanced with technological solutions like *Periscope*, which can enable "just-in-time" training. This technology would allow coaches to remotely observe workers in their production environment and enable workers to acquire or refresh skills right at the moment they are needed. This ensures that training is relevant, timely, and directly applicable to their current tasks. This approach can reduce the disruptions typically associated with conventional training methods, where a worker is taken off the line to attend training at a separate facility. Similarly to the previous scenario, this solution can enable a single coach to simultaneously monitor and assist multiple workers across different locations.

Pursuing the remote on-the-job training or the workplace coaching applications requires greater mobility from the robot platform so that it can cover larger workspaces and also so that it can be conveniently stored when not in use.

5.5 Discussion

The research process in Chapter 5 shares similarities with the one in Chapter 4. The patterns of use outlined in Section 4.4 are interpretive in nature. Based on my field notes, I identified some use patterns that might have remained unnoticed if I had approached the data with a bottom-up coding strategy, as it was not immediately apparent which combination of data channels (*e.g.,* dialogue, interactions with the system, worker actions, camera motions) captured the phenomenon of interest. These patterns informed the thematic analysis, guided the coding of the data, and led to more precise findings. Therefore, I advocate for an interpretive approach to preliminary analysis as a valuable step in understanding complex qualitative data.

In this chapter, the interpretive findings are based on preliminary insights that both Nathan and I, as researchers immersed in the data and facilitating the study, have identified. The task characterization in Section 5.4, while it could benefit from further analysis, is amenable to readers independently verifying it using the video recordings[6]. This is aligned with a criteria of rigor for interpretive research suggested by Lincoln and Guba (1988). I recognize the potential risk of overinterpreting the data (Walsham, 1995), and because of that, the scenarios presented are anchored in real-world needs and issues relevant to this chapter's work. Essentially, our characterization affirms the existence of needs, problems, and behaviors previously recognized, and we contextualize these within the design of the study and the system. I argue that this approach was suitable and valid for the goals of this work: (1) demonstrate the potential utility of the system in an instructional scenario (*e.g.,* instructors and students were able to make progress on wiring and troubleshooting), and (2) provide preliminary insights about potential

[6] `https://youtube.com/playlist?list=PLilPy_VXz5rIfRaYtfRynyR3nBNWI3LLI&si=5iYLSg4RXnm863aL`

utility in real-life contexts for generating interest to support further research (*e.g.*, some of this work was submitted as a concept paper). Our future work will include a full analysis, which we anticipate will take several months and will support this work through quantitative and qualitative insights on usability.

5.6 Chapter Summary

This chapter presented insights from a user study on the *application* of the *Periscope* system in facilitating remote workforce training in manufacturing environments. Our study design drew inspiration from the principles of participatory research and involved potential users of this proposed paradigm in multiple phases of the study. We conducted the study in three phases. *Phase 1* involved discussions with Boeing trainers and subject matter experts to understand the characteristics of the task that would be suitable for the experimental simulation in *Phase 2*. *Phase 2* involved dyads consisting of instructors from a technical college (remote) and students from the UW–Madison campus (local) using the *Periscope* system to collaborate on a car stereo installation task and then reflecting on their experiences. In this phase, we employed the *Periscope* system as a technology probe for providing participants with a hands-on experience to envision a work practice significantly different from the one they are familiar with. *Phase 3* involved reviewing video recordings from *Phase 2* with Boeing trainers. Based on field notes and conversations with participants, we characterize the potential usage contexts where a solution like *Periscope* could be beneficial. This process of characterization drew inspiration from context of use analysis in human-centered design. Our findings include “thick” descriptions of user interactions, an interpretive analysis of the task design, and the characterization of three potential scenarios related to remote workforce training: formal schooling, in-house foundational training, and workplace coaching. These findings suggest the potential utility of robotic camera solutions in real-world settings.

6 GENERAL DISCUSSION

6.1 Significance of Work

This work demonstrates that robotic camera systems, as a form of *robotic groupware*, can support human-to-human collaboration in remote settings that involve physical tasks. The significance of the work in this dissertation is the instantiation and characterization of such a system. Through the design, usage, and application of the *Periscope* system, my work offers insights into a novel point in the design space of robotic groupware, specifically groupware built on cobot platforms and their potential for enabling remote collaborative work.

In addition to the system and data outcomes, this dissertation presents methodological advancements for the design and evaluation process of exploring the design space of robotic groupware. The absence of established guidelines for navigating this nascent design space necessitated an adaptive methodological approach, carefully tailored to meet the unique requirements of our research inquiry. While these methods are rigorous, they have not yet been formalized into a precise framework, and this is typical in initial explorations of a design space. The following discussion reflects on the methodologies employed in each core chapter of the dissertation and some potential strategies for formalizing these processes in future research.

Design

The *Periscope* system serves as an operational proof of concept of a robotic camera system designed using a shared camera control approach and instantiated through the user interactions, autonomous behaviors, and arbitration policies described in Sections 3.5 and 3.6. These specific interactions, behaviors, and policies are important because they collectively work *well enough* within a system to allow us to explore and characterize (as elaborated in Chapters 4 and 5) an interesting point in the design space of robotic groupware — supporting remote expert scenarios for physical tasks within complex and dynamic 3D environments. While other (potentially "better") interface variants could similarly facilitate this exploration, it is crucial to acknowledge that there currently exists no robotic groupware system that supports the context outlined in Section 1.1. Therefore, without prior knowledge of the feasibility or potential value of such a system, my objective was not to create the "best" version of the interface possible, but rather to develop a quick and reasonable probe to investigate an interesting point within the design space of robotic groupware. This approach aligns with the concept of developing a Minimum Viable Product (MVP) and Minimum Viable User Experience (MVUX) as discussed in Section 3.7.

Adopting the MVP approach inherently involves a trade-off between the effort required to implement a feature and its effectiveness. For instance, rather than relying solely on on-screen annotations, using a laser pointer or projector on the robot to overlay annotations directly onto the physical environment may be beneficial (Gurevich et al., 2012). Additionally, providing *stabilized* annotations that remain attached to scene objects even as the robot moves may be valuable (Fakourfar et al., 2016). However, while these enhancements could potentially improve the effectiveness of annotations, they necessitate additional resources and effort, such as integrating extra hardware (*e.g.,* laser pointer or projector) or implementing more algorithms (*e.g.,* object tracking to ensure annotations stay aligned). My goal within the design process (in order to develop a *quick and reasonable probe*) was to implement features that could be realized with minimal effort while still supporting the necessary awareness and control needs. Since this dissertation highlights the potential value of robotic camera solutions, exploring further enhancements of the individual features would be worthwhile in future work (some of these are discussed in Section 6.4).

At a high level, the MVUX for *Periscope* included enabling remote users to independently explore the space (using *Target* and *Adjust* in the *helper-led mode*) and adjustment of a view that was, at least in part, managed by the local user (*Adjust* in *robot-led mode* and *worker-led mode*). *Reset* was required to allow recovery for the remote user from odd configurations and *Annotate* for supporting referencing. The local user's level of control over the view increased from *Point* (in the *helper-led* *mode*), to *Direct* (in the *robot-led mode*), and to Freedrive (in the *worker-led mode*). *Freedrive* also served as a way for local users to recover from odd configurations and they could directly reference using gestures in the physical space. We implemented several autonomous behaviors to support these user interactions with low-dimensional input. These interactions and behaviors were arbitrated using a leader-follower paradigm, which can be extended or modified (discussed in Section 6.3) to suit more nuanced collaboration processes.

Thus, the design process exposes (1) the concepts from HCI, CSCW, HRI, and robotics that are crucial for developing an operational instantiation of robotic groupware, and (2) a set of user interactions, autonomous behaviors, and arbitration policies that work well enough together to serve as a functional prototype for exploring the viability of cobot-based groupware to support remote collaborative work. *Periscope* does not represent the best possible version of such a groupware system; it represents a feasible system that is the first reasonable representation of a novel point in the design space that is being explored in this dissertation.

Research vs Engineering: The research contributions in this work towards building a robotic groupware system were the design of control mechanisms (user interactions and autonomous behaviors) and control policies (leader-follower based arbitration) based on a shared camera control approach. These contributions (1) enabled us to leverage the advanced capabilities of cobot platforms without overwhelming users with the complexity of the tool, and (2) allowed users to maintain the desired level of control expected from a collaborative tool. Additionally, the research team invested a significant engineering effort to make the system operational. This included ensuring safety from collisions, developing a professional-looking web interface, reducing the latency of the camera feed, and deploying it on the Internet, accessible at `periscope.cs.wisc.edu`.

Research prototypes often do not reach this level of operationality. However, this was a priority from the beginning because prior work in robotic groupware that is not operational-level is evaluated in simple contexts (*e.g.*, relatively uncluttered workspaces, with short and structured tasks) that do not reflect the complex and dynamic real-world context of using groupware to support physical work. The fact that the system was operational allowed us to conduct 2-hour evaluation sessions with dyads in Chapter 4. It also enabled us to recruit remote instructors from technical colleges for the study in Chapter 5 and allowing them to access the robotic camera from their homes or offices. Recruiting experts like these is challenging, so making the system operational expands the pool of potential participants who can test the system and provide feedback. On a meta level, I was remote for the entire duration of the study in 5 and used the *Periscope* system to facilitate the study.

Usage

In Chapter 4, our findings included diverse and rich examples of interactions enabled by the *Periscope* system and patterns of use of the system's features. We assessed the system's strengths and limitations based on how well these findings supported our design goals. This approach to understanding system use or *usability* seems somewhat imprecise when compared to many HCI research and industry practices. Typically, usability is assessed by showing improvements in human performance or work practices against a baseline set of metrics or by providing evidence that people can achieve specific goals using the system, like performance measures or task completions (Greenberg and Buxton, 2008). Our evaluation deviates from these norms, as it lacks a comparative benchmark and does not rely on an analysis of outcomes realized by users[1].

Greenberg and Buxton (2008) argued in their essay on usability evaluation that *"the choice of evaluation methodology – if any – must arise from and be appropriate for the actual problem or research question under consideration."* From the outset, my goal was to explore the feasibility and promise of this paradigm of using cobots for robotic groupware. Thus, we developed the *Periscope* system to serve as an instantiation of this paradigm. The focus of the evaluation in Chapter 4 was on understanding the feasibility of the paradigm, rather than on benchmarking the effectiveness of the

[1] The ranking of dyads in Section 4.4 is a measure of performance outcome, but we do not base any significant claims on this ranking.

instantiated system. Hence, I believe that the deviation from usability evaluation norms is justified. I posit that the study's findings demonstrate that a shared camera control approach is a viable method for realizing the feasibility of cobot-based groupware and for supporting consistent and meaningful user engagement.

While beyond the scope of this dissertation, it would be interesting to compare the ways in which users engaged with the system and interacted with each other in the two studies presented in Chapters 4 and 5. The similarities would be encouraging for generalizability. The differences would reveal the various contextual factors that influence user behavior, such as the size of the workspace, worker mobility, shared vocabulary established prior to the initiation of the interaction, and the nature of the task components. In addition, the particular implementation of the system (*e.g.*, specific features, graphics, or devices) can significantly impact user outcomes, and minor changes in system functionalities may result in unexpected variations in user engagement. The methodologies employed in this dissertation should be applicable for characterizing other system implementations, but further research is necessary to determine the extent to which the results of this dissertation generalize across different system implementations.

Application

In Chapter 5, we explored the potential of the *Periscope* system in enabling remote workforce training in manufacturing environments. With evaluating innovative technologies, there is a danger that users do not understand how they can be useful to them, particularly when they fall short of existing practices, such as in-person training in our situation (Christensen, 2013). Hence, utilizing *Periscope* as a technology probe was crucial in enabling participants to envision its utility. Furthermore, the opportunity for some participants to engage in multiple sessions was beneficial, as it allowed them to become more familiar with the paradigm. Venkatesh and Bala (2008) proposed that as users accumulate experience with a system, they are better positioned to evaluate the likelihood of achieving high-level goals (*e.g.*, perceived usefulness) by drawing on insights gained from experience of the low-level actions (*e.g.*, perceived ease of use).

Over the last three decades, the "perceived usefulness" scale from the Technology Acceptance Model (TAM) by Davis (1989) has been widely applied to measure technology usefulness. Extensions to the model (Venkatesh and Davis, 2000; Venkatesh and Bala, 2008) have been made to explain perceived usefulness in terms of more contextual factors, such as job relevance — *the degree to which an individual believes that the target system is applicable to his or her job*. While perceived usefulness, as measured by this scale, may be useful for comparing variations of a system or for benchmarking against a baseline, it offers limited insight for envisioning or characterizing the actual usage scenarios. Thus, in Chapter 5, we gain an initial understanding of the perceived usefulness of the *Periscope* system by characterizing its potential context of use. The findings in Section 5.4 are synthesized from participant interviews and observations of system usage, and enriched by the study team's reflections arising from these discussions. This interpretive methodology yields meaningful contributions for future research endeavors.

In addition to providing insights about potential utility, this study also revealed new usability issues not identified in the previous study described in Chapter 4. For instance, the more expansive workspace compared to the previous user study exacerbated the *split-attention effect* highlighted in Section 4.5. The local user was more mobile within the workspace (unlike the mostly seated participant in the previous study) and seldom looked at the interface on the laptop. Displaying the interface on a tablet or a smartphone would likely have been more beneficial in this context. Our future work will involve an analysis of the dataset for usability issues.

Towards Formalization

The iterative design process could be formalized in future work using methods such as systematic design documentation and reflection (Dalsgaard and Halskov, 2012), better connecting design decisions with broader design theory (Mose Biskjaer and Halskov, 2014), utilizing frameworks for planning and evaluating design documentation (Bardzell et al., 2016), and adopting documentation practices for building a "credible evidence base" for scholarly reporting (Sadokierski, 2020). These practices could better facilitate the translation of knowledge gained from the system's design into broader academic knowledge.

Evaluating the use and utility of new technologies is challenging. Comparative benchmarks can help in formalizing these evaluations. With a baseline system such as *Periscope* and its initial evaluation data, variations of the system can now be implemented to isolate specific parts of interest. I believe these evaluations should be conducted under as natural conditions as possible. For instance, to understand user behavior in the presence or absence of a feature, I suggest conducting an ablation-style study by simply disabling the feature and informing users that the system is experiencing bugs, rather than explicitly setting up separate conditions for usage with and without the feature.

Additionally, I do not recommend making direct quantitative comparisons with other work paradigms (*e.g.,* in-person or extended reality groupware) in the initial phases of exploring the design space. Such comparisons may not accurately capture the future value of emerging technologies and could potentially underestimate their transformative impact. New technologies often serve different purposes or are utilized in different contexts compared to existing ones. Without independently characterizing the new technology, we may not be able to develop fair metrics that consider the unique advantages or constraints of each work paradigm. Users might also be more familiar with existing paradigms, which may bias results in their favor due to comfort rather than quality. Therefore, it would be important to thoroughly familiarize users with new paradigms before drawing precise conclusions from comparisons. For example, in the study described in Chapter 5, we observed that instructors established new social norms and work practices over repeated sessions of using the *Periscope* system. We noticed phenomena such as small talk emerging over time, which may indicate increased familiarity with the paradigm. Thus, early comparisons to existing paradigms may obscure how innovative features of new technologies could lead to new ways of working that are not currently possible.

My reflections on these issues have led me to hypothesize that new computational representations of user and system behavior are necessary to help formalize the design and evaluation of robotic groupware. I address this in Section 6.4, where I present some preliminary work on a Petri net-based representation.

6.2 Why a Robot?

Implementing robotic camera systems, especially with the use of collaborative robots as discussed in this dissertation, demands significant investment in hardware infrastructure, considerable effort from users to learn and operate a complex system, and extensive resources for continuous development and maintenance. This raises the question: *Why incorporate robots?* Based on the work in this dissertation, I propose categorizing scenarios where robotic cameras are beneficial by examining them through two key dimensions: the level of *awareness* needs and the level of *structure* in the task or environment.

I hypothesize that a solution like *Periscope*, which offers substantial control and flexibility of viewpoint to users, is well-suited for scenarios that are more unstructured and have high awareness needs. Conversely, in more structured scenarios that also demand high awareness, increased automation in awareness support may be feasible. For instance, if a task model can predict awareness needs based on historical data from collaborative work, the robot could proactively gather relevant information for the collaborators. In unstructured scenarios, like those explored in this dissertation, where awareness needs may not be easily codifiable, users could benefit from having greater control for independent perceptual exploration.

In contexts with lower awareness needs, simpler robotic solutions, or even the absence of a robot, might be preferable. For more unstructured scenarios with low awareness needs, basic mobility affordances, such as those from mobile telepresence robots or PTZ cameras, might suffice. In more structured scenarios, the simplest solutions, like a single or multi-camera setup, might be effective.

In this dissertation, my argument is that the remote expert scenario, characterized by its unstructured, dynamic tasks and environment, and high awareness needs, aligns well with the viewpoint flexibility offered by cobots.

Beyond Robotic Cameras

A cobot's role in robotic groupware can extend beyond serving as an actuation technology for cameras. The robot may take on a more proactive role and function as a *teammate*, for example, by monitoring worker actions during periods of helper inattention (caused by factors such as distraction, interruption, or assisting other workers). The robot's actions can go beyond supporting shared visual information for collaborators and include providing *physical assistance* to the worker. Remote users could also use the robot for *remote manipulation or demonstration*. Additionally, integrating *other sensors* onto a cobot platform, such as Lidar sensors to generate virtual maps, opens up new user interaction possibilities. For example, remote users can navigate and interact with a virtual reconstruction of the workspace using head-mounted displays.

These additional roles of the robot could also potentially alleviate cost concerns, as the robot serves multiple functions. Cobots are marketed to manufacturing facilities for their versatility and adaptability for various types of tasks, making them an appealing choice for implementing robotic groupware in environments that demand complex, multi-faceted solutions. Additionally, depending on the precision and repeatability required from the robot, there are cheaper alternative robots that can serve as platforms for robotic groupware. For example, Elephant Robotics[2] has a range of affordable robots that could serve as desktop assistants in an educational institution.

Egido (1988) in her essay on *Video conferencing as a technology to support group work: A review of its failures* emphasizes that the success of new technologies depends more on the nature of the application for which they are introduced rather than the specific details and features of the system. Therefore, work such as that described in Chapter 5 is crucial for understanding the applications for which cobot-based groupware solutions may be beneficial.

Robotic Groupware

In this dissertation, I propose the term *robotic groupware* to describe groupware developed with robotic technology. My working definition for the term is: *Robotic systems that are specifically designed with the goal of being placed within groups for improving human-human collaborative work.*

[2] `https://shop.elephantrobotics.com/collections/mycobot`

6.3 Challenges and Limitations

In this section, I consolidate the various limitations and challenges in this work into three broad categories that I believe represent general challenges in developing any effective robotic groupware: *arbitration*, *evaluation*, and *development*.

Arbitration

In *Periscope*, the modes arbitrated inputs from different sources in a relatively simple 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. Some of these nuances were already observed in the use patterns discussed in Section 4.4.

In UP-5, related to the use of the *robot-led mode*, workers often rested their hand on the table to maintain a steady view of the relevant area. This behavior does not qualitatively align with the notion of being robot-led, but rather worker-led. Although the modes were named based on the arbitration policy, the practical use showed that in many instances of hand-tracking (what we called the *robot-led mode*), it was actually worker-led. Conversely, in UP-6, which is also related to the use of the *robot-led mode*, the worker did not explicitly direct the camera and this aligns more closely with the notion of being robot-led. This discrepancy suggests that in refining the arbitration policy, it may be necessary to differentiate these modes further by taking into account the varying levels of worker involvement observed in these two use patterns.

Additionally, consider UP-10, where the helper attempts and fails to move the camera on their own and then asks the worker for assistance. This is the *helper-initiated worker-led mode*, which does not quite feel like the worker is leading but rather assisting. An accurate usage of the *worker-led mode* would be UP-7 when the worker wants to share some information with the helper. Given the frequent observation of the assistance scenario (UP-10) in Chapter 5, we use the term *manual assistance* instead of *worker-led mode* to better reflect the qualitative difference in the same mode's usage. This suggests that the arbitration for these two types of mode usage might need to be different. Thus, one of the design implications suggested in Section 4.5 is a use pattern-based arbitration approach, which might address the nuances in how the same mode is used in different contexts.

In Section 4.5, we also suggest expertise-based arbitration and modeless arbitration to address other observed issues related to arbitration. It is important to note that the arbitration policy of the system coordinates users in a multi-user setup and influences their interactions. For instance, in our design, the helper had excessive authority during arbitration and this diminished the role of the worker. This led to conflict situations where the worker's attempts to move the camera or offers to do so were overruled by the helper.

I contend that the proper design of arbitration that carefully considers the needs of the context of use (*e.g.*, user, task, scenario, time and space) is the key to building *any* robotic groupware. The work in this dissertation lends some insight into this design process for robotic *camera* systems, but variations in context between Chapters 4 and 5 already show differences in how the arbitration policy functions under different contexts. Thus, the design of arbitration presents a significant challenge to building effective robotic groupware.

Evaluation

There were limitations in our study design in Chapter 4. Although we 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. Nonetheless, we recognized that experts who frequently utilize the system might provide more insight into the challenges they face day-to-day, use our system differently, and provide different feedback. Additionally, 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 and the worker was mostly seated. 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, we were interested in understanding how our system would perform in more dynamic workspaces.

We addressed both of these limitations in the study design in Chapter 5, by recruiting real-life instructors who participated in multiple sessions and instructed a realistic task in a more expansive workspace. This setup required the worker to move around considerably. We observed many differences in user behavior between the studies. For instance, annotation usage dropped from an average of 32 times per session in Chapter 4 to only 0.17 times per session in Chapter 5. This change can at least be partially attributed to the students' mobility and their familiarity with task-relevant terminology. These findings illustrate that even minor variations in study design can lead to significant behavior changes. However, pinpointing the exact reasons for these changes is challenging. While the quantitative data on annotation usage clearly differ, attributing these differences to specific aspects of the study design is complex.

Controlled studies may be valuable for isolating specific variables for detailed examination, but there is a risk that overly controlled studies might not reveal these behaviors at all. Further, identifying the right metrics to understand user behavior and system use is challenging because the success of collaboration outcomes is influenced not only by the effectiveness of the groupware system in facilitating collaboration but also by various group and organizational dynamics. For example, factors such as the similarity among group members, familiarity with one another, and the presence of hierarchical structures within the group can exert a substantial impact on collaborative outcomes (Harris et al., 2019). Further, these social dynamics may evolve over time (Twidale et al., 1994) and may be reshaped by the use of the groupware (Neale et al., 2004).

Analyzing data from opportunistic evaluations, where participants interact freely with the system and each session varies, is particularly challenging. While such data provides rich insights into the naturalistic usage of the system, determining which combination of data channels (*e.g.,* dialogue, interactions with the system, worker actions, camera motions) captured the phenomenon of interest is difficult. Another significant challenge lies in determining the appropriate granularity of data analysis. Aggregated data may condense information that is vital for a nuanced understanding of the system's design. On the other hand, a highly granular analysis can lead to an overwhelming amount of data and make it difficult to extract meaningful conclusions. Finally, translating these empirical insights into concrete improvements in the design of arbitration is challenging, especially when the origins of observed behaviors within the system are not clear.

Development

The development of the *Periscope* system presented numerous challenges, many of which are commonly encountered in systems research in HCI and HRI. These challenges include balancing research and engineering goals, sustaining systems post-publication, and securing the necessary resources, such as time and personnel, to achieve these aims. One strategy I employed to maintain the system beyond its initial publication was to utilize it as a platform for other research. In the period between the two studies in this dissertation, two short papers (Praveena et al., 2023a; Meng et al., 2023) were authored using the system as a research/engineering platform. Regular engagement with the system's code base facilitated more efficient issue resolution by the research team. However, replicating the system on a different robot has proven challenging due to the backend code being somewhat disorganized, a byproduct of prioritizing research over engineering goals.

Furthermore, a significant challenge, as alluded to in the previous subsection, involves bridging the gap between evaluation and improved system design. Translating empirical findings into tangible improvements in the arbitration policy is difficult, especially when the origins of observed behaviors within the system are not clear. Additionally, introducing new features can unpredictably interact with existing ones, making even minor implementation changes daunting in large-scale system development. This complexity was the primary reason for my decision to forgo the development of a second, more user-friendly version of the system. Instead, I opted to investigate alternative approaches to system behavior representations with the aim to facilitate the iterative development of complex interactive systems. I will elaborate on this work in Section 6.4.

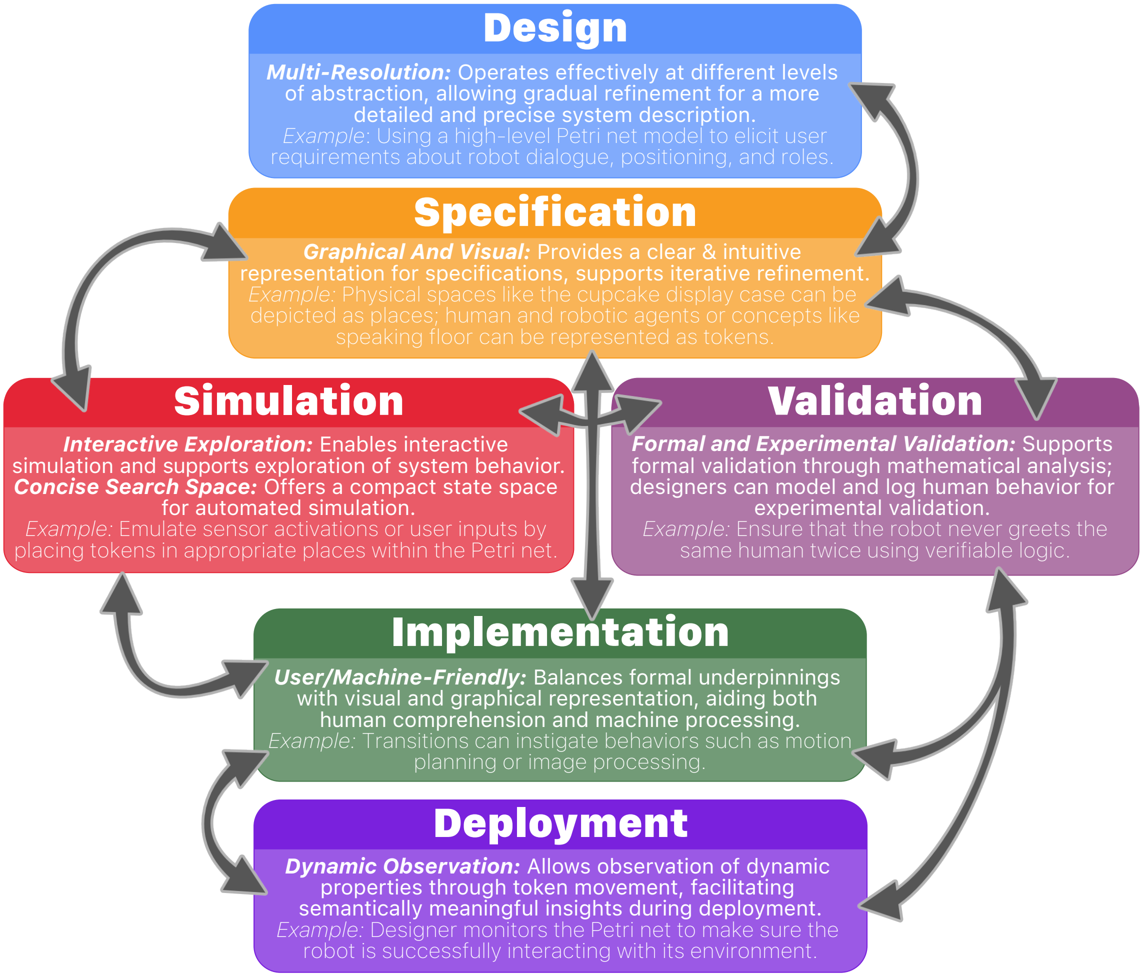
6.4 Future Work

Supporting Iterative Development

As discussed in Section 6.3, the development process of robotic groupware can present significant challenges. I will discuss some of my preliminary prior work in this direction that is motivated by the challenges discussed above related to arbitration, evaluation, and development. This section includes research that was previously published in Praveena et al. (2023b).

The development of complex systems such as robotic groupware draws on the expertise of a diverse array of *application developers*, including domain experts, designers, programmers, and research professionals, each employing methods, tools, and representations tailored to their specific stage of the development cycle. Additionally, given the iterative nature of systems development, there is often a significant amount of refinement and cross-communication between these phases. The representations and tools that application developers utilize for their respective phases of system development — such as low-fidelity prototypes by designers and high-fidelity simulators by algorithm developers — may not be effectively interoperable across different phases. In Praveena et al. (2023b), we proposed that a unifying representation can facilitate a smoother transition across the various stages of the iterative development process.

Petri nets, particularly Timed Colored Petri nets (TCPNs), hold the potential to unify various phases of the development process. Petri nets are a visual and graphical mathematical model that can be used to describe and analyze system behavior (Peterson, 1977). At a very basic level, Petri nets are defined as a set of *place* nodes that hold *tokens*, *transition* nodes that indicate actions that move tokens from place to place, and directed *arcs* between places and transitions that define the logic of where and which tokens are consumed and produced by transitions. Our paper provides more details on TCPNs (Praveena et al., 2023b). While representations and tools tailored for individual phases are indispensable, an accompanying, unified representation can enhance this design process by encoding information that can be used to coordinate between phases.



Design

Multi-Resolution: Operates effectively at different levels of abstraction, allowing gradual refinement for a more detailed and precise system description.
Example: Using a high-level Petri net model to elicit user requirements about robot dialogue, positioning, and roles.

Specification

Graphical And Visual: Provides a clear & intuitive representation for specifications, supports iterative refinement.
Example: Physical spaces like the cupcake display case can be depicted as places; human and robotic agents or concepts like speaking floor can be represented as tokens.

Simulation

Interactive Exploration: Enables interactive simulation and supports exploration of system behavior.
Concise Search Space: Offers a compact state space for automated simulation.
Example: Emulate sensor activations or user inputs by placing tokens in appropriate places within the Petri net.

Validation

Formal and Experimental Validation: Supports formal validation through mathematical analysis; designers can model and log human behavior for experimental validation.
Example: Ensure that the robot never greets the same human twice using verifiable logic.

Implementation

User/Machine-Friendly: Balances formal underpinnings with visual and graphical representation, aiding both human comprehension and machine processing.
Example: Transitions can instigate behaviors such as motion planning or image processing.

Deployment

Dynamic Observation: Allows observation of dynamic properties through token movement, facilitating semantically meaningful insights during deployment.
Example: Designer monitors the Petri net to make sure the robot is successfully interacting with its environment.

Figure 6.1: Flow between development phases and properties and examples of how Petri nets are useful at each.

In Praveena et al. (2023b), we found it useful to think of systems development in terms of six phases as shown in Figure 6.1: (1) *design*, (2) *specification*, (3) *simulation*, (4) *validation*, (5) *implementation*, and (6) *deployment*.

Design: A key property that enables Petri nets to enhance this phase is their ability to be used at different levels of abstraction. Jensen and Kristensen (2009) used a suitably abstracted Petri net-based model to engage end-users (hospital nurses) for elicitation, negotiation, and agreement of user requirements. This high-level model can then be gradually refined through the various systems development phases to yield a more detailed and precise description of the system under consideration, ultimately resulting in an executable prototype of the system.

Specification: Specification involves defining the requirements, behavior, features, and constraints of the system being developed. Petri nets provide a precise, concise, and human-friendly way to represent system behavior, yielding models that are both mathematically sound and accessible to systems developers through their visual and graphical nature (Jensen and Kristensen, 2009). The Petri net representation can be designed to map to semantically meaningful domain properties (*e.g.,* distinct physical areas within a setting, human and robotic agents, and abstract concepts like speaking floor). For instance, physical spaces like tables can be depicted as *places*, and the robot can be represented by a *token*. The movement of the robot can be specified and visualized as the movement of the *token* through the Petri net. Additionally, Petri nets can facilitate the development of complex systems by enabling developers to specify smaller subprocesses, which can then be automatically combined using mathematical properties into a behavior model of a larger and more intricate system.

Simulation: Simulation involves viewing how a system behaves under different conditions via a simplified representation of real-world phenomena. Developers have the ability to actively engage with a Petri net model to explore hypothetical scenarios, *e.g.,* by placing *tokens* to emulate real-world events such as sensor activations and user commands. Due to the visual and graphical nature of the representation, developers can visually track the movement of *tokens* as they traverse the graph. In addition, Petri nets can also facilitate the methodical exploration and discovery of optimal system parameters via the use of automated simulations that do not require user intervention.

Validation: Validation involves assessing whether the system is built correctly according to its design and performs its intended functions accurately and reliably. Robotic groupware systems can benefit from both *formal* and *experimental* validation techniques. Formal techniques can be used to prove whether programs adhere to specific properties (Wing, 1990), such as ensuring a robot never greets the same human twice (Porfirio et al., 2018). On the other hand, experimental techniques are useful for validation of user behavior and ergonomic properties (Ait-Ameur and Baron, 2006), such as ensuring that an individual is able to effectively communicate with the robot in the presence of ambient noise. Petri nets are *formal* models that can be shared between formal and experimental validation, and bridge the gap between these techniques.

Implementation: The *Design*, *Specification*, *Simulation*, and *Validation* stages of the systems development process produce a precise, concise, and human-friendly model in the form of a Petri net, which can be integrated into the system's implementation. This integration eliminates any loss of information or unintended alterations that might occur during translations between different representations at various phases. As a result, the specified behaviors and interactions captured within the Petri net model can be faithfully translated into the program's execution. If, at any point, the designer realizes changes need to be made to the implementation, since the model is still represented as a Petri net, this can seamlessly be introduced into the previous *Specification*, *Simulation*, and *Validation* steps.

Deployment: Petri nets can be beneficial to monitor and visualize the real-time execution of finished systems. The compact and graphical nature of Petri nets provides a clear depiction of the system flow, especially with the use of tokens to model dynamic behaviors and allow for tracing an entity through the system. Petri nets are ideal for run-time analysis and experimental validation during deployment because they model complex interactions, capture dynamic behaviors, and provide a visual presentation of system behavior.

Complex systems are characterized by many free interactions, leading to elaborate interaction traces that are challenging for analysis and interpretation by people. Muratet et al. (2016) use a Petri net representation to *algorithmically* analyze and label player behaviors during a game. However, even for the *manual* generation of labels, qualitative coding of Petri net traces could serve as a complement to, or even a potential substitute for, video coding. Moreover, Petri net elements could be time-stamped or tagged with data for dynamic behavior (*e.g.,* when transitions are fired) to facilitate downstream analysis such as understanding response times, waiting times, and throughput. This approach could enable a more direct translation of empirical insights from deployment into tangible improvements in the design, specification, or implementation of system behavior.

The efficacy of this approach hinges on the model effectively capturing all aspects that the developer seeks to analyze. This is achievable using Petri nets because of the representation's versatility in expressing a variety of concepts, including user-activity model, task model, and context model.

Current and Future Work: Petri nets could be an indispensable tool for robotic groupware development. However, there is a need for a domain-specific instantiation of Petri nets that is tailored for HRI/groupware development, and is accessible to developers lacking experience with Petri nets. There is also a need for development tools that selectively expose aspects of the underlying representation to developers in an intuitive, graphical way. Currently, I am working with a team of undergraduate and graduate researchers on the development of a new tool, named *Statewise*, specifically towards this goal.

Extensions of the System

Improving robot initiative: In the *Periscope* system, the robot exhibited passive behaviors to assist and facilitate user interactions, but did not initiate actions. This measured approach in our preliminary prototype prioritized safety for the worker from any possible collisions with the robot and minimized the potential for discomfort from unexpected robot actions. The inherent risks and uncertainties of using a powerful robotic arm, which could be remotely controlled in close proximity to a worker, necessitated a more cautious and passive role for the robot. However, as these systems evolve and gain acceptance, there is immense potential for the robot to take on a more proactive role and function as a *teammate*. For example, the robot may monitor worker actions during periods of helper inattention (caused by factors such as distraction, interruption, or assisting other workers) and provide task summaries to the helper upon their return.

Additionally, the robot may serve different roles during the collaboration process as a tool (*e.g.,* a camera holder in Gurevich et al. (2012)), a surrogate (*e.g.,* representing the helper's gaze in Kuzuoka et al. (1995)), or a collaborator (*e.g.,* completing a task on behalf of the helper in Stolzenwald and Mayol-Cuevas (2019)). The robot can fulfill these roles for the remote user, the local user, or both.

Expanding robot functions: The robot's actions can go beyond supporting shared visual information for collaborators and include providing physical assistance to the local user. For example, Senft et al. (2021a) developed an approach that enables end-users to interactively create robot programs to collaboratively complete a task with a robot. In this context, the robot serves multiple functions by providing both visual and physical assistance to users. Sophisticated arbitration policies need to be developed to address the competing needs associated with these functions, for example, when the remote user needs to move the robot to *see* something while the local user needs the robot to *fetch* something. Additionally, remote users could also use the robot for remote manipulation or demonstration. This allows remote users not only to observe but also to interact with the physical workspace. However, these manipulation actions by the robot, especially when performed in close proximity to local users, raise safety concerns that must be addressed.

Additionally, integrating other sensors and displays onto a cobot platform, such as Lidar sensors to generate virtual maps or projectors to overlay annotations directly onto the physical environment, opens up new user interaction possibilities. For example, remote users can navigate and interact with a virtual reconstruction of the workspace using head-mounted displays. Local users could use a projection-based tabletop interface for robot programming.

Exploring alternative collaboration setups: *Periscope* was designed for a specific *local worker–remote expert* configuration as described in Section 1.1, but future work could explore other configurations. In the study described in Chapter 5, we discussed several interesting alternative setups with participants. These alternatives include mutual collaboration, where both collaborators engage in physical tasks, and scenarios where experts provide remote assistance to multiple workers.

In the mutual collaboration scenario, both collaborators could be co-located with robotic cameras. They would have access to each other's robotic camera, thereby enabling them to observe and interact (if the robot is equipped with a manipulator) with each other's workspaces. For this setup, the interaction mechanisms of the *Periscope* system, which are currently designed for mouse input from a remote user, would need to be modified. A more suitable approach might involve using gesture or touch-based inputs that take advantage of the co-location of both users with their respective robots.

In scenarios where experts assist multiple workers across various locations (each worker is co-located with a robot), the expert might only be able to actively engage with one worker at a time. As a result, the robots could serve as the expert's deputy and autonomously monitor workers not directly supervised. This work paradigm imposes unique cognitive demands on the expert. Unlike guiding multiple workers who are physically present in the same space, assisting workers across different locations requires the expert to switch between contexts and rapidly acquire an awareness of each worker's progress.

Alternative setups might also involve changes in the expertise level of each collaborator. For example, in Normandale's use case, a *local expert* instructs a *remote student*, which is the opposite of our context. In Li et al. (2022), a *local expert* instructs several *remote students*. In Mentis et al. (2020), both collaborators are surgeons who are *experts*. In all examples, only the local user has access to the physical workspace. These alternative setups would likely result in variations in shared camera control, stemming from changes in the frequency and types of information that collaborators need to share, which vary according to the context.

There may also be situations where a complex problem requires multiple remote experts to simultaneously or intermittently interact with the worker. When multiple experts attempt to control a robot simultaneously, user inputs may need to be prioritized and combined based on factors such as predefined roles, expertise level, or the criticality of the action. If control shifts intermittently from one expert to another, then handoff protocols may be necessary to ensure knowledge transfer between experts. Such protocols might include easily accessible logs detailing past actions or sensor readings, and the system might even proactively prepare this information in anticipation of expert needs.

Introducing new modalities: The incorporation of robotic technology in groupware systems does not exclude the possibility of incorporating additional technologies. Introducing new modalities (*e.g.*, natural language (Tellex et al., 2020)) can enrich user interactions. Verbal communication is the primary modality of interaction between workers and experts. However, users might not want to use natural language to interact with the robot; they may choose to use a different modality with the robot while simultaneously employing natural language for communication with their human collaborators. Alternatively, they might opt for natural language as a unified modality of interaction with both the robot and their human collaborators. The choice of modality could depend on the robot's role: if the robot acts as a tool, users may favor mouse input, whereas if the robot functions as a collaborator, natural language could be more appropriate. Natural language could also be utilized implicitly to infer user intent and trigger autonomous behaviors.

The *Periscope* system uses screen interfaces for controlling and displaying shared information; however, there is potential to integrate head-mounted displays into this context. A virtual view of the workspace can enable interactive capabilities for both input and output. Moreover, users could transition between live video and the virtual workspace or experience a combination of the two. In addition to the feed from the robotic camera, users could also have access to a camera on a tablet or a smartphone that can be moved around by the worker, or to a head-mounted camera on the worker. The physical workspace may host multiple robots, allowing the user to access their feeds separately for varied viewpoints across a broader workspace. Alternatively, while the user temporarily accesses one feed, another robot could be utilized to scan the environment and update the dynamically changing details in a virtual setting. The use of multiple (dynamic) camera feeds may cause issues with the control frame — the frame of reference in which the user provides inputs (Praveena et al., 2022). This would require further research to understand its impact.

Integrating haptic (Hannaford and Okamura, 2016) and actuated tangible interfaces (Poupyrev et al., 2007; Ihara et al., 2023; Li et al., 2023) into robotic groupware opens new opportunities for enhancing user interaction. By incorporating tactile feedback, users can receive information about the force exerted during physical interactions. This could enable a remote user to be informed when the robot or worker is performing an action, or allow a remote expert to convey to a local worker the amount of force required for a specific task. Such feedback could be particularly valuable for an expert monitoring multiple workers, as it would allow for unobtrusive updates on the activities of peripheral workers. Finally, the use of tabletop tangible interfaces can potentially expand the supported range of tasks. For example, a coordinated use of miniature tabletop robots for precise wiring adjustments could be an alternative to relying solely on the cobot.

Improving arbitration: All the extensions discussed above necessitate more sophisticated arbitration policies. Shared control systems rely on two aspects: (1) understanding *user intent*, and (2) robust *arbitration policies* to combine user and autonomous behaviors to fulfill this intent. In the *Periscope* system, user intent is often explicitly provided, *e.g.,* for mode selection and target selection. However, there is potential for implicitly predicting user intent, such as by analyzing conversation data or gaze patterns (see Belardinelli (2023) for a recent review). Aronson et al. (2021) demonstrated the use of the remote user's natural eye gaze behavior to infer user intentions during telemanipulation, and Oh et al. (2021a) explored the use of natural grabbing hand gestures for inferring the user's intent to grasp an object.

Learning-based approaches for developing arbitration policies (Dragan and Srinivasa, 2013) are widely used in the robotics literature, and these strategies can also be applied to robotic groupware. For example, Oh et al. (2021b) presented a policy that granted increased control to the human operator at critical decision points, such as choosing among various routes to avoid an obstacle. In robotics research, shared control typically involves one human and one robot. The development of arbitration policies for *multi-user* shared control systems introduces significantly increased complexity.

Application to New Domains

The work presented in this dissertation focuses on physical tasks within the manufacturing domain. Robotic groupware holds the potential to facilitate new work practices in various other domains as well. I explored this in the domain of cinematography in prior work through exploratory interviews with cinematography practitioners (Praveena et al., 2023a). I will elaborate on one of our findings that is particularly relevant to this dissertation.

One participant described a collaborative rehearsal process where the camera operator and director work together in real-time to fine-tune camera movements until they achieve the desired shot. This interaction allows for immediate adjustments and the ability to save the final movement for use during the actual shoot. This is an example of *co-located, real-time* robotic groupware. Another participant proposed the idea of multiple individuals in different locations simultaneously accessing and operating a robotic camera to storyboard and prototype shots, possibly using virtual reality headsets. This is an example of *remote, real-time* robotic groupware.

Similar to the work in Chapter 5, in the cinematography work, we used *Periscope* as a technology probe which served as a starting point for brainstorming and exploring new ideas. This approach ultimately led to interesting discussions about innovative and alternative cinematography processes. Thus, the research methodologies and technical approaches presented in this dissertation are relevant for exploring new application domains in robotic groupware.

Building effective robotic groupware for new application domains presents challenges similar to those encountered in this work, especially in understanding the context of use for future work practices. Salovaara et al. (2017) call this the *present-future* gap and propose several mitigation strategies. One such strategy is *staging*, such as configuring a laboratory to resemble a living room. Carefully staged studies can represent high-fidelity versions of a potential context of use, which can elicit more natural behavior from participants.

6.5 Conclusion

This dissertation presents work on the design, usage, and application of the *Periscope* system to offer insights into a novel point in the design space of robotic groupware, specifically groupware built on cobot platforms and their potential for enabling remote collaborative work. *Periscope* is a robotic camera system that is a synergistic integration of various methods and techniques from HCI, CSCW, HRI, and robotics that demonstrates the feasibility of building a groupware system on cobot platforms. This dissertation contributes to the understanding of how and where a highly capable cobot platform can be integrated into a collaborative system. My hope is that the use and application of *Periscope* in real-world contexts demonstrates the potential for new paradigms of collaborative work, such as remote workforce training. My belief is that robots, particularly collaborative robots, have immense potential to help people work together better on physical tasks. While there is plenty of exploration to be done to deepen our understanding of this design space, my dissertation begins to establish a vocabulary and framework for more general-purpose robot-supported collaborative work and lays the groundwork for how we can build effective robotic groupware.

A IMPLEMENTATION DETAILS

Figure A.1 illustrates the system architecture. The system is built on the Robot Operating System (ROS)[1], which enables communication between system components and real-time control of the robot arm. In our prototype, we mount an Azure Kinect camera[2] on a Universal Robot UR5 collaborative robot arm[3]. The camera provides both color images, which the users can view, and depth data for use in computer vision algorithms. The color and depth data have a resolution of 2048x1536 and 512x512, respectively. The resolution of the color image viewed by the user may be reduced due to bandwidth constraints.

We used the React framework[4] for the front-end interface and it connects to the back-end ROS server using `roslibjs`[5], which uses WebSockets to connect with `rosbridge`[6]. The video feed from the robot-mounted camera is streamed in real-time to both the helper's and worker's interfaces using `web_video_server`[7]. Visual feedback on the camera feed for input commands and annotations is implemented using React Conva[8]. The 3D view, built on `ros3djs`[9], includes a digital twin of the robot and its surrounding objects in `threejs`[10] and updates their states in real-time from the back-end ROS server.

We integrated a third-party video conferencing API from `Dolby.io`[11] to host real-time video conferencing. Both the helper and the worker automatically join the video conference after logging in to the system. Similar to commercial video conferencing tools, users have the options to turn on or off their microphones or

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

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

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

[4] https://reactjs.org/

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

[6] https://wiki.ros.org/rosbridge_suite

[7] https://wiki.ros.org/web_video_server

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

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

[10] https://threejs.org/

[11] https://dolby.io/

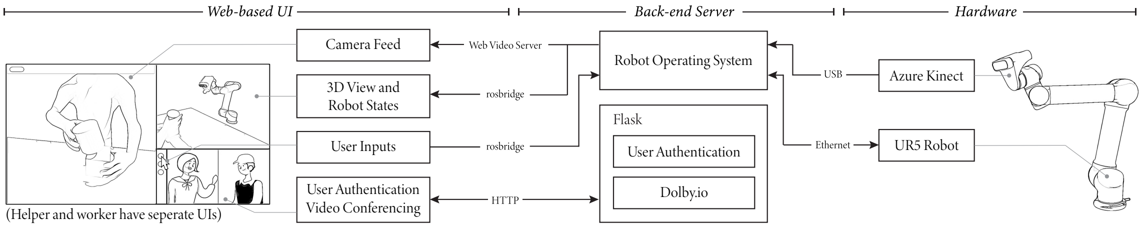


Figure A.1: *Periscope* includes separate web-based UIs for each collaborator and a back-end server that controls a robot-mounted camera.

camera feeds. The control panel consists of buttons that interact with the ROS backend. The interface is accessible at `periscope.cs.wisc.edu` and has a password-based authentication and authorization mechanism that only allows authorized users to access the system and control the robotic camera.

Motion Generation

We cast the real-time motion generation problem in a constrained multiple-objective optimization structure, as in Rakita et al. (2021). Most user interactions and autonomous behaviors are formulated as objectives.

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

Here, $\mathbf{q} \in \mathbb{R}^n$ is the configuration of an 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. (2017) that normalizes objective values for accommodating multiple-objectives.

The forward kinematics function $\mathbf{\Psi}$ calculates the camera pose given a joint configuration. Forward kinematics functions $\mathbf{\Psi}_p(\mathbf{q}), \mathbf{\Psi}_R(\mathbf{q}), \mathbf{\Psi}_q(\mathbf{q})$ represent the position, rotation matrix, and quaternion of the camera for 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*.

Helper Interactions

Target

To point the camera towards a target, we adapt the "look-at" objective from prior work (Rakita et al., 2018).

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

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.

Adjust

To move the camera according to directional inputs, the objective is:

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

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

Reset

To move the camera to a pre-defined starting configuration, a pre-defined joint configuration is sent to the robot arm via the `servoj` command in URScript.

Annotate

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

Worker Interactions

Point

Pointing detection is built upon the open-source MediaPipe solution (Lugaresi et al., 2019)[12], in which a hand pose is represented by 21 2D landmarks. To detect a pointing gesture, 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 $\mathbf{t}$ is set to the position of the index fingertip in the robot frame.

Direct

As described in §Point, we detect landmarks of the worker's hand using MediaPipe. The landmarks are converted from the camera's frame of reference to the robot's frame of reference. The average position of 5 landmarks on the worker's right hand (wrist, base of all fingers) are used as the target $\mathbf{t}$.

Freedrive

The robot is switched to freedrive directly via URScript. In freedrive, the robot can be manually moved by the worker into a desired pose. 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 user.

Autonomous Behaviors

Keep distance

To maintain a specified distance between the camera and a target point, we use an objective from prior work (Rakita et al., 2018).

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

$$\chi_{\text{dist}}(\mathbf{q}) = ||\mathbf{t} - \boldsymbol{\Psi}_p(\mathbf{q})||_2 - d \tag{A.4}$$

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

Keep upright

We adapt an objective that keeps the camera upright from prior work (Rakita et al., 2018).

$$\chi_{\text{lookat}}(\mathbf{q}) = (\boldsymbol{\Psi}_R(\mathbf{q})[0,1,0]^\mathsf{T}) \cdot [0,0,1]^\mathsf{T} \tag{A.5}$$

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.

Track hand

Same as §Direct.

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 (Rakita et al., 2018) 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:

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

We use the same objective as prior work (Rakita et al., 2018) to minimize the velocity of the camera.

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

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

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

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

Avoid collisions

We use collision avoidance methods from prior work (Rakita et al., 2021) 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 (Rakita et al., 2021), 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 (Kenwright, 2015).

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

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

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 (Cao et al., 2021) system to extract human body poses from RGB images. Human body poses are represented as 25 keypoints in the RGB image, which we then map to 3D using depth data. Since the depth data can be noisy, we use a median filter to get smooth and stable body keypoints. With these stable 3D keypoints, we create convex hull spaces (*e.g.,* spheres, cuboids) around body parts 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 (Malyuta et al., 2020) to identify the poses of dynamic objects in the environment. In future iterations, other vision-based pose estimation technologies (*e.g.*, SSD-6D (Kehl et al., 2017)) could replace the AR tags.

B LESSON PLAN

The lesson plan for instructors in *Phase 2* of the study in Chapter 5 is included below.

I. Objectives

At the end of the session, students should be able to:

1. Strip wires and attach terminals/disconnects
2. Route wires, connecting speakers to the amplifier
3. Identify and fix problems with the audio setup

II. Subject Matter

Topic: Car radio and speaker assembly

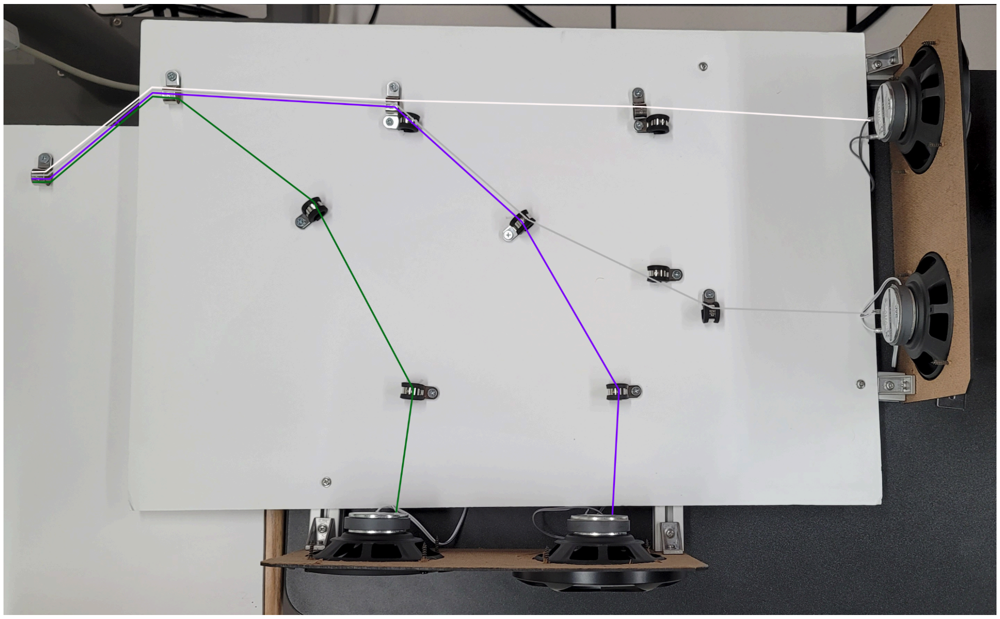
Tool and materials:

- Wire stripper
- Wire crimper
- Red/Blue/Yellow insulated terminal and disconnect
- 16 gauge wire
- Assorted fuses
- Fuse puller
- Philips head screwdriver
- Multimeter

Specific skills: wire stripping, wire crimping, troubleshooting

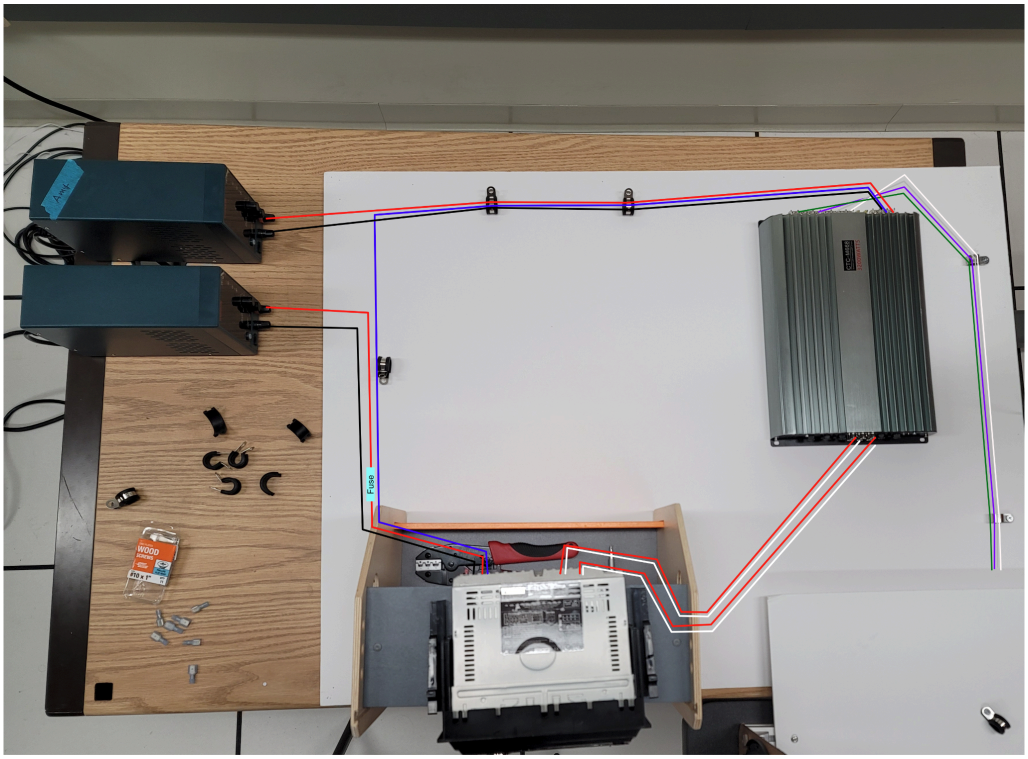
III. Setup and Wiring Diagrams

Wiring diagram for speaker setup:



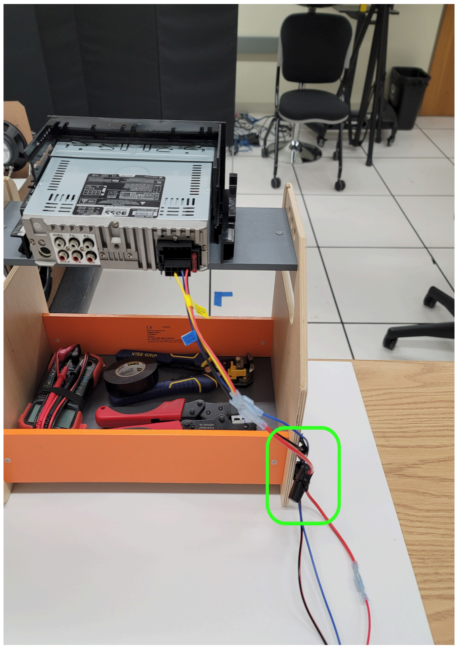
Marked in white is the positive/negative connection to the front left speaker. Marked in gray is the positive/negative connection to the front right speaker. Purple is the rear left. And finally green is the rear right.

Wiring for amplifier and radio:



Red and black on the left side of the image mark the power and ground wires to the radio and amplifier unit. In blue is marked the “power on” cable that connects the radio to the amplifier. Marked in red/white on the right side of the image, are the audio cables connecting the radio to the amplifier. The white, gray, purple, and green cables on the right side of the image mark the connections from the amplifier to the speakers, as noted in the prior image.

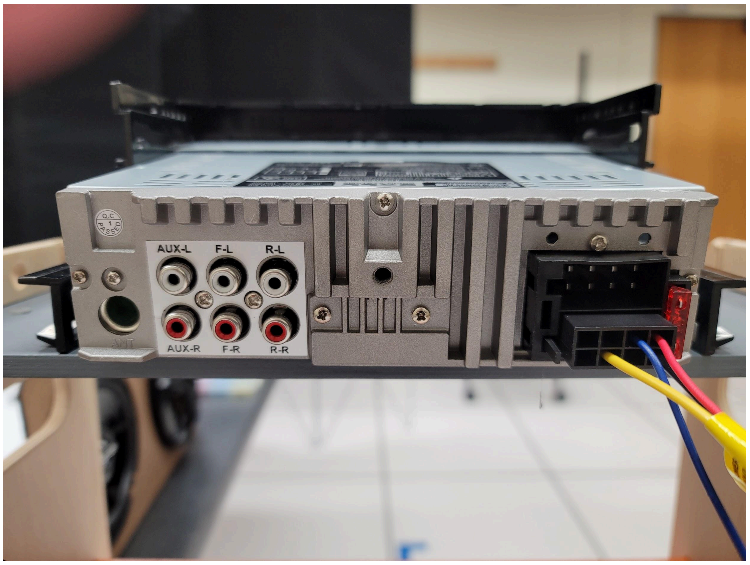
Fuse location on the power connection to the radio unit marked in green:

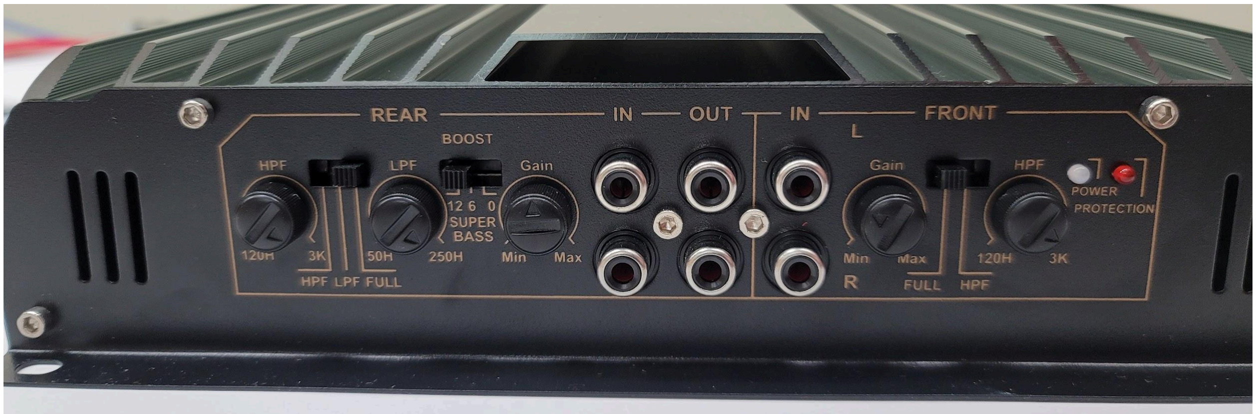


IV. Procedures

A. Students should have already completed:

a. Connected the radio to the amplifier





F-L and F-R on the back of the radio will connect to the top and bottom (respectively) on the FRONT-IN side of the amplifier (the rightmost set of inputs). While R-L and R-R on the back of the radio will connect to the top and bottom (respectively) on the REAR-IN side of the amplifier (the leftmost set of inputs).

b. Practiced stripping wires and crimping them to terminals/disconnects

c. Connecting a speaker to the amplifier

When you start the interaction, ask the student to explain/demo what they have done/learned so far.

B. Collaborative task activity:

a. For each of the remaining 2 speakers:

i. Have students run the wire through the harness based on the guide

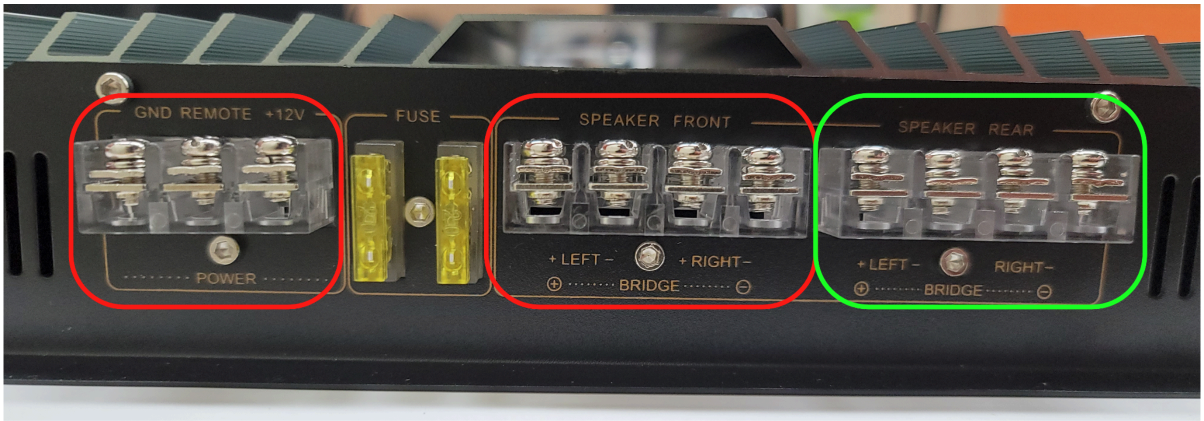
1. See **Section III**

ii. Have them cut and strip the ends of that wire

iii. Have then attach terminals/disconnect

iv. Plug the wires into the speakers and amplifier based on the diagrams in **Section III** and the image below

Speaker connections to the amplifier:



Marked in red should already be connected in advance and open/available connections are marked in green.

- **C.** Verify and troubleshoot the system
  - **a.** Ensure the system turns on
    - i. Have students power on the bench supply units, turn on the radio, and play a test song from their computer to the system.
  - **b.** Testing the system for audio quality and other potential issues
    - i. Ask the students about the quality of audio, as it may be difficult for you to discern remotely through the computer's microphone.

V. Troubleshooting

- A. The system does not turn on
  - a. Check if power is making its way through the system
  - b. Check if all connections are secure
  - c. Check if wires are damaged
  - d. Check if fuses are blown
- B. The audio quality is distorted
  - a. Check if connections are secure
  - b. Check if the amp is distorting the output
  - c. Check if the positive and negative wires are flipped for the speaker
  - d. Check if the amp is receiving enough current
- C. The audio signal is sporadic (cutting in and out)
  - a. Check if the connections are secure
  - b. Check if the amp is receiving enough current
  - c. Make sure the wires aren't damaged

VI. Assessment

Later, during the interview, we would like you to share with us your assessment about how the student **progressed** in their skills/knowledge and how **comfortable** and **confident** they seemed with the task.

REFERENCES

Abi-Farraj, Firas, 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)*, 4266–4273.

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

Adams, Marilyn Jager, Yvette J Tenney, and Richard W Pew. 1995. Situation awareness and the cognitive management of complex systems. *Human Factors* 37(1):85–104.

Adcock, Matt, 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. VRCAI '13, New York, NY, USA: Association for Computing Machinery.

Aditham, R.K., R. Jain, and M. Srinivasan. 1997. Interest based collaboration framework. In *Proceedings of IEEE 6th Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*, 75–80.

Ait-Ameur, Yamine, and Mickaël Baron. 2006. Formal and experimental validation approaches in HCI systems design based on a shared event B model. *International Journal on Software Tools for Technology Transfer* 8:547–563.

Akkil, Deepak, 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. CHI EA '16, New York, NY, USA: Association for Computing Machinery.

Andriessen, JH Erik. 2012. *Working with groupware: Understanding and evaluating collaboration technology*. Springer Science & Business Media.

Aronson, Reuben M, Nadia Almutlak, and Henny Admoni. 2021. Inferring goals with gaze during teleoperated manipulation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 7307–7314. IEEE.

Arthur Jr, Winfred, Winston Bennett Jr, Pamela L Stanush, and Theresa L McNelly. 1998. Factors that influence skill decay and retention: A quantitative review and analysis. *Human Performance* 11(1):57–101.

Autor, David H., Frank Levy, and Richard J. Murnane. 2003. The skill content of recent technological change: An empirical exploration. *The Quarterly Journal of Economics* 118(4):1279–1333.

Bai, Huidong, 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. CHI '20, New York, NY, USA: Association for Computing Machinery.

Banfield, Richard, C. Todd Lombardo, and Trace Wax. 2015. *Design sprint: A practical guidebook for building great digital products*. O'Reilly Media, Incorporated.

Bardzell, Jeffrey, Shaowen Bardzell, Peter Dalsgaard, Shad Gross, and Kim Halskov. 2016. Documenting the research through design process. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*, 96–107. DIS '16, New York, NY, USA: Association for Computing Machinery.

Beer, Jenay M, Arthur D Fisk, and Wendy A Rogers. 2014. Toward a framework for levels of robot autonomy in human-robot interaction. *Journal of human-robot interaction* 3(2):74.

Belardinelli, Anna. 2023. Gaze-based intention estimation: principles, methodologies, and applications in hri. *arXiv preprint arXiv:2302.04530*.

Bohren, Jonathan, Chavdar Papazov, Darius Burschka, Kai Krieger, Sven Parusel, Sami Haddadin, William L Shepherdson, Gregory D Hager, and Louis L Whitcomb. 2013. A pilot study in vision-based augmented telemanipulation for remote assembly over high-latency networks. In *2013 IEEE International Conference on Robotics and Automation*, 3631–3638. IEEE.

Braun, Virginia, and Victoria Clarke. 2012. Thematic analysis. In *APA Handbook of Research Methods in Psychology, Vol. 2. Research Designs: Quantitative, Qualitative, Neuropsychological, and Biological*, ed. Harris Cooper, Paul M. Camic, David L. Long, A. T. Panter, David Rindskopf, and Kenneth J. Sher, 57–71. American Psychological Association.

Brown, S. 2020. What Microsoft's Satya Nadella thinks about work of the future. `https://mitsloan.mit.edu/ideas-made-to-matter/what-microsofts-satya-nadella-thinks-about-work-future`.

Bruner, Jerome. 1995. From joint attention to the meeting of minds. In *Joint Attention: Its Origins and Role in Development*, ed. Carolyn Moore and Patrick Dunham. Hillsdale, N.J.: Erlbaum.

Burmeister, Anne, and Jürgen Deller. 2016. Knowledge retention from older and retiring workers: What do we know, and where do we go from here? *Work, Aging and Retirement* 2(2):87–104.

Cao, Z., G. Hidalgo, T. Simon, S. Wei, and Y. Sheikh. 2021. OpenPose: Realtime multi-person 2D pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 43(01):172–186.

Chaumette, François, Seth Hutchinson, and Peter Corke. 2016. Visual servoing. In *Springer Handbook of Robotics*, ed. Bruno Siciliano and Oussama Khatib, 841–866. Cham: Springer International Publishing.

Chen, Jessie YC, Ellen C Haas, and Michael J Barnes. 2007. Human performance issues and user interface design for teleoperated robots. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 37(6):1231–1245.

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

Christensen, Clayton M. 2013. *The innovator's dilemma: When new technologies cause great firms to fail*. Harvard Business Review Press.

Christianson, David B., Sean E. Anderson, Li-wei He, David H. Salesin, Daniel S. Weld, and Michael F. Cohen. 1996. Declarative camera control for automatic cinematography. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 1*, 148–155. AAAI'96, AAAI Press.

Christie, Marc, Patrick Olivier, and Jean-Marie Normand. 2008. Camera control in computer graphics. vol. 27, 2197–2218.

Clark, Herbert H., and Catherine R. Marshall. 1981. Definite knowledge and mutual knowledge. In *Elements of Discourse Understanding*, ed. Aravind K. Joshi, Bonnie L. Webber, and Ivan A. Sag, 10–63. Cambridge, UK: Cambridge University Press.

Crasborn, Onno, 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.

Dalsgaard, Peter, and Kim Halskov. 2012. Reflective design documentation. In *Proceedings of the Designing Interactive Systems Conference*, 428–437. DIS '12, New York, NY, USA: Association for Computing Machinery.

Daly-Jones, Owen, 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):21–58.

Davis, Fred D. 1989. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly* 319–340.

DeLong, David W. 2004. *Lost knowledge: Confronting the threat of an aging workforce*. Oxford University Press.

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

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

Egido, Carmen. 1988. Video conferencing as a technology to support group work: a review of its failures. In *Proceedings of the 1988 ACM Conference on Computer-Supported Cooperative Work*, 13–24. CSCW ’88, New York, NY, USA: Association for Computing Machinery.

Ellis, Clarence A., Simon J. Gibbs, and Gail Rein. 1991. Groupware: Some issues and experiences. *Commun. ACM* 34(1):39–58.

Endsley, Mica R. 1995. Toward a theory of situation awareness in dynamic systems. *Human Factors* 37(1):32–64.

Ens, Barrett, Joel Lanir, Anthony Tang, Scott Bateman, Gun Lee, Thammathip Piumsomboon, and Mark Billinghurst. 2019. Revisiting collaboration through mixed reality: The evolution of groupware. *Int. J. Hum.-Comput. Stud.* 131(C): 81–98.

Fakourfar, Omid, Kevin Ta, Richard Tang, Scott Bateman, and Anthony Tang. 2016. Stabilized annotations for mobile remote assistance. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 1548–1560. CHI ’16, New York, NY, USA: Association for Computing Machinery.

Feick, Martin, 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. CHI ’18, New York, NY, USA: Association for Computing Machinery.

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

Fussell, Susan R., 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*, 21–30. CSCW ’00, New York, NY, USA: Association for Computing Machinery.

Fussell, Susan R., Leslie D. Setlock, and Robert E. Kraut. 2003a. 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*, 513–520. CHI ’03, New York, NY, USA: Association for Computing Machinery.

Fussell, Susan R., Leslie D. Setlock, and Elizabeth M. Parker. 2003b. Where do helpers look? Gaze targets during collaborative physical tasks. In *CHI ’03 Extended Abstracts on Human Factors in Computing Systems*, 768–769. CHI EA ’03, New York, NY, USA: Association for Computing Machinery.

Fussell, Susan R., Leslie D. Setlock, Elizabeth M. Parker, and Jie Yang. 2003c. 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. CHI EA ’03, New York, NY, USA: Association for Computing Machinery.

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

Gauglitz, Steffen, 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*, 241–250. MobileHCI ’12, New York, NY, USA: Association for Computing Machinery.

Gauglitz, Steffen, 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. VRST '14, New York, NY, USA: Association for Computing Machinery.

Gaver, William W. 1992. The affordances of media spaces for collaboration. In *Proceedings of the 1992 ACM Conference on Computer-Supported Cooperative Work*, 17–24. CSCW '92, New York, NY, USA: Association for Computing Machinery.

Gaver, William W., 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*, 335–341. CHI '93, New York, NY, USA: Association for Computing Machinery.

Geertz, Clifford. 2008. Thick description: Toward an interpretive theory of culture. In *The Cultural Geography Reader*, 41–51. Routledge.

Giusti, Leonardo, Kotval Xerxes, Amelia Schladow, Nicholas Wallen, Francis Zane, and Federico Casalegno. 2012. Workspace configurations: Setting the stage for remote collaboration on physical tasks. In *Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design*, 351–360. NordiCHI '12, New York, NY, USA: Association for Computing Machinery.

Gleicher, Michael, and Andrew Witkin. 1992. Through-the-lens camera control. In *Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques*, 331–340. SIGGRAPH '92, New York, NY, USA: Association for Computing Machinery.

Greenberg, Saul, and Bill Buxton. 2008. Usability evaluation considered harmful (some of the time). In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 111–120. CHI '08, New York, NY, USA: Association for Computing Machinery.

Greif, Irene. 1988. *Computer-supported cooperative work: A book of readings*. Morgan Kaufmann Publishers Inc.

———. 2019. How we started CSCW. *Nature Electronics* 2(3):132–132.

Grudin, J. 1994. Computer-supported cooperative work: History and focus. *Computer* 27(5):19–26.

Gupta, Kunal, 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):2413–2422.

Gurevich, Pavel, 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. CHI '12, New York, NY, USA: Association for Computing Machinery.

Gutwin, Carl, and Saul Greenberg. 2002. A descriptive framework of workspace awareness for real-time groupware. *Comput. Supported Coop. Work* 11(3):411–446.

Hagenow, Michael, Emmanuel Senft, Robert Radwin, Michael Gleicher, Bilge Mutlu, and Michael Zinn. 2021. Corrective shared autonomy for addressing task variability. *IEEE Robotics and Automation Letters* 6(2):3720–3727.

Hannaford, Blake, and Allison M Okamura. 2016. Haptics. *Springer Handbook of Robotics* 1063–1084.

Hansen, Nicolai Brodersen, Christian Dindler, Kim Halskov, Ole Sejer Iversen, Claus Bossen, Ditte Amund Basballe, and Ben Schouten. 2020. How participatory design works: Mechanisms and effects. In *Proceedings of the 31st Australian Conference on Human-Computer-Interaction*, 30–41. OzCHI '19, New York, NY, USA: Association for Computing Machinery.

Harris, Alexa M., Diego Gómez-Zará, Leslie A. DeChurch, and Noshir S. Contractor. 2019. Joining together online: The trajectory of CSCW scholarship on group formation. *Proc. ACM Hum.-Comput. Interact.* 3(CSCW).

Herlant, Laura V, Rachel M Holladay, and Siddhartha S Srinivasa. 2016. Assistive teleoperation of robot arms via automatic time-optimal mode switching. In *2016*

11th ACM/IEEE international conference on human-robot interaction (HRI), 35–42. IEEE.

Higuch, Keita, 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. CHI '16, New York, NY, USA: Association for Computing Machinery.

Hix, Deborah, and H. Rex Hartson. 1993. *Formative evaluation: Ensuring usability in user interfaces*, 1–30. USA: John Wiley & Sons, Inc.

Hokkanen, Laura, Kati Kuusinen, and Kaisa Väänänen. 2016. Minimum viable user experience: A framework for supporting product design in startups. In *Agile Processes, in Software Engineering, and Extreme Programming: 17th International Conference, XP 2016, Edinburgh, UK, May 24-27, 2016, Proceedings 17*, 66–78. Springer.

Hutchinson, Hilary, Wendy Mackay, Bo Westerlund, Benjamin B. Bederson, Allison Druin, Catherine Plaisant, Michel Beaudouin-Lafon, Stéphane Conversy, Helen Evans, Heiko Hansen, Nicolas Roussel, and Björn Eiderbäck. 2003. Technology probes: Inspiring design for and with families. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 17–24. CHI '03, New York, NY, USA: Association for Computing Machinery.

Hutchinson, S., G.D. Hager, and P.I. Corke. 1996. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation* 12(5):651–670.

Ihara, Keiichi, Mehrad Faridan, Ayumi Ichikawa, Ikkaku Kawaguchi, and Ryo Suzuki. 2023. Holobots: Augmenting holographic telepresence with mobile robots for tangible remote collaboration in mixed reality. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. UIST '23, New York, NY, USA: Association for Computing Machinery.

Jensen, Kurt, and Lars Michael Kristensen. 2009. *Coloured petri nets*. Berlin, Heidelberg: Springer.

Jiang, Shu, 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.

Johansen, Robert. 1988. *Groupware: Computer support for business teams*. The Free Press.

Johnson, Steven, Madeleine Gibson, and Bilge Mutlu. 2015a. 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. CSCW '15, New York, NY, USA: Association for Computing Machinery.

Johnson, Steven, Irene Rae, Bilge Mutlu, and Leila Takayama. 2015b. 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. CHI '15, New York, NY, USA: Association for Computing Machinery.

Jones, Brennan, Yaying Zhang, Priscilla N. Y. Wong, and Sean Rintel. 2021. Belonging there: VROOM-ing into the uncanny valley of XR telepresence. *Proc. ACM Hum.-Comput. Interact.* 5(CSCW1).

Karamcheti, Siddharth, Albert J Zhai, Dylan P Losey, and Dorsa Sadigh. 2021. Learning visually guided latent actions for assistive teleoperation. In *Learning for Dynamics and Control*, 1230–1241. PMLR.

Karsenty, Laurent. 1999. Cooperative work and shared visual context: An empirical study of comprehension problems in side-by-side and remote help dialogues. *Human-Computer Interaction* 14(3):283–315.

Kasahara, Shunichi, Shohei Nagai, and Jun Rekimoto. 2014. Livesphere: Immersive experience sharing with 360 degrees head-mounted cameras. In *Adjunct*

Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology, 61–62. UIST '14 Adjunct, New York, NY, USA: Association for Computing Machinery.

Kehl, Wadim, 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 (ICCV)*.

Kenwright, Benjamin. 2015. Generic convex collision detection using support mapping. *Technical Report*.

Kim, Seungwon, 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)*, 1–6.

Kirk, David, 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*, 1039–1048. CHI '07, New York, NY, USA: Association for Computing Machinery.

Kiselev, Andrey, 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. HRI '14, New York, NY, USA: Association for Computing Machinery.

Knapp, J., J. Zeratsky, and B. Kowitz. 2016. *Sprint*. London, England: Bantam Press.

Kratz, Sven, 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)*, 638–645.

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

Kurata, T., 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.

Kuzuoka, Hideaki, Gen Ishimoda, Yushi Nishimura, Ryutaro Suzuki, and Kimio Kondo. 1995. Can the gesturecam be a surrogate? In *Proceedings of the Fourth European Conference on Computer-Supported Cooperative Work ECSCW'95*, ed. Hans Marmolin, Yngve Sundblad, and Kjeld Schmidt, 181–196. Dordrecht: Springer.

Kuzuoka, Hideaki, 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. CSCW '94, New York, NY, USA: Association for Computing Machinery.

Kuzuoka, Hideaki, 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. CSCW '00, New York, NY, USA: Association for Computing Machinery.

Lanir, Joel, 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*, 2243–2252. CHI '13, New York, NY, USA: Association for Computing Machinery.

Lee, Charlotte P., and Drew Paine. 2015. From the matrix to a Model of Coordinated Action (MoCA): A conceptual framework of and for CSCW. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, 179–194. CSCW '15, New York, NY, USA: Association for Computing Machinery.

Lee, Gun A., Theophilus Teo, Seungwon Kim, and Mark Billinghurst. 2017. Mixed reality collaboration through sharing a live panorama. In *SIGGRAPH Asia 2017 Mobile Graphics & Interactive Applications*. SA '17, New York, NY, USA: Association for Computing Machinery.

Levinson, Stephen C. 2006. Deixis. In *The Handbook of Pragmatics*, ed. Laurence R. Horn and Gregory Ward.

Li, Jiannan, 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 *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. CHI '22, New York, NY, USA: Association for Computing Machinery.

Li, Jiatong, Ryo Suzuki, and Ken Nakagaki. 2023. Physica: Interactive tangible physics simulation based on tabletop mobile robots towards explorable physics education. In *Proceedings of the 2023 ACM Designing Interactive Systems Conference*, 1485–1499. DIS '23, New York, NY, USA: Association for Computing Machinery.

Lincoln, Yvonna S, and Egon G Guba. 1988. Criteria for assessing naturalistic inquiries as reports.

Losey, David P., 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. *ASME Applied Mechanics Reviews* 70(1):010804.

Lugaresi, Camillo, 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*.

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

Machino, T., S. Iwaki, H. Kawata, Y. Yanagihara, Y. 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.*, 4063–4068.

Maddikunta, Praveen Kumar Reddy, Quoc-Viet Pham, B Prabadevi, Natarajan Deepa, Kapal Dev, Thippa Reddy Gadekallu, Rukhsana Ruby, and Madhusanka Liyanage. 2022. Industry 5.0: A survey on enabling technologies and potential applications. *Journal of Industrial Information Integration* 26:100257.

Maguire, Martin. 2001a. Context of use within usability activities. *International Journal of Human-Computer Studies* 55(4):453–483.

———. 2001b. Methods to support human-centred design. *International Journal of Human-Computer Studies* 55(4):587–634.

Malyuta, Dmytro, Christoph Brommer, David Hentzen, Thomas Stastny, Roland Siegwart, and Roland Brockers. 2020. Long-duration fully autonomous operation of rotorcraft unmanned aerial systems for remote-sensing data acquisition. *Journal of Field Robotics* 37:137–157.

Marques, Bruno, Susana Silva, João Alves, and et al. 2022. Remote collaboration in maintenance contexts using augmented reality: Insights from a participatory process. *International Journal of Interactive Design and Manufacturing* 16:419–438.

McGrath, Joseph Edward. 1984. *Groups: Interaction and performance*, vol. 14. Prentice-Hall Englewood Cliffs, NJ.

Meng, Haoming, Yeping Wang, Pragathi Praveena, Michael Gleicher, and Bilge Mutlu. 2023. Demonstrating Periscope: A robotic camera system to support remote physical collaboration. In *Companion Publication of the 2023 Conference on Computer Supported Cooperative Work and Social Computing*. CSCW'23 Companion, New York, NY, USA: Association for Computing Machinery.

Mentis, Helena M., 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. CHI '20, New York, NY, USA: Association for Computing Machinery.

Michaelis, Joseph E., Amanda Siebert-Evenstone, David Williamson Shaffer, and Bilge Mutlu. 2020. Collaborative or simply uncaged? Understanding human-cobot interactions in automation. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 1–12. CHI '20, New York, NY, USA: Association for Computing Machinery.

Mose Biskjaer, Michael, and Kim Halskov. 2014. Decisive constraints as a creative resource in interaction design. *Digital Creativity* 25(1):27–61.

Muratet, Mathieu, Ameni Yessad, and Thierry Carron. 2016. Understanding Learners' Behaviors in Serious Games. In *Advances in Web-Based Learning – ICWL 2016*, ed. Dickson Chiu, Ivana Marenzi, Umberto Nanni, Marc Spaniol, and Marco Temperini, vol. 10013 of *Lecture Notes in Computer Science*. Springer.

Nanavati, Amal, Patricia Alves-Oliveira, Tyler Schrenk, Ethan K. Gordon, Maya Cakmak, and Siddhartha S. Srinivasa. 2023. Design principles for robot-assisted feeding in social contexts. In *Proceedings of the 2023 ACM/IEEE International Conference on Human-Robot Interaction*, 24–33. HRI '23, New York, NY, USA: Association for Computing Machinery.

Neale, Dennis C., John M Carroll, and Mary Beth Rosson. 2004. Evaluating computer-supported cooperative work: Models and frameworks. In *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work*, 112–121. CSCW '04, New York, NY, USA: Association for Computing Machinery.

Nedelkoska, Ljubica, and Glenda Quintini. 2018. Automation, skills use and training (202). `https://www.oecd-ilibrary.org/content/paper/2e2f4eea-en`.

Neisser, Ulric. 1976. *Cognition and reality*. San Francisco: W. H. Freeman.

Nicolis, Davide, 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):796–803.

Niemeyer, Günter, Carsten Preusche, Stefano Stramigioli, and Dongjun Lee. 2016. Telerobotics. In *Springer Handbook of Robotics*, 1085–1108. Springer.

Oda, Ohan, 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. UIST '15, New York, NY, USA: Association for Computing Machinery.

Oh, Yoojin, Tim Schäfer, Benedikt Rüther, Marc Toussaint, and Jim Mainprice. 2021a. A system for traded control teleoperation of manipulation tasks using intent prediction from hand gestures. In *2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*, 503–508. IEEE.

Oh, Yoojin, Marc Toussaint, and Jim Mainprice. 2021b. Learning to arbitrate human and robot control using disagreement between sub-policies. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 5305–5311. IEEE.

Onishi, Yuya, 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. HAI '16, New York, NY, USA: Association for Computing Machinery.

Otsuki, Mai, 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. CHI '18, New York, NY, USA: Association for Computing Machinery.

Palmer, Doug, 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*, 103–110. OZCHI '07, New York, NY, USA: Association for Computing Machinery.

Peterson, James L. 1977. Petri nets. *ACM Computing Surveys (CSUR)* 9(3):223–252.

Pinelle, D., and C. Gutwin. 2000. A review of groupware evaluations. In *Proceedings IEEE 9th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE 2000)*, 86–91.

Piumsomboon, Thammathip, 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. CHI '18, New York, NY, USA: Association for Computing Machinery.

Piumsomboon, Thammathip, 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*, 1–17. CHI '19, New York, NY, USA: Association for Computing Machinery.

Porfirio, David, Allison Sauppé, Aws Albarghouthi, and Bilge Mutlu. 2018. Authoring and verifying human-robot interactions. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, 75–86. UIST '18, New York, NY, USA: Association for Computing Machinery.

Poupyrev, Ivan, Tatsushi Nashida, and Makoto Okabe. 2007. Actuation and tangible user interfaces: The Vaucanson duck, robots, and shape displays. In *Proceedings of the 1st International Conference on Tangible and Embedded Interaction*, 205–212. TEI '07, New York, NY, USA: Association for Computing Machinery.

Praveena, Pragathi, Bengisu Cagiltay, Michael Gleicher, and Bilge Mutlu. 2023a. Exploring the use of collaborative robots in cinematography. In *Extended Abstracts* *of the 2023 CHI Conference on Human Factors in Computing Systems*. CHI EA '23, New York, NY, USA: Association for Computing Machinery.

Praveena, Pragathi, Luis Molina, Yeping Wang, Emmanuel Senft, Bilge Mutlu, and Michael Gleicher. 2022. Understanding control frames in multi-camera robot telemanipulation. In *Proceedings of the 2022 ACM/IEEE International Conference on Human-Robot Interaction*, 432–440. HRI '22, IEEE Press.

Praveena, Pragathi, Andrew Schoen, Michael Gleicher, David Porfirio, and Bilge Mutlu. 2023b. Petri nets for the iterative development of interactive robotic systems. In *Proceedings of the AAAI Symposium Series*, vol. 2, 526–531.

Praveena, Pragathi, Yeping Wang, Emmanuel Senft, Michael Gleicher, and Bilge Mutlu. 2023c. Periscope: A robotic camera system to support remote physical collaboration. *Proc. ACM Hum.-Comput. Interact.* 7(CSCW2).

Praveena, Pragathi, Nathan White, Jill Streamer, Richard Gardner, and Bilge Mutlu. 2024. Towards robotic assistance for just-in-time worker training. Working title, Authorship not finalized, Manuscript in preparation.

Rae, Irene, Bilge Mutlu, and Leila Takayama. 2014. Bodies in motion: Mobility, presence, and task awareness in telepresence. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2153–2162. CHI '14, New York, NY, USA: Association for Computing Machinery.

Rae, Irene, Leila Takayama, and Bilge Mutlu. 2013a. 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. CHI '13, New York, NY, USA: Association for Computing Machinery.

———. 2013b. The influence of height in robot-mediated communication. In *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 1–8.

Rae, Irene, 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. CSCW '15, New York, NY, USA: Association for Computing Machinery.

Rakita, Daniel, 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. HRI '17, New York, NY, USA: Association for Computing Machinery.

———. 2018. An autonomous dynamic camera method for effective remote teleoperation. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, 325–333. HRI '18, New York, NY, USA: Association for Computing Machinery.

———. 2019. Remote telemanipulation with adapting viewpoints in visually complex environments. *Robotics: Science and Systems XV*.

Rakita, Daniel, 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)*, 9995–10001.

Ranjan, Abhishek, 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*, 1177–1186. CHI '07, New York, NY, USA: Association for Computing Machinery.

Rasmussen, Troels Ammitsbøl, 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.

Rea, Daniel J, and Stela H Seo. 2022. Still not solved: A call for renewed focus on user-centered teleoperation interfaces. *Frontiers in Robotics and AI* 9:704225.

Ries, Eric. 2011. *The lean startup: How today's entrepreneurs use continuous innovation to create radically successful businesses*. Currency.

Rodden, Tom, and Gordon Blair. 1991. CSCW and distributed systems: The problem of control. In *Proceedings of the Second European Conference on Computer-Supported Cooperative Work ECSCW'91*, 49–64. Springer.

Runkel, Philip J., and Joseph E. McGrath. 1972. *Research on human behavior*. Holt, Rinehart, and Winston, Inc.

Sabet, Mehrnaz, 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*. CHI '21, New York, NY, USA: Association for Computing Machinery.

Sadokierski, Zoë. 2020. Developing critical documentation practices for design researchers. *Design Studies* 69:100940.

Sakashita, Mose, Hyunju Kim, Brandon Woodard, Ruidong Zhang, and François Guimbretière. 2023. VRoxy: Wide-area collaboration from an office using a VR-driven robotic proxy. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. UIST '23, New York, NY, USA: Association for Computing Machinery.

Sakashita, Mose, E. Andy Ricci, Jatin Arora, and François Guimbretière. 2022. RemoteCoDe: Robotic embodiment for enhancing peripheral awareness in remote collaboration tasks. *Proc. ACM Hum.-Comput. Interact.* 6(CSCW1).

Salovaara, Antti, Antti Oulasvirta, and Giulio Jacucci. 2017. Evaluation of prototypes and the problem of possible futures. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 2064–2077. CHI '17, New York, NY, USA: Association for Computing Machinery.

Sasikumar, Prasanth, 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)*, 393–394.

Schäfer, Alexander, Gerd Reis, and Didier Stricker. 2022. A survey on synchronous augmented, virtual, and mixed reality remote collaboration systems. *ACM Comput. Surv.* 55(6).

Schloesser, Sebastian, Michael Riesener, and Günther Schuh. 2017. Prototyping in highly-iterative product development for technical systems. In *Proceedings*, vol. 7, 149–157.

Schmaus, Peter, Daniel Leidner, Thomas Krüger, Ralph Bayer, Benedikt Pleintinger, Andre Schiele, and Neal Y Lii. 2019. Knowledge driven orbit-to-ground teleoperation of a robot coworker. *IEEE Robotics and Automation Letters* 5(1):143–150.

Schmidt, Kjeld. 2002. The problem with awareness: Introductory remarks on awareness in CSCW. *Computer Supported Cooperative Work (CSCW)* 11(3-4):285–298.

Sebo, Sarah, Brett Stoll, Brian Scassellati, and Malte F. Jung. 2020. Robots in groups and teams: A literature review. *Proc. ACM Hum.-Comput. Interact.* 4(CSCW2).

Senft, Emmanuel, 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.

Senft, Emmanuel, Michael Hagenow, Robert Radwin, Michael Zinn, Michael Gleicher, and Bilge Mutlu. 2021a. Situated live programming for human-robot collaboration. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, 613–625. UIST '21, New York, NY, USA: Association for Computing Machinery.

Senft, Emmanuel, Michael Hagenow, Kevin Welsh, Robert Radwin, Michael Zinn, Michael Gleicher, and Bilge Mutlu. 2021b. Task-level authoring for remote robot teleoperation. *Frontiers in Robotics and AI* 8:707149.

Shah, Chirag. 2014. Collaborative information seeking. *Journal of the Association for Information Science and Technology* 65(2):215–236. `https://asistdl.onlinelibrary.wiley.com/doi/pdf/10.1002/asi.22977`.

Sirkin, David, 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. HRI '12, New York, NY, USA: Association for Computing Machinery.

Sodhi, Rajinder S., 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*, 179–188. CHI '13, New York, NY, USA: Association for Computing Machinery.

Sostero, Matteo, Santo Milasi, John Hurley, Enrique Fernandez-Macías, and Martina Bisello. 2020. Teleworkability and the COVID-19 crisis: A new digital divide? Tech. Rep., JRC working papers series on labour, education and technology.

Speicher, Maximilian, Jingchen Cao, Ao Yu, Haihua Zhang, and Michael Nebeling. 2018. 360anywhere: Mobile ad-hoc collaboration in any environment using 360 video and augmented reality. *Proc. ACM Hum.-Comput. Interact.* 2(EICS).

Spiel, Katta, Laura Malinverni, Judith Good, and Christopher Frauenberger. 2017. Participatory evaluation with autistic children. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 5755–5766. CHI '17, New York, NY, USA: Association for Computing Machinery.

Spinuzzi, Clay. 2005. The methodology of participatory design. *Technical communication* 52(2):163–174.

Stahl, Christoph, 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. PETRA '18, New York, NY, USA: Association for Computing Machinery.

Stolzenwald, Janis, and Walterio W. Mayol-Cuevas. 2019. Reach out and help: Assisted remote collaboration through a handheld robot. *CoRR* abs/1910.02015. `1910.02015`.

Tang, John C. 1991. Findings from observational studies of collaborative work. *International Journal of Man-Machine Studies* 34(2):143–160. Special Issue: Computer-supported Cooperative Work and Groupware. Part 1.

Tecchia, Franco, 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. VRCAI '12, New York, NY, USA: Association for Computing Machinery.

Tellex, Stefanie, Nakul Gopalan, Hadas Kress-Gazit, and Cynthia Matuszek. 2020. Robots that use language. *Annual Review of Control, Robotics, and Autonomous Systems* 3:25–55.

Teo, Theophilus, 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*, 1–14. CHI '19, New York, NY, USA: Association for Computing Machinery.

Thomas, Cathy, and Nigel Bevan. 1996. Usability context analysis: A practical guide.

Thoravi Kumaravel, Balasaravanan, 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. UIST '19, New York, NY, USA: Association for Computing Machinery.

Thoravi Kumaravel, Balasaravanan, and Björn Hartmann. 2022. Interactive mixed-dimensional media for cross-dimensional collaboration in mixed reality environments. *Frontiers in Virtual Reality* 3.

Tsui, Katherine M., and Holly A. Yanco. 2013. Design challenges and guidelines for social interaction using mobile telepresence robots. *Reviews of Human Factors and Ergonomics* 9(1):227–301. https://doi.org/10.1177/1557234X13502462.

Twidale, Michael, David Randall, and Richard Bentley. 1994. Situated evaluation for cooperative systems. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, 441–452. CSCW '94, New York, NY, USA: Association for Computing Machinery.

Vartiainen, Elina, 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. HAI '15, New York, NY, USA: Association for Computing Machinery.

Vaughn, Lisa M, and Farrah Jacquez. 2020. Participatory research methods—choice points in the research process. *Journal of Participatory Research Methods* 1(1).

Venkatesh, Viswanath, and Hillol Bala. 2008. Technology acceptance model 3 and a research agenda on interventions. *Decision Sciences* 39(2):273–315.

Venkatesh, Viswanath, and Fred D Davis. 2000. A theoretical extension of the technology acceptance model: Four longitudinal field studies. *Management Science* 46(2):186–204.

Villanueva, Ana M, 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*. CHI '21, New York, NY, USA: Association for Computing Machinery.

Wallace, James R., Saba Oji, and Craig Anslow. 2017. Technologies, methods, and values: Changes in empirical research at CSCW 1990–2015. *Proc. ACM Hum.-Comput. Interact.* 1(CSCW).

Walsham, Geoff. 1995. Interpretive case studies in IS research: Nature and method. *European Journal of Information Systems* 4(2):74–81.

———. 2006. Doing interpretive research. *European Journal of Information Systems* 15(3):320–330.

Wang, Peng, 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)*, 1219–1220.

Wang, Yeping, Carter Sifferman, and Michael Gleicher. 2023. Exploiting task tolerances in mimicry-based telemanipulation. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 7012–7019. IEEE.

Warner, Mark, Laura Lascau, Anna L Cox, Duncan P Brumby, and Ann Blandford. 2021. "Oops...": Mobile message deletion in conversation error and regret remediation. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. CHI '21, New York, NY, USA: Association for Computing Machinery.

Wing, J.M. 1990. A specifier's introduction to formal methods. *Computer* 23(9): 8–22.

Xiao, Chun, Weidong Huang, and Mark Billinghurst. 2021. Usage and effect of eye tracking in remote guidance. In *Proceedings of the 32nd Australian Conference on Human-Computer Interaction*, 622–628. OzCHI '20, New York, NY, USA: Association for Computing Machinery.

Yang, Longqi, David Holtz, Sonia Jaffe, Siddharth Suri, Shilpi Sinha, Jeffrey Weston, Connor Joyce, Neha Shah, Kevin Sherman, Brent Hecht, et al. 2022. The effects of remote work on collaboration among information workers. *Nature Human Behaviour* 6(1):43–54.

Zhang, Xujing, 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. CHI '19, New York, NY, USA: Association for Computing Machinery.