Demonstrating *Periscope*: A Robotic Camera System to Support Remote Physical Collaboration

Haoming Meng
Department of Computer Sciences,
University of Wisconsin–Madison
Madison, WI, USA
hmeng29@wisc.edu

Yeping Wang
Department of Computer Sciences,
University of Wisconsin–Madison
Madison, WI, USA
yeping@cs.wisc.edu

Pragathi Praveena
Department of Computer Sciences,
University of Wisconsin–Madison
Madison, WI, USA
pragathi@cs.wisc.edu

Michael Gleicher
Department of Computer Sciences,
University of Wisconsin–Madison
Madison, WI, USA
gleicher@cs.wisc.edu

Bilge Mutlu
Department of Computer Sciences,
University of Wisconsin–Madison
Madison, WI, USA
bilge@cs.wisc.edu

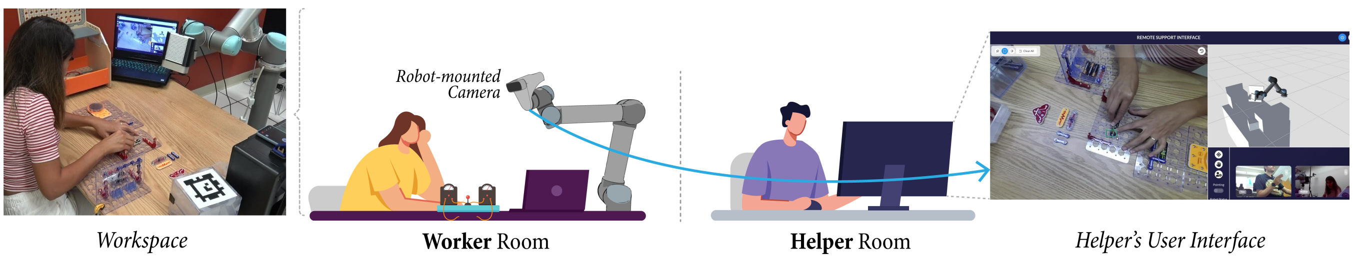

Figure 1: We demonstrate *Periscope*, a robotic camera system that is designed to support two individuals (a worker and a helper) working synchronously and collaboratively on a physical task, while being located in different spaces.

ABSTRACT

This paper proposes a demonstration of *Periscope*, a robotic camera system that allows two people to collaborate remotely on physical tasks. With *Periscope*, a worker co-located with a robotic arm can receive guidance from a remote helper who observes the workspace through a camera mounted on the robot. *Periscope* facilitates remote collaboration by providing the worker and the helper with shared visual information that enhances their verbal communication and coordination processes.

CCS CONCEPTS

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

ACM Reference Format:
Haoming Meng, Yeping Wang, Pragathi Praveena, Michael Gleicher, and Bilge Mutlu. 2023. Demonstrating *Periscope*: A Robotic Camera System to Support Remote Physical Collaboration. In *Proceedings of The 26th ACM Conference On Computer-Supported Cooperative Work And Social Computing (CSCW'23)*. ACM, New York, NY, USA, 4 pages. https://doi.org/XXXXXXX.XXXXXXX

CSCW'23, October 14–18, 2023, Minneapolis, MN

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

This paper proposes a demonstration of a system designed to support two individuals working synchronously and collaboratively on a physical task, while being located in different spaces. The system has potential applications in various scenarios, such as a field technician seeking guidance from an expert to repair a wind turbine, an expert providing training to car mechanics on how to repair a new engine model, or an astronaut getting help from ground control to maintain critical infrastructure on the space station. To support effective collaboration in such scenarios, it is crucial for the remote expert to have flexible views of the workspace from different angles and in varying levels of detail. Our system called *Periscope* [3] uses a robotic camera to provide real-time, dynamic information and viewpoint flexibility to users to facilitate collaboration (see Figure 1).

The *Periscope* system supports a "worker" in completing physical tasks with remote guidance from a "helper" who observes the workspace through a robot-mounted camera. The camera view, displayed on separate screen interfaces for both users, is essential to the *shared environment* [2] created by *Periscope* for facilitating collaboration between the two users. This shared environment enables both collaborators to share task-relevant visual information and promotes the development of a mutual understanding during the collaboration process. The shared environment also includes

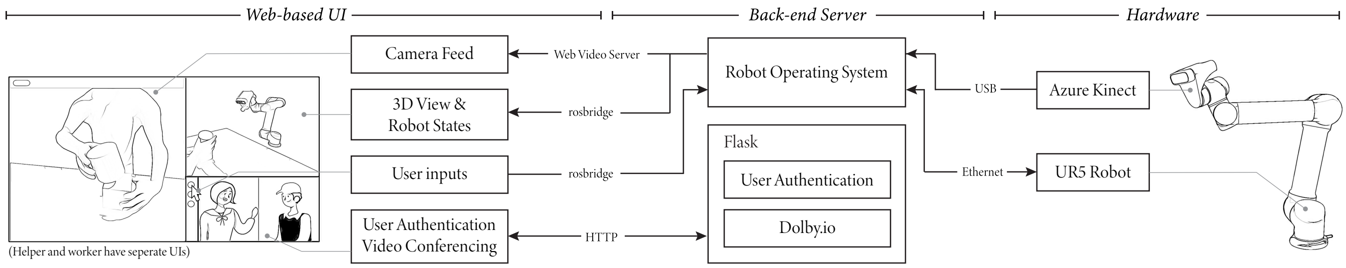

Figure 2: The *Periscope* system establishes a shared environment between the worker and the remote helper using web-based interfaces and a robot-mounted camera.

capabilities for collaborators to annotate the camera view to refer to objects (annotations are visible to both users), a 3D model of the workspace, and video conferencing. Additionally, the camera view can be adjusted by both users through intuitive control mechanisms that we have developed, to gather and share relevant visual information for the task at hand. Our demonstration will highlight intuitive interaction modalities and autonomous robot behaviors that make it easy for both local and remote users to control a robotic camera, and how the system can facilitate successful remote collaboration on physical tasks. Our demo of the *Periscope* system aims to showcase to the CSCW audience how robotic camera systems can provide novel capabilities for computer-supported cooperative work. Below, we present an overview of the system design, highlighting its key ideas (§2), and describe our proposed demo workflow (§3).

2 THE PERISCOPE SYSTEM

In this section, we will describe the key capabilities afforded by the *Periscope* system to create a shared environment, shape the camera view, and ensure safety. To support remote collaboration, *Periscope* consists of separate web-based, front-end UIs for each collaborator and a back-end server which controls a robot-mounted camera (Figure 2).

2.1 Creating a Shared Environment

A shared environment that provides timely and group context to collaborators is essential to the success of a remote collaboration system [2]. The *Periscope* system offers a shared environment to collaborators through a live video feed from a robot-mounted camera, screen annotation capabilities for referencing objects or regions in the camera feed, a 3D visualization of the worker's space, and video conferencing.

The video feed from the robot-mounted camera serves as the primary source of shared visual information between the collaborators. In order to establish a shared environment, both the helper and the worker can see the same camera view on their respective UIs, and they can *annotate* on the camera feed to refer to specific objects or regions within the view. Besides the camera feed, *Periscope* also provides a 3D simulated workspace of the robot and its surroundings. This allows the remote user to build a spatial mental model of the workspace. Finally, *Periscope* features a video conferencing module that enhances the shared environment by enabling verbal communication and provides users with the ability to see non-verbal cues, such as gestures or facial expressions.

Implementation - The video feed from the robot-mounted camera is streamed in real-time to both the helper's and worker's interfaces. In order to facilitate referential communication, both the worker and the helper can annotate the camera feed by drawing rectangles, arrows, or pinpoints with a specific color associated with the respective user. We developed the 3D simulated view using the `three.js` library. The 3D view includes a digital twin of the robot, which mimics the joint states of the physical robot. In our implementation, static objects, *e.g.*, a table, in the robot's surroundings are hard-coded and dynamic objects, *e.g.*, a toolbox that can be moved, are detected and located using AR markers. Alternatively, a more advanced 3D reconstruction method can be employed. We integrated a third-party video conferencing API from `Dolby.io` [1] 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 camera feeds.

2.2 Shaping the Camera View

Shaping the robotic camera view can be challenging because users can find it difficult to deal simultaneously with all of the camera's degrees of freedom in 3D space [1]. Our system provides users with three modes for shaping the view depending on the needs of the task. Each mode enables both the helper and the worker to shape the view to varying degrees, with assistance from different types of autonomous robot behaviors. These autonomous behaviors enable users to control the robotic camera with simple inputs, while allowing us to map those inputs to complex robot motions. We refer the reader to our full paper for a detailed explanation on the reasoning behind each mode [3].

Mode 1 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. The helper moves the camera from a remote location using their web-based UI. The helper uses through-the-lens control, where the camera view is adjusted through controls in the image plane. If the worker does a pointing gesture toward an object, the helper can

[1]https://dolby.io/

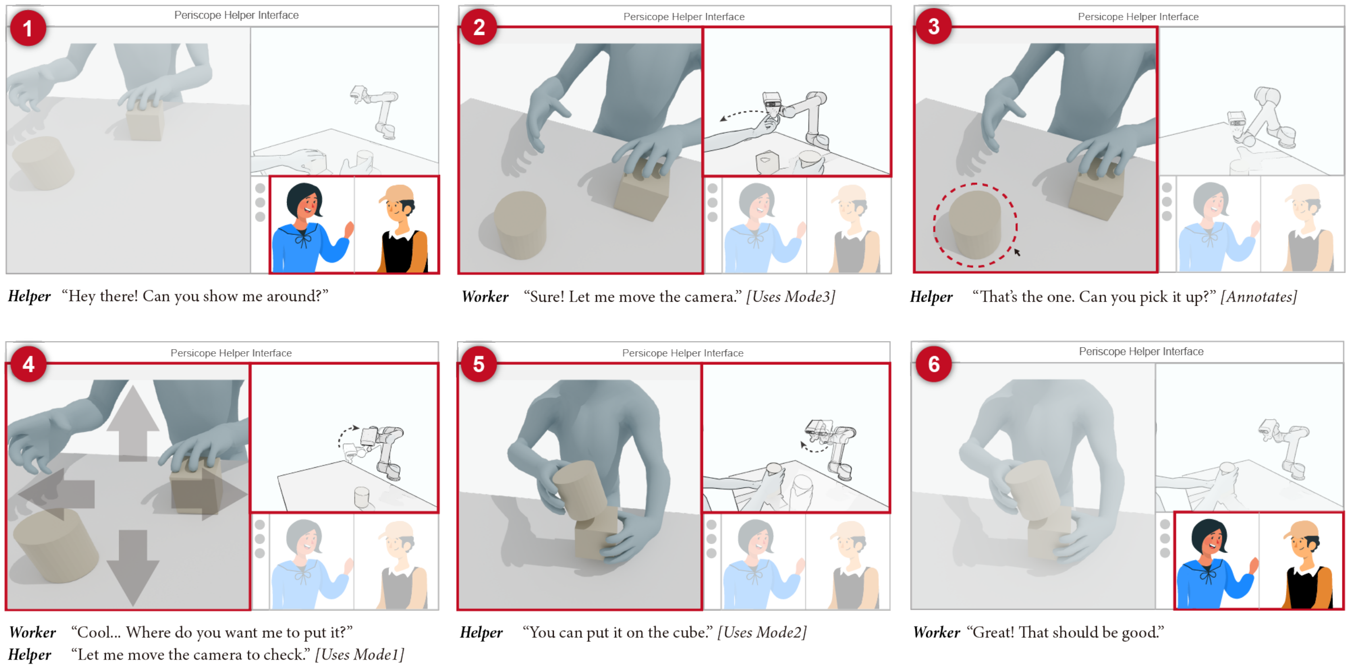

Figure 3: (1) Our demonstration starts with verbal greetings between a conference attendee (the helper) and a remote worker who are collaborating on an assembly task. (2) The worker manually moves the camera using Mode 3 to assist the helper with exploring the workspace. (3) The helper identifies a component and informs the worker by annotating on the camera feed. The worker will use this component in the assembly task. (4) The helper shapes the camera view to search for the next component using Mode 1. (5) The worker assembles the two components together with instructions from the helper while the robot tracks the worker's hand in Mode 2. (6) The worker confirms that the components are correctly assembled.

approve the target, prompting the robot to move autonomously and position the camera to view the target.

Mode 2 is designed to reduce the workload of camera control for both the helper and the worker. In Mode 2, the robot automatically tracks the worker's hand and maintains it in the camera view. The helper can make minor adjustments to the view using their UI. In this mode, the worker can focus on completing the physical task, while the robot captures the worker's activity in the workspace. This mode allows the helper to focus on providing guidance without the need to control the camera to monitor the worker's behaviors.

Mode 3 gives the worker substantial control of the camera. The worker can set the camera's pose through manually moving the robot while the helper can adjust the view through their UI. 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.

Implementation - The helper can shape the view remotely using mouse inputs. The helper can *right-click* on the camera feed to specify a target for the camera to look at, *left-click and drag* on the camera feed to move the camera around the target, and *scroll the mouse wheel* to move the camera closer or further away from the target. The worker can shape the view using gestures or touch. The system detects pointing gestures and the worker's hand using MediaPipe[2]. When the worker attempts to move the robot, the robot arm detects the applied forces and moves accordingly in the direction of the force. The web-based UIs provide toggle buttons for both users to switch modes to shape the view.

2.3 Ensuring Safety

Safety concerns arise when a robot is co-located with a person and sharing the same workspace, particularly when the robot can be remotely controlled by another person. In *Periscope*, only authorized users can access the system through a login page. Additionally, safety measures are in place through autonomous robot behaviors to ensure that the robot can automatically avoid collisions with objects in its surroundings, including the worker, regardless of the selected mode.

Implementation - *Periscope* has a password-based authentication and authorization mechanism that only allows authorized users to access the system and control the robotic camera. For collision avoidance and safe robot motion, we use an optimization-based method called CollisionIK [4].

3 DEMONSTRATION

In our demonstration, an attendee will collaborate with a remote worker (located in our university lab along with the robot) to finish an assembly task. The attendee will be given simple task instructions by the demo facilitators and will serve as the helper. As shown

[2]https://developers.google.com/mediapipe

in Figure 3, our demonstration will include different stages of a collaboration process, where the helper, the worker, or the autonomous robot may shape the camera view. The demonstration will showcase the shared environment enabled by the *Periscope* system to facilitate successful remote collaboration on physical tasks.

REFERENCES

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

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

[3] Pragathi Praveena, Yeping Wang, Emmanuel Senft, Michael Gleicher, and Bilge Mutlu. 2023. Periscope: A Robotic Camera System to Support Remote Physical Collaboration. *Proceedings of the ACM on Human-Computer Interaction* CSCW (to appear) (2023).

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