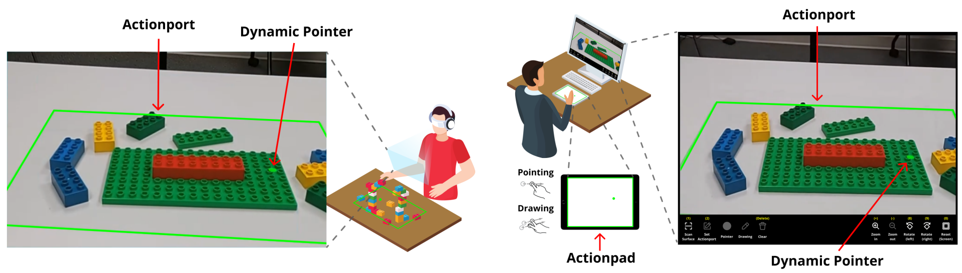
HoloMentor: Enabling Remote Instruction through Augmented Reality Mobile Views

JWAWON SEO, University of Maryland, Baltimore County, USA
IGNACIO AVELLINO, Sorbonne Université, CNRS, ISIR, France and University of Maryland, Baltimore County, USA
DAMARUKA PRIYA RAJASAGI, University of Maryland, Baltimore County, USA
ANITA KOMLODI, University of Maryland, Baltimore County, USA
HELENA M. MENTIS, University of Maryland, Baltimore County, USA



(a) Local Worker (b) Remote Helper

Fig. 1. Overview of HoloMentor. (a) A local worker wears an AR HMD while performing a physical task. They see a world-stabilized, 2D Actionport (green square) overlaid on their physical workspace. Within the Actionport's boundaries is displayed the remote helper's annotations and pointer (pointer is a green dot on right side of green flat building block plate). (b) A remote helper sits in front of a computer display and uses the Actionpad to point and draw in the Actionport

The introduction of Augmented Reality (AR) Head Mounted Displays (HMDs) in collaboration between remote and local workers, introduces new challenges given that camera views are now mobile. We introduce HoloMentor, an AR HMD-based collaborative system designed for remote instruction over *live mobile* views during physical tasks. Through Actionport, we provide a world-stabilized area where remote helpers can dynamically place a pointer and annotations on the physical environment. Through Actionpad, we provide an indirect input mechanism with an absolute position to the Actionport. We show how these innovations worked for participants engaged in a remote instructional task and how they supported effective and efficient communication. Finally, we provide the next steps for addressing AR on mobile views for remote instruction.

CCS Concepts: • **Human-centered computing → Mixed / augmented reality**; **Collaborative and social computing systems and tools**.

Authors' addresses: Jwawon Seo, jwawon.seo@umbc.edu, University of Maryland, Baltimore County, 1000 Hilltop Cir, Baltimore, MD, USA, 21250; Ignacio Avellino, ignacioavellino@gmail.com, Sorbonne Université, CNRS, ISIR, 4 Place Jussieu, Paris, France, 75005 and University of Maryland, Baltimore County, USA; Damaruka Priya Rajasagi, u17@umbc.edu, University of Maryland, Baltimore County, USA; Anita Komlodi, komlodi@umbc.edu, University of Maryland, Baltimore County, USA; Helena M. Mentis, mentis@umbc.edu, University of Maryland, Baltimore County, USA.

2573-0142/2023/1-ART11 $15.00
https://doi.org/10.1145/3567561

Additional Key Words and Phrases: augmented reality, remote instruction, shared information, pointing and annotating, decoupled interaction

ACM Reference Format:
Jwawon Seo, Ignacio Avellino, Damaruka Priya Rajasagi, Anita Komlodi, and Helena M. Mentis. 2023. HoloMentor: Enabling Remote Instruction through Augmented Reality Mobile Views. *Proc. ACM Hum.-Comput. Interact.* 7, GROUP, Article 11 (January 2023), 29 pages. https://doi.org/10.1145/3567561

1 INTRODUCTION

Collaborative systems that provide shared visual information through shared workspaces have been studied for decades in CSCW, and their benefits when remote helpers guide local novices during physical tasks are well known [12, 18, 19]. For example, shared workspaces improve situation awareness and are resources during conversational grounding [18] improving communication between collaborators [19], and virtual pointing in shared workspaces improves performance by reducing movement quantity for physical tasks [12] as implicit guidance becomes explicit visual cues [11]. Now, with Augment Reality (AR), a remote helper can view a local worker's *physical* workspace and directly augment the local worker's view. Thus, industry developers and academic researchers alike have begun designing for remote instruction with AR HMDs. For instance, Microsoft has touted the benefit of the HoloLens for remote instruction with Microsoft's Dynamics 365 Remote Assist and researchers have been studying the benefits of augmenting the local worker's AR HMD view with virtual arms for a remote helper to point to objects of interest [26, 32].

However, new challenges arise when designing for remote instruction through an AR HMD. One significant challenge is there is a shift from a *fixed* view to a *mobile* view, that is, the camera is no longer in a fixed position, but rather it moves with respect to the worker's position through an HMD or any mobile device. Jones et al. [25] pointed out that referencing objects is difficult with a mobile camera and so typically a local worker is required to move carefully to a suitable location where the remote helper can then refer to a target object. Without this coordinated manual stabilization of the view, it is not clear how remote helpers can dynamically create stabilized annotations for instruction — i.e., how to point or draw a circle on a real-world object while the camera moves about in the space. The primary obstacle with a moving camera is that, as a remote expert clicks on a pixel at a given moment to draw, the next moment that pixel corresponds to another location in the local person's real world. The main approach to overcome this obstacle thus far has been to require the remote expert to take a snapshot of the view before interacting on it; essentially "freezing" in time the shared workspace, drawing on it, and then either the system (a) presents it as a floating image as done by the Microsoft HoloLens Dynamics 365 Remote Assist, or (b) transfers the finished drawing as a stabilized annotation on the local worker's HMD [9, 16, 17, 27, 28]. However, the "freeze" approach limits the remote helper's ability to engage in deictic referencing — i.e. dynamically pointing or annotating while providing verbal utterances such as "here" — which is important for facilitating clear as well as efficient communication [5, 27].

Working on these challenges is a pressing concern as AR HMDs become more common and available as a consumer technology [35, 36]. In the following paper, we introduce HoloMentor, an AR HMD-based collaborative system designed for remote instruction over *live mobile* views during local worker's completion of physical tasks. We present two innovations that address the above identified challenges: (1) dynamically aligning a pointer on a *live mobile* view, i.e. without the need for the remote helper to freeze the view, and (2) providing an accurate and intuitive interaction mechanism for a remote helper to point and annotate on those live mobile views. We evaluated these innovations with a local worker completing a building-blocks task via the instruction of a remote helper.

Our findings showed that the pointer and the annotations were sufficiently accurate for the remote helper to use them during instruction and clear for the local worker to understand their reference. In addition, the remote worker was able to understand how to reliably use our input method for moving over the live mobile view. Both of these innovations, in essence, enabled the remote helper to provide clear instructions through dynamic pointing and annotating over live mobile views. We end by presenting additional challenges that arose during the study that point to further areas of development.

2 RELATED WORK

We first present previous work on the concept of deictic referencing and annotating, highlighting that the communication benefits of both functions in remote work has been studied using *fixed* views—where the camera capturing the view does not move. We then present recent work that implements remote instruction in mobile views, highlighting that they require the remote user to freeze the live mobile view in order to annotate, a limitation that we aim to overcome. We finish by discussing techniques previously studied to annotate 3D environments using 2D devices, to contextualize our design rationale.

2.1 The Need for Deictic Referencing and Annotating in Remote Instruction

Deictic referencing and annotating have a long history within the CSCW community. It has been shown that remote collaboration systems that support deictic referencing facilitate the effectiveness [5], conversational grounding [14], and situational awareness [31] of collaboration. Researchers have also shown that the visual cues from deictic referencing on a live video stream with a fixed view of a physical workspace provide rich communication through tools such as a telepointer [5], annotations [9], and the users' hand gestures [24]. Those tools are commonly used to provide spatial cues for remote instruction. For instance, a telepointer is well known as an effective mean for collaborators to establish a common understanding of what is being discussed [5]. For instance, Fussell et al. [14] introduced a system with a telepointer and drawings as referencing tools on a shared live video from a fixed camera. They found that helpers more often used the telepointer during discussions, and also found that drawing tools were used to indicate objects and locations. Thus, CSCW researchers have already made a strong case for ensuring that remote collaborative systems support deictic referencing and annotating over live views by the remote collaborator. Although this work has been solely focused on remote interactions over a fixed view, this need does not disappear simply due to the moved to a *mobile* view, such as with an AR HMD. However, in the next section, we will see that, in the move to a live *mobile* camera view in AR systems, the ability to realize such collaborative tools presents a significant development challenge.

2.2 Supporting Pointing and Annotating on a Live Mobile View in Remote Instruction

Because of the demonstrated need for pointing and annotating on live views and the proliferation of AR, HMDs, and telecommunication systems in distributed collaboration, researchers have been developing mechanisms for pointing and annotating with live *mobile* views. When the camera viewpoint changes as the local worker navigates a physical space, annotations become misplaced when they are not world-stabilized and pointing can be inaccurate. Previous work has largely addressed the live mobile view interaction problem by having the remote helper freeze the view to annotate, essentially making the view not mobile (nor live) for a short period [5, 7, 9, 16, 17, 28]. Bauer et al. [5] first proposed this approach and showed that users understood the use of the image freezing functionality and were able to use it for remote pointing and annotating. Later, Chen et al. [7] also implemented world-stabilized annotations and hand overlays on a snapshot of a live mobile view for remote assistance in a car repair task, showing that this technique outperforms voice only assistance. Gauglitz et al. [15, 17] proposed a world-stabilized annotation system for HMDs in two different ways: by tracking the point of view of the camera [15] and by reconstructing a remote environment through Simultaneous Localization And Mapping (SLAM) for remote collaborator interaction [17]. However, to annotate, remote users still needed to freeze the view and pointing was not supported. Finally, there has also been an effort to make the freezing function automatic where video is frozen when the user first clicks to begin an annotation, and unfrozen when the user unclicks at the end of the annotation [9, 28].

The closest one has gotten in annotating directly on a live mobile view is Lukosch et al. [34]. They developed a system where remote helpers can click on a live mobile video of a remote space to drop world-stabilized virtual objects, the system computes the drop position in world coordinates by ray-casting the mouse position from screen coordinates. Additionally, users can also freeze the view when they need longer interaction (e.g., place multiple objects). Although, directly interacting with the live mobile video, there was still no deictic referencing, and the annotations were pre-defined shapes dropped into the space, thus still not achieving the tight overlap of talk and action that is the ideal realization of remotely annotating live mobile views. Although the freeze approach provides the remote helper with the ability to accurately point and annotate on snapshots of live mobile views, it comes with drawbacks. It can cause visible confusion when there is a misalignment between the frozen frame and the real world [5]. More importantly, it can lead to loss of awareness when users unfreeze the view and stumble upon a dynamic world that has changed without them seeing it, as well as to lesser or delayed feedback compared to a view that is not frozen [9].

In summary, although ideally pointing and annotating should be possible directly on mobile views that are live [27, 29], without having to freeze them, this has not been achieved to date. The motivation of our work is to provide this ability in a manner that is accurate and easy to use.

2.3 Interaction for AR Annotations on a 3D Remote Environment

In order to be able to seamlessly interact with live mobile views, we also needed to devise an input interaction mechanism. Researchers have suggested two modalities for a remote helper to add annotations to a local worker's AR-enabled view: 2D interface on a 2D view or a 3D interface through a zSpace display that renders a 3D reconstruction of the remote workspace. Unfortunately, Anton et al. [2] revealed that a 3D interface increased workload for the remote helper when compared to the 2D interface although the performance between the 2D and 3D interfaces remained the same. Thus, it is preferable to design for a 2D interface on a 2D view.

In terms of 2D interaction for remote helpers, once world-stabilized pointing and annotating is technically achieved, the next challenge is how to interact with a 2D surface that has an effect on a remote 3D world. Some researchers have devised mouse interaction for a remote helper to control a pointer or draw annotations on a 2D display [5, 6, 14, 17, 27]. In theses examples, the remote helper perceives the local worker's 3D environment and interacts directly with the mouse. Gauglitz et al. [16] proposed a touchscreen-based interface to translate 2D interactions to the 3D environment. Here, a remote helper can draw annotations to the 3D environment using a single finger, as well as can navigate the live view with two-finger gestures. Still, other researchers have presented systems that support hand gestures to directly interact with the 3D environment [24, 40]. For instance, Kirk et al. [30] found that gestures-based cues are beneficial for collaborative behavior. Moreover, they revealed that shared gestures improved the user experience of natural interaction. Stafford et al. [39] proposed a new interaction metaphor termed "god-like interaction" where a remote helper pointed at a location with their hand on the table-top surface, and their 3D hand gesture was represented in physical workspace. Wang et al. [40] proposed sharing 2.5D gestures to support remote instruction, the remote helper used 3D hand gestures using a Leap Motion, which projected onto the remote workspace. And Huang et al. [23] developed a system that enabled a remote helper to interact with the physical environment with augmented 3D hands by utilizing 3D real-time capturing camera.

In spite of these various efforts to interact with a remote 3D world, only direct interaction has been studied as a way to interact with a 2D live mobile view by moving a mouse or touching the screen. This design choice has ultimately precluded the addition of functionality for deictic referencing on live, mobile video. In the design of the input interaction mechanism for HoloMentor that we will outline below, we propose an indirect interaction that is decoupled with live mobile view to control world-stabilized pointers and annotations of the remote environment.

3 HOLOMENTOR DESIGN RATIONALE

In the design of a technique for remotely performing world-stabilized pointing and annotating on a live mobile view, we made several choices that concern both the local worker and remote helper. We have made the following design decisions based on the research presented in the prior section as well as our interaction goals and our own iterative design experience, which we document here.

Local worker. Local workers are responsible for the live mobile view, as they wear the AR HMD on their heads. The first decision then is how much to limit the local worker's head movement (free vs. constrained). Constraining head movement entails a social rather than technical solution. It includes having the remote helper ask the local person to stay still when annotating. The advantage with this approach is that there is no synchrony problem (both remote and local collaborators see the same view) and minimal movement means any pointing or annotating is accurately placed on that view. However, in environments that are already cognitively and physically taxing, such as in surgical telementoring or paramedic teleconsulting, this approach puts an additional coordinative burden. In fact, researchers have shown that interacting with supporting technologies for secondary tasks while performing a complex primary task like surgery can interfere with cognitive and physical demands of the latter [1, 3, 4], which in turn has been shown to lead to errors [33, 41]. Thus, our aim was to find a solution that does not unnecessarily constrain the local worker's movements, as we aimed to support dynamic tasks.

Remote helper. Remote helpers are responsible for pointing and creating annotations, to which the system provides feedback. Most important to note is that existing solutions do not provide a pointer that can represent the remote helper's dynamic cursor movements over a snapshot or a live view. From prior work in the field of CSCW, we knew that deictic referencing was an important part of remote instruction (e.g. [5, 10, 13, 27]). Thus, we needed to find a mechanism for pointing as well as annotating a live mobile view as one would naturally do on a static video image. Therefore, we lay choices on both the input mechanism and the output display.

Output. Output — visualization of the pointer or annotation locations — could happen on the shared view, or elsewhere. As we intend to support live mobile views, and dynamic worlds, we want helpers to focus their attention on the shared view, thus, we chose to make all output reside there. Therefore, we chose to fix the pointer and annotations on the real world by making them world-stabilized. We made two further choices as the volume where the pointer and annotations can live is potentially infinite, where most of it is irrelevant to the task. First, to constrain the output space to a 2D plane where the task is executed, such as a table, rather than a 3D volume. Second, to constrain the output space to a portion of this plane, where most of the task takes place. These choices are guided by simplicity, we started our exploration considering instructions on a flat surface, rather than in mid-air, and with a limited number of objects.

Input. The real challenge in meeting the interaction need came from finding an input mechanism to accurately and intuitively interact in the 2D input surface showing the live mobile view. The first option was having the helper use the desktop mouse to click on the video, and then project that point to the output space to produce an annotation. However, this proved to not be ideal in a mobile view. As soon as the view changes from the worker moving their head, the pixel that the remote worker clicks also changes. Therefore, if the helper looks away when the helper clicks to annotate, the result is the unintended annotation of a line that follows the head movement. Our solution was to create a fixed coordinate space for both user, by transforming the XY coordinates of the mouse cursor in the video window, to the corresponding XY coordinates of the output space, assuming border alignment. For this, we defined the output space as a rectangle on the table in front of the worker. As such, when a remote helper would move their mouse towards the upper right corner of the video display window, the AR pointer would correspondingly move to the upper right corner of the output space. As this resulted in having two cursors/pointers on the remote helper's video display window, which would be confusing for a remote helper, we made the mouse cursor invisible when in the video display window's bounds.

We tested this first iteration of the design in a pilot test. Two professional HCI researchers pretested HoloMentor with the same procedure we describe later in Section 5.2, and three researchers observed the pilot test. We identified the following problems with mouse interaction in this first pilot test. The first issue was that there was confusion as they did not know how far to move the mouse to get to a corner of the view without having that on-screen cursor feedback. There was also a discrepancy in the degree of movement with the mouse in the video display window and the distance the AR pointer moved in the output space, given the difference in sizes between the two. Second, interacting with controls below the live video, such as a color palette to change the color of annotations, was also a problem. It was difficult for remote helpers to know how much to move the mouse in order to go from the video area to the button menu area, because the position of the pointer is visible only in the output space, not on the screen. Therefore, from this pilot test, we identified the need to separate the interaction with the desktop application and its toolbar from the interaction in the output space by the remote helper. We decided to devise an *indirect* input mechanism. In addition, we determined that an *absolute position* input device that supports the one-to-one mapping from the position of one's finger to the cursor location would be ideal.

4 HOLOMENTOR DESCRIPTION

We introduce HoloMentor, an AR HMD-based collaborative system designed for remote instruction over *live mobile* views during local worker's completion of physical tasks. HoloMentor addresses two significant challenges: (1) dynamically aligning a pointer on a *live mobile* view, i.e. without the need for the remote helper to freeze the view, and (2) providing an interaction mechanism for a remote helper to point and annotate on those live mobile views. The basis of HoloMentor consists of two parts: a desktop application for the remote helper (Figure 1(b)); and a HoloLens2 application for the local worker (Figure 1(a)). The desktop application provided the remote worker with the ability to point on, annotate, and erase annotations on, rotate, and zoom in/out the remote helper's video view. The desktop and HoloLens2 apps are programmed in Unity 2019.3.13f1. We use the MixedReality-WebRTC[1] framework to establish a two-way audio and one-way video connection between the HoloLens2 app and the desktop app over a local WiFi network. Video is captured using the HoloLens2 world-facing camera and streamed to the remote helper's application enabling the remote helper to see the environment the local worker is looking at in real-time.

In the following sections, we first introduce **Actionport**, a world-stabilized segment of the local worker's physical environment where a dynamic pointer can accurately reference and draw over a live mobile view. We then introduce **Actionpad**, a tablet- or mobile phone- based input mechanism that lets the remote helper manipulate the pointer to accurately and intuitively point, gesture, and annotate in the Actionport.

[1] https://github.com/microsoft/MixedReality-WebRTC

4.1 Actionport

In order to provide remote helpers with pointing and annotating functionality over a live stream from the local worker's head-mounted camera, we developed the **Actionport**, which is placed by the remote helper in a fixed position as a virtual overlay in the local worker's physical environment. By fixing the position of the Actionport, there is now a stable, defined space in the local worker's environment with a one-to-one mapping to a defined space on the remote helper's display. This defined shared workspace is not affected by the movement of the camera, thus a remote helper can control a cursor in the space to dynamically point or draw. For instance, when a remote technician points to a wire on a control panel and then the local technician moves their head to the right by 2 inches, the pointer remains over the wire.

In order to facilitate remote and local worker coordination in the placement of the Actionport, we provided a feedforward mechanism. Even though the remote helper cannot move the position of the Actionport themselves, they can coordinate with the local workers to place the Actionport through this feedforward mechanism. This enables both collaborators to preview where an Actionport will be placed by using a raycast along the local worker's gaze direction (Figure 2). The remote helper then pushes a button on the remote desktop application's toolbar to affix the Actionport to the area shown in the preview. An Actionport is then displayed as a green rectangle in the shared workspace seen by both the local and remote helper (refer back to Figure 1(b)). Once the Actionport is placed, the remote helper's pointing functionality is automatically activated.

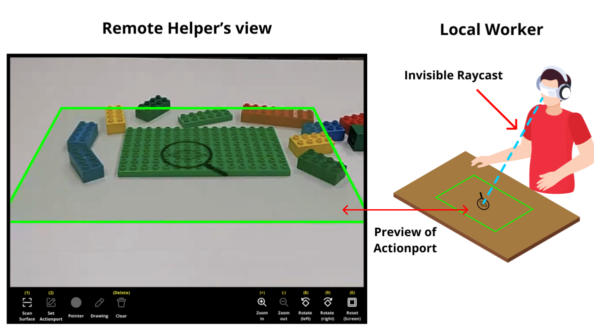


Fig. 2. Demonstration of the raycast for placement of the Actionport.

4.1.1 Actionport Development. To develop this, we rely on the spatial mapping feature of the open-sourced Mixed Reality Toolkit to stabilize the Actionport and its content (i.e. the pointer and annotations) in the local worker's environment. The spatial mapping feature creates triangle meshes on real-world surfaces in the environment around the HoloLens. We first find the nearest spatial surface where the Actionport will be overlaid by using a raycast which lies within the camera's view orientation. Then, we calculate a position and a normal vector (ie, a vector perpendicular to a plane) of a triangle mesh that the ray hit. We place here an Actionport and with the normal vector, that is, parallel to the triangle mesh. Also, to rotate the bottom direction of the Actionport toward the local worker, we calculated a vector between the position of the local worker and the Actionport, and then adjusted the x angle of the Actionport.

4.2 Actionpad

We devised **Actionpad**, an indirect input mechanism, for remote helpers to *act* on a tablet through touch and *perceive* the effect in the Actionport on the desktop application's video display window (Figure 1(b)). The tablet face's rectangular boundaries represented the Actionport's rectangular boundaries and so when moving one's finger on the tablet's face the HoloLens' pointer would move in the corresponding XY coordinates in the Actionport. Also, if the remote helper rotates the video display, they could rotate the tablet in accordance with the resulting orientation of the Actionport to make it easier to map the movement of the finger with the intended movement of the pointer. Based on an indirect input mechanism and one-to-one mapping, the Actionpad provided feedback for the remote helper to determine whether their cursor's location is in real-time. We implemented two interactions: to move a pointer, a user touches the touchscreen with one finger and moves it over the surface in the XY plane, and to draw, the user touches the touchscreen with two fingers and moves them in tandem over the surface. The user could change from one to two finger and back again, akin to a clutch mechanism, to switch modes seamlessly; e.g. move the cursor to a starting point, drop the second finger to the surface of the touchscreen, move both fingers in tandem to draw a line, raise the second finger to stop drawing.

4.2.1 Actionpad Development. We developed a web-application loaded on a tablet (Figure 1(b)) as an accompaniment to the desktop application. The web app was programmed with JavaScript and we used websockets through the Socket.io[2] library to stream touch interaction from the tablet to the desktop app. Lastly, we used Photon Unity Networking 2[3] (Photon Engine) to synchronize the pointer and drawings over the network.

5 EVALUATION METHOD

We evaluated HoloMentor in a controlled setting in order to understand if and how it can overcome the challenge of pointing and annotating over a live mobile view. Specifically, we were interested in knowing: (i) whether the *Actionport* was able to sufficiently align the remote helper's dynamic pointer and annotations and maintain that alignment with a mobile camera view, and (ii) whether the remote workers were able to use the *Actionpad* to effectively point and annotate in the Actionport.

5.1 Participants

Sixteen participants were recruited in June 2021, through targeted email solicitation. As this study was conducted during the COVID pandemic, we were required by the university to only recruit from students in University of Maryland, Baltimore County who already had approval to be on campus. Once participants replied with their interest to participate, we manually paired them based on their availability. All of our participants were between 18-30 years of age, of which four identified as female and twelve as male and none had any prior experience with the Microsoft HoloLens. Each participant was thanked with $20 for their time and effort.

5.2 Procedure

Each participant was placed in a separate room. The participant who played the role of the remote helper was sat in front of a desktop computer and provided a mouse, keyboard, and the Actionpad tablet. The participant who played the role of a local worker was sat at a table and provided with

[2] https://socket.io/
[3] https://photonengine.com/pun

the HoloLens2 AR glasses.[4] The communication between the remote helper and the local worker to complete the task was accomplished solely by using the HoloMentor's two-way audio channel, video from the AR HMD system, and augmented reality visual overlays.

After consent was attained, a pre-questionnaire was distributed (see section 5.3). After completion of the pre-questionnaire, each participant watched the same introductory presentation that outlined how the system worked (e.g. how to set the Actionport, how to use the Actionpad) and how HoloMentor could be used for remote instruction. After completion, the participants then engaged in a practice task with HoloMentor. The remote helper was provided with a drawing of a house and a guide that ensured the participants would gain experience in using each functionality at least once. The local worker was provided a pen and a plain sheet of paper. During and after the practice task, both the participants were free to ask any clarification questions to the researchers.

The participants were then introduced to the main task: building a blocks structure. The remote helper was presented a pre-built structure situated on a base board. The base board was fixed to the table such that the remote helper can view the structure but not move it in any direction. The local worker was presented with several building blocks and a fixed base board in front of them. The base board of the local worker was fixed horizontally and the base board of the remote helper was fixed vertically. The fact that the two base boards are differently fixed at an angle of 90 degrees is deliberately designed to encourage the remote helper to use the provided rotate screen functionality. This additionally also provided us with the opportunity to see if and how the remote helper oriented the Actionpad to align with the orientation of the Actionport.

When the main task began, the remote helper instructed the local worker using HoloMentor. After successfully completing the task, each participant was presented a post-questionnaire. The participants then switched rooms and thereby switched roles. Another practice task was completed with a different drawing, followed by the main task again. For the second round of the main task, a different building blocks structure was assigned to the remote helper to avoid a learning effect. A second post-questionnaire was filled out by the participants to elaborate on their new role as a remote helper and local worker accordingly.

Lastly, a semi-structured interview was conducted where both researchers and both participants gathered in one large room. The interview contained questions from what the researchers observed during the main tasks and other questions from an interview script (see section 5.3). Each participant took turns to answer each question in their role as a remote helper and a local worker.

5.3 Data Collection

Audio-Video Recordings & Researcher Notes: The interaction between participants during the main task was audio and video recorded using Go Pro cameras in each room and using a screen recorder software to capture the use of HoloMentor on a desktop. The placement of the cameras captured the desktop application's display, the interactions with the Actionpad, the profile of the remote helper, and the upper body and workspace of the local worker. The researchers also took observation notes during the main task.

Post-Interview: After participants performed two main tasks, the researchers and the participants came together for a semi-structured interview that was audio recorded. The interview consisted of semi-structured questions (see Appendix C for full list of questions) as well as additional questions added from what the researchers observed during the main tasks. Each participant took turns to answer each question in their role as a remote helper and a local worker.

[4]We took COVID-19 precaution by following the CDC and university guidelines, and made sure everyone wore a mask, frequently disinfected the area and equipment, and put on a surgical cap to create a barrier between the forehead and fabric headpiece of the HoloLens2 headset.

Questionnaires: (1) A pre-questionnaire that included demographic questions and information about participants' previous experience with AR and the Microsoft HoloLens equipment (see Appendix A). (2) Two post-questionnaires — one for the remote helper and one for the local worker. The questions were designed to gather information about the usefulness of each HoloMentor functionality and how or why they chose to use any functionality when the participant performed each role. See Appendix B for full remote helper and local worker questionnaires. The remote helper post-questionnaire specifically asked how useful they were for using the system's functions (Q1, Q2, and Q3) and how they used the functions to convey remote instructions (Q4 and Q5). The local worker post-questionnaire specifically asked whether the features were useful to perform physical tasks (Q1 and Q2) and how easy to understand the remote helper's action (Q3 and Q4). The questions were in the form of 5-point Likert scales (1= Strongly Disagree, 5= Strongly Agree), rephrased from Fussell et al. [14].

5.4 Data Analysis

Three sets of data were analyzed and triangulated. First, the interaction videos captured during the main task were analyzed. Each significant moment when a remote worker used the Actionport or Actionpad to instruct on the main task was first highlighted in this data. In these moments the first author performed the analytical practice used by Heath et al. [20], ethnomethodologically-informed Conversation Analysis (CA) [22]. This practice includes a detailed reflective analysis that captures utterances, pauses, overlaps, intonations, and visible actions. In the presented vignettes, we indicate the participant number and then the role the participant was playing (e.g. $P5_{RH}$ for Participant 5 playing the Remote Helper, $P6_{LW}$ for Participant 6 playing the Local Worker). Our analytic interests followed the usability of the Actionport and the Actionpad: specifically, the accuracy of use of the pointer, understanding the relationship between the Actionport and the Actionpad, and how and when the remote helper uses the pointer and annotations instead of verbal instructions. We observed participants' behaviors, such as how they communicated through the Actionport, how the remote helper used the Actionpad, and how local workers responded to the information they received.

Second, the interview data was transcribed using Otter.ai[5] and then manually checked for missing data. Using NVivo software, the first and third authors open-coded the transcribed data along with the questionnaire's open-ended answers to identify themes regarding the tools' use. Using selective coding, they then categorized those themes based on common patterns. We focused on a detailed description of the particular qualitative themes that reflect the participants' thoughts regarding the dynamic pointer, annotations, the Actionpad, and the Actionport. For example, we identified how the dynamic pointer affects the remote helper's instructions. By integrating the conversation analysis and the interview analysis, we generated 32 initial open codes that helped us mark the content and organize it in a meaningful way, facilitating analytical thinking in the form of high-level descriptive themes. The final high-level themes were associated with the quality of deictic reference by the pointer and annotations, the relationship between the Actionpad and the Actionport, and point/annotate on live mobile views.

Third, the questionnaires' Likert scale responses were descriptively analyzed and the counts were reported as the median (M), standard deviation (SD), and in diverging bar charts [21]. Although the design of the study does not allow for inferential statistical analysis, we present the descriptive statistics as evidence towards our research questions in support of the primary qualitative analysis.

[5]https://otter.ai/

6 RESULTS

In the following, we delve into the ways HoloMentor helps the remote helpers overcome the challenge caused by pointing and annotating over live mobile views during the local worker's completion of physical tasks. We highlight how the local worker perceived the remote helper's actions through the *Actionport* and how the remote helper leveraged the *Actionpad* to convey their pointing and annotations. As evidence, we present vignettes from the captured videos during the tasks and quotes from the transcribed post-study interviews.

6.1 Alignment of Pointer and Annotations with Actionports

Our first research question was whether the Actionport was able to sufficiently align the remote helper's *dynamic* pointer and annotations and maintain that alignment with a *mobile camera view*. We present evidence through two types of examples: (1) the remote helper's demonstrated satisfaction with the alignment of the pointer and the intended target; and, (2) the local worker's demonstrated ability to understand the locations referred to and instructions provided by the remote helper.

6.1.1 Actionport provides remote workers with good alignment of a dynamic pointer and annotations on intended physical world target. To investigate pointer accuracy during instruction, we first present quantitative data from the post-questionnaire, and then our qualitative analysis of the video data and interviews. Participants acting as *remote helpers* rated highly their perception on the ease with which to convey where they were looking ($M = 5$, $SD = 0.63$ on a 5 point scale), where they were pointing ($M = 5$, $SD = 0.51$), when articulating a spatial measure ($M = 4$, $SD = 0.68$), motion ($M = 4$, $SD = 0.95$), and specific objects ($M = 5$, $SD = 0.74$). As shown in Figure 3, the great majority of responses are above 4.0. The remote helpers tend to rate highly for where they were looking at and where they were pointing.

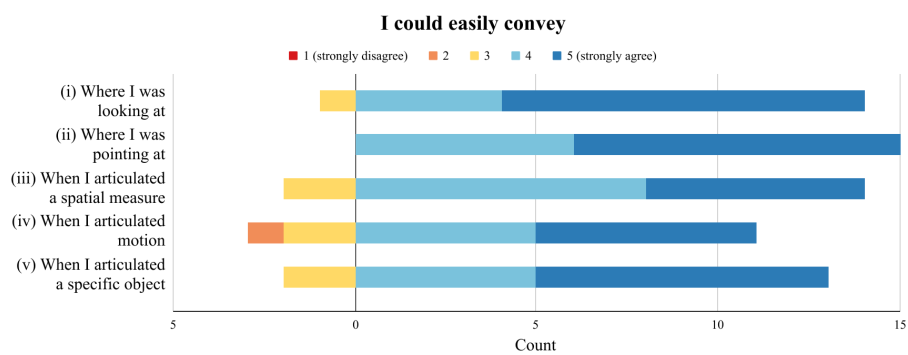
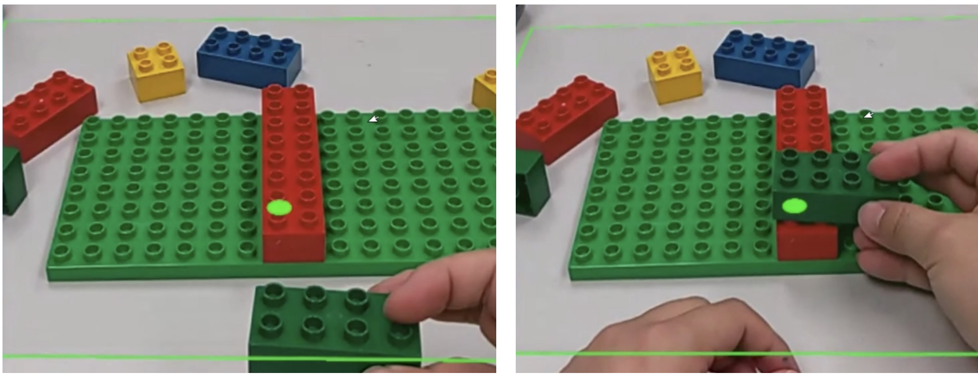


Fig. 3. Remote helpers ratings for the group of questions Q4_R (y-axis). The x-axis corresponds to the counts for each Likert scale level. (Questions (i), (ii), and (v) had 1 n/a; (iv) had 2 n/a)

Through analysis of our video data, we see that as the remote helpers used the pointer and annotation tools to provide instructions, our sixteen participants showed no evidence of being unsatisfied with the alignment of the dynamic pointer/annotation on the view. Two participants did notice that the annotations were drifting slightly due to a technical problem which is related to the accuracy of spatial mapping, although that slight misalignment did not cause the block to be placed incorrectly by the local worker. Overall, though, we saw seamless use of the pointer and annotations to share location or placement instructions.



(a) Remote helper is pointing at a location (b) Local worker is placing the block

Fig. 4. Alignment of Pointer. Vignette 1. (a) A remote helper is pointing at a green block (pointer is a green dot on the red square block). (b) A remote worker is placing the green block where the pointer is indicating.

We begin by showing the use and presenting the reflections on the dynamic pointer. In Vignette 1, the remote helper is moving his pointer towards the location of the next block placement. He is able to show and verbally reference the location without any pause to correct the alignment. In addition, we see his affirmation that the local worker placed the block in the location he indicated.

Vignette 1

P5$_{RH}$: [You're going to place it]
[((Moving the pointer towards block))] (2.0)
It's the second. (3.0) ((Stops moving the pointer and hovers over stud)) In this corner. I mean, top of the red bar. ((Figure 4(a)))

P6$_{LW}$: ((Moves the block towards pointer and begins to place the block down over pointer)) ((Figure 4(b))) Like this?

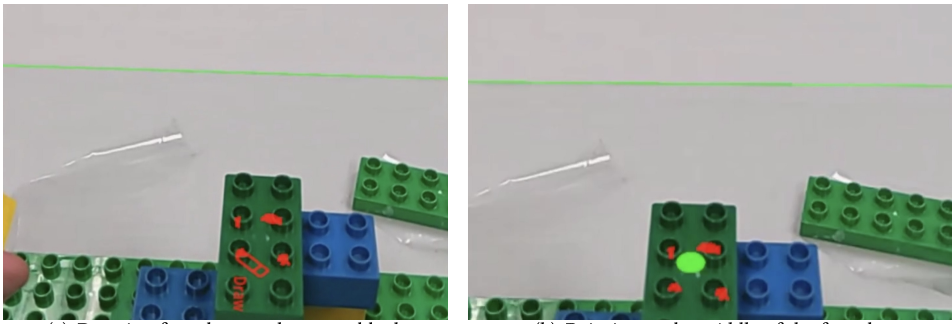
P5$_{RH}$: Yes, like that. Yes.

In Vignette 1, we see the remote helper dynamically moving the pointer over the live mobile view while simultaneously talking to the local worker. Because of the stability of Actionport over the live mobile view, there was a reference frame for the pointer to then be able to move over this live view and accurately point to a location on it. As we previously saw in the reporting of the remote helper's post-questionnaire data, this was perceived to be sufficiently accurate by all of the remote worker participants. Thus, they mainly used the pointer for instructions they provided the local workers. We counted every instances remote helpers used the pointer (113), annotations (83), or verbal only (83) at each instruction moment, and the pointer was used for 40.5% of instructions.

Likewise, during the post-study interviews, we had no participants present examples or complaints that the pointer was not accurate enough for their use. Although lack of evidence is not equal to support, in fact, if discussed at all, the participants would focus on the ability the pointer provided them in making their references clear. For instance, the following are examples of the participant's reflection in reference to the accuracy of the pointer.

"If I was pointing [at] a block with a pointer, he was very easily going to pick it up. And we could move forward to the task. I overall thought that he understood what I was trying to convey." - P5$_{RH}$

"As [a remote helper], I found it much easier to convey what I wanted along with the ability to point to the object that I want." - P11$_{RH}$



(a) Drawing four dots on the green block (b) Pointing at the middle of the four dots

Fig. 5. Accurately aligned annotations. Vignette 2. (a) A remote helper draws the fourth dot on the green block. (b) A remote helper moves the pointer towards the middle of the four dots.

In addition to pointing, annotations were also perceived to be correctly aligned with the intended target. Vignette 2 depicts an occasion where a remote helper drew four dots to guide the local worker in placing a square block. The four dots were fixed on the correct position on the green block, so the local worker placed the square block in that position. After drawing the four dots, the remote helper instructed the placement of the block by moving the pointer to its center.

Vignette 2

P3$_{RH}$: Hold on. Let me try something else. Let me just um. (2.0) How about put it on (5.0) ((Draws a dot on a stud of the green block)) (2.0) ((Draws the second dot on the next stud of the green block)) These four dots at (1.0) ((Draws the third dot)) (2.0) ((Draws the fourth dot)) Put it on this. (2.0) That makes sense. (3.0) ((Figure 5(a)))

P3$_{RH}$: [So place it right here]
[((Moves the pointer to the middle of the dots))] ((Figure 5(b)))

A key assessment of the suitability of our solution is whether it can accurately maintain the placement of a pointer or annotation despite the camera view coming from a HMD, which can introduce both slight natural head movements as well as more pronounced head re-orientation. Thus, we chose to evaluate the user's perception of placement accuracy. What we have shown through three different data types, is the overall suitability of the placement. We say suitability as we cannot definitively measure whether the system was 100% accurate. However, we can say that users did not perceive any significant deviation to cause them to not consider the accuracy to be supportive of all they wished to accomplish. And considering the exactness of some of the intended references and annotations, the system was still perceived to work to a high degree for the fine referencing required.

6.1.2 Actionport provides local workers with good alignment so they can understand location references. The post-questionnaire quantitative data analysis shows that *local workers* also responded with a high rating when asked if they could easily understand where the remote helper was looking ($M = 4.5$, $SD = 0.98$), pointing ($M = 5$, $SD = 0.62$), that they could interpret spacial measures ($M = 4$, $SD = 0.70$), motion ($M = 5$, $SD = 0.69$), and specific objects ($M = 5$, $SD = 0.47$). As shown in Figure 6, the great majority of responses are above 4.0. The local workers tend to rate highly their understanding for where the remote helper was pointing and talking about a specific object.

In addition, all local workers answered that they were able to understand where the remote helper was indicating through the pointer and/or annotations.

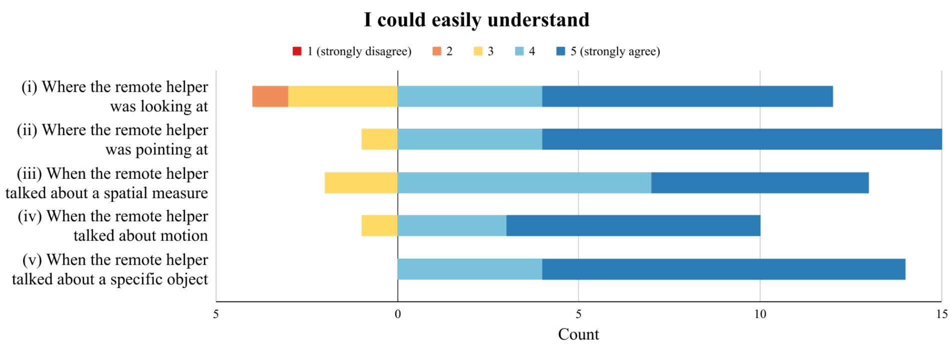
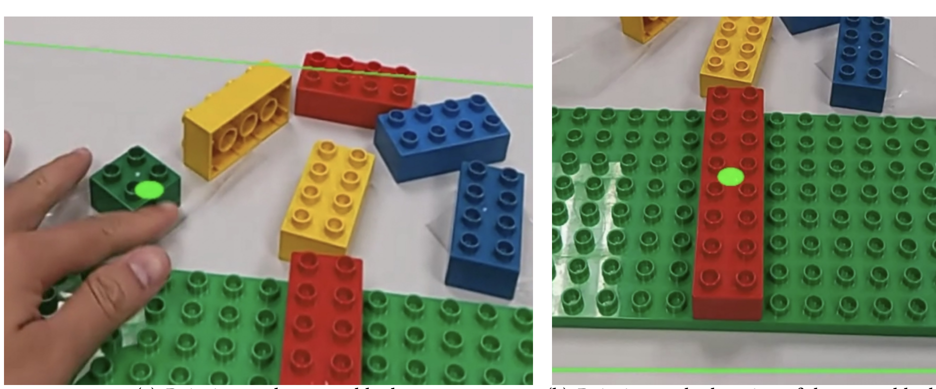


Fig. 6. Local workers ratings for the group of questions Q4_L (y-axis). The x-axis corresponds to the counts for each Likert scale level. (Question (iii) had 1 n/a; (iv) had 5 n/a; and (v) had 2 n/a)

From our qualitative analysis of the video data, we observe that local workers were able to understand the particular targets being referenced by the remote workers in the course of task completion. For example, when referencing a concrete element in the real world (Figure 7(a) and Figure 7(b)) or when aligning a block in the direction an annotation indicated (Figure 8(a) and Figure 8(b)). In the following example, the remote worker is using deictic referencing with the pointer to indicate not only the object that needs to be moved, but also where to move that object.



(a) Pointing at the green block

(b) Pointing at the location of the green block's placement

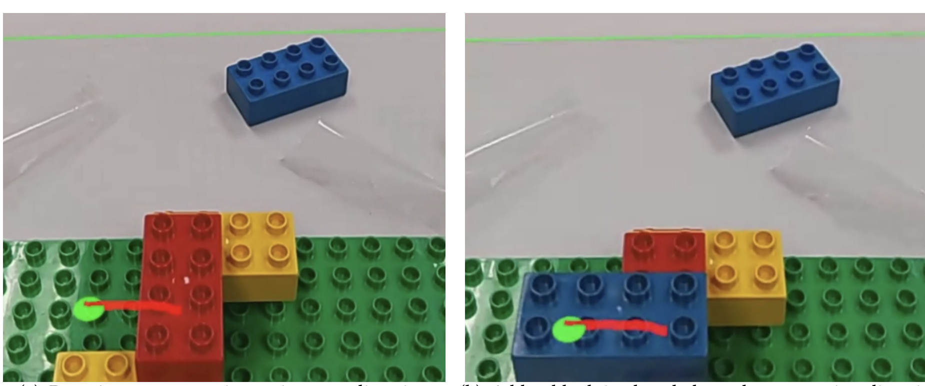
Fig. 7. Indicating movement with the Pointer. Vignette 3. (a) A remote helper is pointing at a green block. (b) A remote helper moves the pointer towards a location where the block should be placed.

Vignette 3

P3$_{RH}$: Then I want you to grab (3.0)
[((Moves the pointer to the green block))]
[This piece here.] ((Figure 7(a)))
P4$_{LW}$: ((Grabs the green block))
P3$_{RH}$: [((Moves the pointer to the middle of the red block))]
[I want you to put it right here.] ((Figure 7(b)))
P4$_{LW}$: ((Puts the block where the pointer is))
P3$_{RH}$: Okay.

What we see in Vignette 3 is the local worker being able to act in conjunction with the instructions being provided. He is able to know exactly which block to move as well as then where to move that block. There is no need for clarifying questions by the local worker, and, more to the point, the collaborative pair is able to be more efficient in their communication by the remote helper using words such as "this" and "here", and these are unambiguous statements for the local worker.

Likewise, in Vignette 4, the remote helper uses the annotation functionality to indicate the orientation of a block's placement and the local worker demonstrates the exactness of the annotation location by his ability to satisfy the request without any clarifying questions.

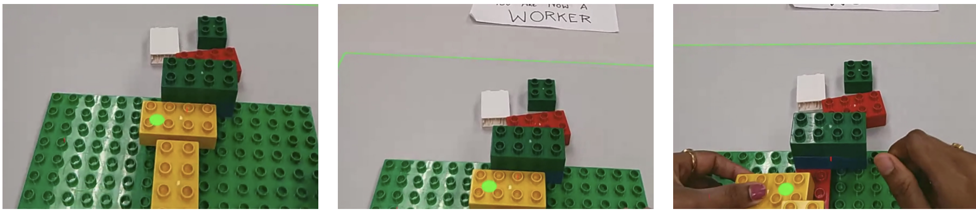


(a) Drawing an annotation to instruct direction (b) A blue block is placed along the annotation direction

Fig. 8. Local worker aligning a block in the direction of the annotation. Vignette 4. (a) A remote helper is drawing an annotation to indicate the direction of a block (Annotation is a red line on the red block). (b) A block is placed by the local worker along the direction indicated.

Vignette 4

P3$_{RH}$: [((Moves the pointer toward a blue block))]
[Go ahead and grab this piece.]
P4$_{LW}$: [((Moves and grabs the blue block))]
P3$_{RH}$: [Yap.] And you're going to put it (2.5) ((Moves the pointer on a red block)) (1.0)
P3$_{RH}$: [Here.]
[((Starts drawing a red line vertically)] (4.0) ((Finishes drawing the line)) ((Figure 8(a)))
P4$_{LW}$: [((Places the block following the direction of the line))] ((Figure 8(b)))
P3$_{RH}$: [Yap. Looks Good.]



(a) Helper points at a location (b) Worker leans towards the pointer (c) Worker moves a yellow block

Fig. 9. Supporting pointing over the live mobile view. Vignette 5. (a) A remote helper is placing the pointer over a yellow block. (b) A local worker moves head pose towards the pointer. (c) The head movement changes while the local worker is moving the yellow block

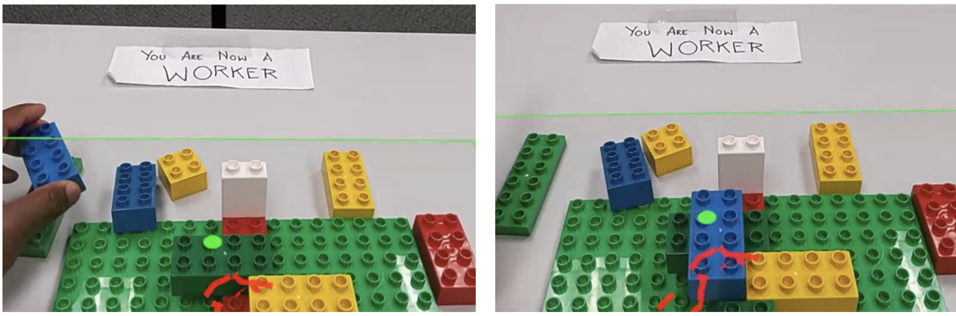
In the example from Vignette 4, the annotation was a strong indication for the local worker. The remote helper only said "here" while drawing the annotation, but the local worker saw the starting position and direction of the annotation and placed the block correctly. There was no need for a conversation to confirm the location or direction.

Finally, in this next example, we show the unique benefit of our approach. Actionport is meant to support pointing over a moving camera view in order to support complex collaborative interactions. Vignette 5 shows a pointing action while at the same time the local person moves their head, changing the view. The result is nonetheless an interlacing of talk and action by both parties that is efficient and effective in getting the work done, without significant overlap. This reinforces the communication benefit of being able to point/annotate on live mobile views during task execution.

Vignette 5

P13$_{LW}$: ((Moves her head closer to the building block structure.)) (3.0) ((Places a building block))
P12$_{RH}$: Okay, move the yellow. (.) ((Points at the bottom of a horizontally placed yellow block)) (2.0) Close towards two studs, blue and yellow. ((Figure 9(a)))
P13$_{LW}$: ((moves her head pose up)) ((Figure 9(b)))
P12$_{RH}$: ((Pointer still in intended location)) Leave three positions.
P13$_{LW}$: ((Moves the building block to the right))
P12$_{RH}$: [((Moves the pointer from the building block to the left side))]
[Same thing. Do the other side.]
P13$_{LW}$: ((Still tries to putting the building block on the left))
P12$_{RH}$: [No, the opposite side.]
[((Points to the building block again))] (.)
P12$_{RH}$: [((Moves the pointer to the left.))]
P13$_{LW}$: [((Moves the building block to where the pointer is))] ((Figure 9(c)))
P12$_{RH}$: ((Keeps pointing)) Yeah, awesome.

In this case, we showed that the pointer had sufficient alignment with both side to side head movement as well as when getting closer or further away from the physical space. This example also showed how useful this is for the collaborators as the remote helper can continue to provide deictic referencing while the local worker is moving their head. In other words, by supporting pointing over a live mobile video during instruction, we have been able to support the interlacing of talk and deictic referencing by the remote helper and natural head/body movement for closer inspection as well as physical object movement for the task at hand by the local worker.



(a) Pointing on the first row of a green block (b) Pointer is projected on the second row of a blue block

Fig. 10. Obfuscation of the pointer's location. Vignette 6. (a) A remote helper is pointing at the first row of a green block. (b) A blue block is placed on the pointer's location and the pointer is placed on the second row of the blue block.

We finish this section by noting that we observed a hindrance inherent to world-stabilized annotations during physical tasks. Guiding a physical task will inevitably lead to changes in the physical world as the local worker modifies the world (e.g. assembles equipment or dissects tissues). What we observed is instances of local workers associating verbal instructions with annotations that remote helpers placed at a previous moment in time when the state of the world was different.

Vignette 6

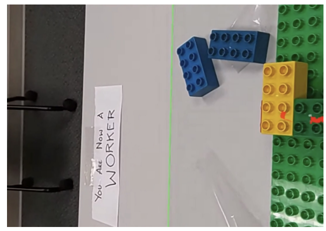
P9$_{RH}$: ((Moves the pointer on a green block)) The first hole of the blue block on the top should align with the pointer that I am pointing right now. (Figure 10(a))
P10$_{LW}$: ((Places the blue block where the pointer is pointing)) (Figure 10(b))
P9$_{RH}$: ((Sees the pointer is positioned on the second row of the blue block)) Yeah, the first hole actually.

We see in Vignette 6, that the worker correctly placed a building block given an annotation, then the remote helper gives a new instruction, but the old annotation is now on top of the newly-placed block, and as the local worker now associates the new instruction with the old annotation, the action leads to a misplacement of the next building block. Thus, the dynamic nature of the physical world being augmented with instructions brings to light a new challenge to overcome when designing for remote annotation of live mobile views.

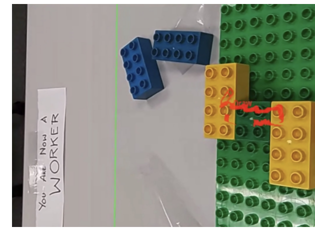
6.2 Use of the Actionpad to Point and Annotate

Our second research question was whether the remote workers were able to use the Actionpad to effectively point and annotate in the Actionport. We show evidence that the remote helpers were able to effectively understand how to use the Actionpad to move the pointer and draw annotations. We present evidence that this was achieved through examples of the remote helpers demonstrating the ability to point and annotate on the Actionpad independently of the live mobile view. We also present further findings on how the remote workers established their understanding of the relationship between the Actionpad screen and resulting movements of the pointer on the Actionport as well as the relationship of the orientation of the Actionpad with the orientation of the Actionport.

(a) Drawing an annotation through the Actionpad



(b) The pointer goes off the view



(c) Continued drawing when the view changes

Fig. 11. Drawing an annotation independently of the live mobile view. Vignette 7. (a) A remote helper is drawing through the Actionpad. (b) The pointer is moved out of the live mobile view. (c) The annotation is drawn seamlessly. Note that in figures (b) and (c), the remote helper had previously rotated the view.

6.2.1 Independent Interaction for Pointer and Annotations through Actionpad. The first thing we observed is that the remote helpers controlled the Actionport pointer with the Actionpad consistently and their actions were not affected by the live mobile view. We did not observe moments when remote helpers complained of disorientation when interacting with the the Actionpad. In addition, when the pointer was not visible in the live mobile view due to the local worker's head movement, the remote helpers could continue pointing and annotating while asking the local worker to move their head back to make the pointer visible to them again.

Vignette 7

P3$_{RH}$: ((Alternating looking at the pre-built blocks structure and the HoloMentor desktop application display)) We're going to put it. (1.0)
[((Moves the pointer to the block at the far right of the screen))] (Figure 11(a))
[Here.] (.)

P3$_{RH}$: ((Begins to draw a line - annotation continues out of the view in the desktop's video display window - continues drawing)) (Figure 11(b)) (3.0) Look down a little bit.

P4$_{LW}$: ((Looks down))

P3$_{RH}$: ((Continues movement of fingers on the ActionPad and then ends)) Put it right here. (Figure 11(c))

In this Vignette 7 and Figure 11(a), we see that while the remote helper was drawing, he was able to move the pointer on the tablet (Actionpad), but the annotation went out of view (Figure 11(b)). Therefore, the remote helper asked the local worker to look down, and nonetheless the pointer stayed in the same position regardless of the view changing. As soon as the remote helper saw the pointer in the live view as the local worker moved their head, he moved the pointer to continue drawing the annotation (Figure 11(c)). This shows that the Actionpad enables the remote helper to interact with a pointer or draw independently of the local worker's head movement.

6.2.2 Remote Workers Establish Alignment between Actionport and Actionpad. In all cases, we observed that remote helpers first aimed to establish an alignment between their interactions on the Actionpad and the movement of their pointer in the Actionport before providing an instruction. When they wanted to move the pointer in the Actionport, first, they checked to see if their finger was positioned where they wanted on the Actionpad. Finally, while moving the pointer, they looked at whether the pointer actually moved to the location they intended in the Actionport.

(a) Looking down towards the Actionpad

(b) Looking towards the movement of the pointer in the Actionport

Fig. 12. Establishing alignment between Actionpad and Actionport. Vignette 8. (a) A remote helper is looking towards the Actionpad and moving their finger towards it. (b) The remote helper is looking at the Actionport while moving his finger on the Actionpad.

Vignette 8

P16$_{RH}$**:** ((Looks towards the screen)) (1.5) ((Looks down towards the Actionpad and touches the tablet)) (.) (Figure 12(b))

P16$_{RH}$**:** ((Looks back up towards the screen)) (.) ((Moves the pointer to the left)) (Figure 12(b))

In Vignette 8, we see the remote helper looked at the screen first, then the Actionpad, and then moved his finger while looking at the screen again before he began deictic referencing. Therefore, for a moment, the remote helper took the time to first check whether the interaction through the Actionpad corresponds to the movement of the Actionport's pointer. However, after learning how the two interfaces aligned, the remote helpers rarely looked at the tablet again when pointing and kept their gaze fixed on the desktop monitor when moving the pointer or drawing via the Actionpad.

6.2.3 Coupling Rotated Actionport with Actionpad. We observed that remote helpers understood the relationship between the Actionpad and the Actionport. In Vignette 9, the remote helper had previously rotated the video view counterclockwise and rotated the Actionpad accordingly to align. After a while, the remote helper makes the decision to rotate the video view back to the original orientation.

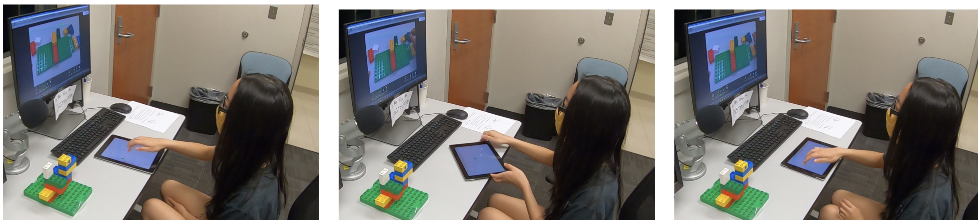
Vignette 9

P7$_{RH}$**:** [((Rotates her view on the desktop application's video display window clockwise))]
[Actually, probably think this is better.] (1.0)
((Looks at pre-built structure)) Yeah, Okay. (.) So, then you wanna take. (.) ((Touches tablet screen)) (2.0)

P7$_{RH}$**:** [this.]
[((Moves her finger on the Actionpad screen to the left))] (.) (Figure 13(a))

P7$_{RH}$**:** Where? Where is my pointer? (.) ((Sees the pointer is moving upwards as she is moving her finger left)) (2.0) hold on. (.)
[((Turns the tablet clockwise))] (Figure 13(b))
[Okay. There it is. That's fine. Makes sense.] (1.0)

P7$_{RH}$**:** [((Moves the pointer to the red block on the left))]
[So, this block right here.] (Figure 13(c))



(a) Misalignment of orientation between the Actionport and the Actionpad

(b) Rotating the Actionpad

(c) Realignment of orientation between the Actionport and the Actionpad

Fig. 13. Recognizing the relationship between the Actionpad and the Actionport. Vignette 9. (a) A remote helper is moving the pointer to the left in the Actionpad, and the pointer is moving upward on the screen. (b) The remote helper is rotating the Actionpad. (c) The remote helper recognizes the pointer is moving in the same direction as her hand gesture.

This is the moment that the remote helper rotated back to the original view but she did not rotate her tablet at the same time. However, we see that as soon as she moved the pointer and saw that the pointer moved in a different direction, she recognized she needed to also rotate the tablet and then checked again if the pointer in the Actionport moved in the direction she wanted. Vignette 9 shows the way the remote helper couples the direction of the Actionport with the Actionpad. All remote helpers who used the rotation function went through the same coupling process.

7 DISCUSSION

In this study, we investigated HoloMentor consisting of an Actionport and an Actionpad, an embodiment that enables remote helpers to produce world-stabilized pointing and annotations without the need to "freeze" the remote view. The main takeaway is the demonstrated advantages over current approaches for producing world-stabilized annotations: in contrast to interfaces where remote helpers can only interact by taking a snapshot of the view first, our approach (1) supports pointing and thus deictic referencing in instruction, and, (2) supports live pointing and annotations while the view changes, and thus seamless communication without the need to disrupt the local work. We now summarize and discuss our findings.

First, our results showed the Actionport provided a suitable alignment of pointers and annotations to the intended target of the remote helper. In addition, none of the remote helpers indicated a loss of control of the pointer as the local workers moved their heads to explore the space. Because of this, the world-stabilized Actionport provided an embodiment for the remote helper to accurately provide deictic referencing and movement direction, without being affected by the local worker's head movement. Above all, pointing and annotating in the live mobile view had a positive effect on maintaining shared understanding between the remote helper and the local worker. For example, when the remote helper referred to a block and moved the pointer to a specific position, the local worker followed the pointer movement while holding the block almost simultaneously and then placing it in the intended position. This is interesting considering prior systems for producing world-stabilized annotations on mobile views (e.g. HoloLens Dynamics 365 Remote Assist and [17]) have focused only on supporting annotations, not pointing. However, what we see here is that, if given both the option of pointing or annotating, users will choose dynamic pointing for many references, which is a prime motivator of this project. More so, the Actionport provided a mechanism for achieving *accurate* deictic referencing. In essence, it would not be sufficient if we had provided a pointer that participants did not feel to be accurate enough for effective use.

Second, we identified the ability of the remote helper in using the Actionpad and that the Actionpad enabled remote helpers to continue manipulating the pointer regardless of changes in the live mobile view. Moreover, this was still true many times when the pointer was off-screen. Here, the remote helper asked the local worker to look back at the pointer in order to see its position, but in the meantime, they still continued interacting on the Actionpad. Interestingly, remote helpers quickly and intuitively understood the relation between their actions on the Actionpad and the result on the Actionport. For example, as there is a one-to-one correspondence between the XY coordinates on the Actionpad and Actionport, when the remote helper rotated the live mobile view, the Actionpad could also rotate in the same direction to ease the mapping process. Even if the remote helper rotated the video view first and then forgot to rotate the Actionpad, they did so immediately when realizing that the movement (point or annotate) on the Actionpad did not match the expected outcome on the Actionport pointer.

7.1 Interaction for Creating World-Stabilized Annotations

Previous work [9, 16, 17, 28] proposed systems to support world-stabilized annotations by freezing the live view. Fakourfar et al. [9] revealed that the advantage of freezing is that it helps to draw annotations when the local worker changes the view. However, at the same time, they argued that the gap that occurs when the video stops and then restarts can cause disorientation and confusion for the remote helper. Our interaction technique supports stabilized annotating on the live mobile view without freezing the view, which our study shows enabled smooth communication between collaborators and enabled the remote helper to immediately perceive their local worker's action. Ens et al. [8] points out that we need to provide remote experts with mechanisms to navigate the local worker's space independently of the local worker's view — as when the local worker uses a HMD, they control the view. As beneficial this may be for *visualizing* the remote world, it may actually be disadvantageous when *acting* on the remote world, creating misalignment in communication and difficulty for providing immediate feedback.

However, the Actionpad also had limitations in that the remote helpers do not really know where they are drawing when they put their fingers down on the tablet. In our study, participants were hesitant to draw on the Actionpad, as they had just a blank canvas. Therefore, they had to compensate by first pointing somewhere in the Actionpad, then adjusted by looking at the result in the Actionport. Future work can explore how this can be mitigated through different approaches, for example (1) feedforward mechanism: a hovering function where the remote helper first adjusts their pointer position in a personal view and when they are ready to communicate to the local worker, they press the screen (e.g. force touch) to perform the action; (2) show a 2D Reflection of the physical workspace on the Actionpad using 3D reconstruction like Gauglitz et al. [17]: the image of the physical environment in which the Actionport is placed is reconstructed and it is reflected in Actionpad in real time. Lastly, we observed that formerly-placed annotations can be misinterpreted as the physical world changes beneath them. Future work can explore how to mitigate this problem, for example automatically removing the annotation when the real world is updated, or having mechanisms that create semantic links between annotations and objects (e.g. by drawing a line between them).

7.2 Dynamic Interaction for Remote Instruction in Live Mobile View

We found that the Actionport provided a stable frame for the dynamic pointer to reference the live mobile view. Because of this, the remote helper frequently used the dynamic pointer to provide deictic referencing. We showed that, if given the option of pointing or annotating, the remote helpers chose dynamic pointing for many references. Fakourfar et al. [9] highlighted that the dynamic pointer is more beneficial than stabilized annotations on transient, short, procedural collaboration tasks. In addition, in situations where the live mobile view continues to change dynamically, immediate interaction is needed rather than stabilized annotation. Fussell et al. [14] suggested an automatic erase function for remote instruction, in which annotations disappear after a few seconds, was more efficient than a manual erase function. Kim et al. [29] also pointed out that the drawing annotations need to be erased after completing each step of a task because the remaining annotations would cause confusion.

Our system provided a manual erase function through the desktop application's toolbar, but some remote helpers did not use them, even though confusion could arise due to the remaining annotations. We posit that this is because remote helpers interacted mainly through the Actionpad. Therefore, an automatic deletion function or the function of editing or erasing the previous annotations in the Actionpad directly should be considered. Finally, while HoloMentor lets the remote helper reliably control the dynamic pointer in the live mobile view, the local worker does not have control over the pointer or annotations. Mentis et al. [37] have argued for enabling equal ability to interact on both sides. Therefore, it is also necessary to study the interface or cooperative social process that enables local workers to point or control annotations.

7.3 Supporting Conversational Grounding and Situational Awareness

HoloMentor mainly supports maintaining conversational grounding and situational awareness by real-time pointing and annotating on the live mobile view. Fussell et al. [14] suggested that the use of the four types of gestures (deictic, iconic representations, spatial/distance, and kinetic/motion) facilitated the conversational grounding. HoloMentor provides deictic and kinetic/motion gestures through the dynamic pointer and spatial/distance and kinetic/motion gestures through annotations in the Actionport. Furthermore, Kraut et al. [31] highlighted that shared visual cues facilitated situational awareness so that the remote helpers provided better instruction. In our study, the remote helper can perceive the local worker's action and change the pointing/annotating through the live mobile view, which supports the remote helper in maintaining situational awareness in a complex and dynamic environment.

Additionally, HoloMentor supports not only deictic instruction in the live mobile view but also it is compatible with setups commonly used in which a remote helper uses a desktop and a monitor. Previous studies have developed systems that provide deictic instruction through 2.5 hands interaction [40] using a leap motion or hand gesture interactions using a virtual reality (VR) HMD [38]. However, in remote instruction where the local worker is working with physical objects, AR could be more beneficial than VR because the local worker can see the objects directly. On the other hand, to see the physical objects through VR, not only would the system need to reconstruct the environment, but also the reconstructed environment in VR is not as accurate as of the real world. Since the video view the remote helpers see is streamed from the local worker's mobile camera, the view may not be the best view for giving instructions. Therefore, remote helpers may need to independently manipulate the view.

However, currently remote helpers can only explore their view in the XY plane or by zooming or rotating. There is a gap because the shared workspace viewed by local workers is 3D. Exploring with zoom and rotation is too limited in degrees of independent exploration. This has been mitigated by asking the local worker to change head position. When the remote helpers were not able to see the other side of the local worker's building blocks structure even with rotation and zooming, they asked their local worker to move their head pose. In this way, remote helpers tried to form the best view to explain by integrating the local worker's point of view.

8 CONCLUSION

In this paper, our contribution was the presentation of HoloMentor — an AR HMD-based collaborative system designed for remote instruction over live mobile views during local worker's completion of physical tasks — and empirical results of a laboratory study to evaluate its effectiveness and use. First, HoloMentor realizes pointing and annotating on a world-stabilized Actionport that lets the remote helper communicate with the local worker smoothly even though the view of the local worker's environment is not fixed. We were able to show that the Actionport is able to sufficiently align the remote helper's dynamic pointer and annotations and maintain that alignment with a mobile camera view. Second, the Actionpad entails a decoupling of interaction on a tablet (action) from its visualization in the live mobile view (perception). These two innovations tackle fundamental challenges that need addressing for collaborative systems based on AR through HMDs, where the view of the local worker's environment is mobile. Our approach stands in contrast with prior work whose solution was to freeze the view and thus could not support deictic reference. This is an important distinction that demonstrates the innovative necessity of identifying effective collaborative functionality with new interactive devices. In the future, we envision HoloMentor can be used as a playground to study and further understand remote instruction under the new challenges that mobile views bring. For example, we would see how HoloMentor affects one-to-many remote collaboration such as telementoring situations where multiple local paramedics use live mobile views and remote helpers guide them. We could investigate how a remote helper manages the screens from multiple local workers and how they use HoloMentor to deliver different instructions to each.

ACKNOWLEDGMENTS

This work was supported in part by NSF IIS-1552837 and NSF BCS-2026510.

REFERENCES

[1] Steven Arild Wuyts Andersen, Peter Trier Mikkelsen, Lars Konge, Per Cayé-Thomasen, and Mads Sølvsten Sørensen. 2016. Cognitive load in distributed and massed practice in virtual reality mastoidectomy simulation. The Laryngoscope 126, 2 (2016), E74–E79.

[2] David Anton, Gregorij Kurillo, and Ruzena Bajcsy. 2018. User experience and interaction performance in 2D/3D telecollaboration. Future Generation Computer Systems 82 (2018), 77–88.

[3] Ignacio Avellino, Gilles Bailly, Mario Arico, Guillaume Morel, and Geoffroy Canlorbe. 2020. Multimodal and Mixed Control of Robotic Endoscopes. In Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20). Association for Computing Machinery, Honolulu, HI, USA, 1–14. https://doi.org/10.1145/3313831.3376795

[4] Ignacio Avellino, Gilles Bailly, Geoffroy Canlorbe, Jérémie Belgihti, Guillaume Morel, and Marie-Aude Vitrani. 2019. Impacts of Telemanipulation in Robotic Assisted Surgery. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19). ACM, New York, NY, USA, 583:1–583:15. https://doi.org/10.1145/3290605.3300813 event-place: Glasgow, Scotland, UK.

[5] Martin Bauer, Gerd Kortuem, and Zary Segall. 1999. "Where Are You Pointing At?" A Study of Remote Collaboration in a Wearable Videoconference System. In Proceedings of the 3rd IEEE International Symposium on Wearable Computers (ISWC '99). IEEE Computer Society, USA, 151. https://doi.org/10.1109/ISWC.1999.806696

[6] Henry Chen, Austin S. Lee, Mark Swift, and John C. Tang. 2015. 3D Collaboration Method over HoloLens™; and Skype™; End Points. In Proceedings of the 3rd International Workshop on Immersive Media Experiences (ImmersiveME '15). Association for Computing Machinery, New York, NY, USA, 27–30. https://doi.org/10.1145/2814347.2814350

[7] Sicheng Chen, Miao Chen, Andreas Kunz, Asim Evren Yantaç, Mathias Bergmark, Anders Sundin, and Morten Fjeld. 2013. SEMarbeta: mobile sketch-gesture-video remote support for car drivers. In Proceedings of the 4th Augmented Human International Conference (AH '13). Association for Computing Machinery, New York, NY, USA, 69–76. https://doi.org/10.1145/2459236.2459249

[8] Barrett Ens, Joel Lanir, Anthony Tang, Scott Bateman, Gun Lee, Thammathip Piumsomboon, and Mark Billinghurst. 2019. Revisiting collaboration through mixed reality: The evolution of groupware. International Journal of Human-Computer Studies 131 (2019), 81–98.

[9] Omid Fakourfar, 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 (CHI '16). Association for Computing Machinery, New York, NY, USA, 1548–1560. https://doi.org/10.1145/2858036.2858171

[10] Yuanyuan Feng, Hannah McGowan, Azin Semsar, Hamid R. Zahiri, Ivan M. George, Adrian Park, Andrea Kleinsmith, and Helena Mentis. 2019. Virtual pointer for gaze guidance in laparoscopic surgery. Surgical Endoscopy 34 (Oct. 2019), 3533–3539. https://doi.org/10.1007/s00464-019-07141-x

[11] Yuanyuan Feng, Hannah McGowan, Azin Semsar, Hamid R Zahiri, Ivan M George, Adrian Park, Andrea Kleinsmith, and Helena Mentis. 2020. Virtual pointer for gaze guidance in laparoscopic surgery. Surgical endoscopy 34, 8 (2020), 3533–3539.

[12] Yuanyuan Feng, Hannah McGowan, Azin Semsar, Hamid R Zahiri, Ivan M George, Timothy Turner, Adrian Park, Andrea Kleinsmith, and Helena M Mentis. 2018. A virtual pointer to support the adoption of professional vision in laparoscopic training. International journal of computer assisted radiology and surgery 13, 9 (2018), 1463–1472.

[13] Yuanyuan Feng, Hannah McGowan, Azin Semsar, Hamid R. Zahiri, Ivan M. George, Timothy Turner, Adrian Park, Andrea Kleinsmith, and Helena M. Mentis. 2018. A virtual pointer to support the adoption of professional vision in laparoscopic training. International Journal of Computer Assisted Radiology and Surgery 13, 9 (Sept. 2018), 1463–1472. https://doi.org/10.1007/s11548-018-1792-9

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

[15] Steffen Gauglitz, Cha Lee, Matthew Turk, and Tobias Höllerer. 2012. Integrating the physical environment into mobile remote collaboration. In Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services (MobileHCI '12). Association for Computing Machinery, New York, NY, USA, 241–250. https://doi.org/10.1145/2371574.2371610

[16] Steffen Gauglitz, Benjamin Nuernberger, Matthew Turk, and Tobias Höllerer. 2014. In touch with the remote world: remote collaboration with augmented reality drawings and virtual navigation. In Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology (VRST '14). Association for Computing Machinery, New York, NY, USA, 197–205. https://doi.org/10.1145/2671015.2671016

[17] Steffen Gauglitz, Benjamin Nuernberger, Matthew Turk, and Tobias Höllerer. 2014. World-stabilized annotations and virtual scene navigation for remote collaboration. In Proceedings of the 27th annual ACM symposium on User interface software and technology - UIST '14. ACM Press, Honolulu, Hawaii, USA, 449–459. https://doi.org/10.1145/2642918.2647372

[18] Darren Gergle, Robert E Kraut, and Susan R Fussell. 2013. Using visual information for grounding and awareness in collaborative tasks. Human–Computer Interaction 28, 1 (2013), 1–39.

[19] Carl Gutwin and Saul Greenberg. 2002. A descriptive framework of workspace awareness for real-time groupware. Computer Supported Cooperative Work (CSCW) 11, 3 (2002), 411–446.

[20] Christian Heath, Paul Luff, Hideaki Kuzuoka, Keiichi Yamazaki, and Shinya Oyama. 2001. Creating Coherent Environments for Collaboration. In ECSCW 2001: Proceedings of the Seventh European Conference on Computer Supported Cooperative Work 16–20 September 2001, Bonn, Germany, Wolfgang Prinz, Matthias Jarke, Yvonne Rogers, Kjeld Schmidt, and Volker Wulf (Eds.). Springer Netherlands, Dordrecht, 119–138. https://doi.org/10.1007/0-306-48019-0_7

[21] Richard Heiberger and Naomi Robbins. 2014. Design of diverging stacked bar charts for Likert scales and other applications. Journal of Statistical Software 57 (2014), 1–32.

[22] Alexa Hepburn and Galina B Bolden. 2013. The conversation analytic approach to transcription. The handbook of conversation analysis 1 (2013), 57–76.

[23] Weidong Huang, Leila Alem, Franco Tecchia, and Henry Been-Lirn Duh. 2018. Augmented 3D hands: a gesture-based mixed reality system for distributed collaboration. Journal on Multimodal User Interfaces 12, 2 (2018), 77–89.

[24] Weidong Huang, Seungwon Kim, Mark Billinghurst, and Leila Alem. 2019. Sharing hand gesture and sketch cues in remote collaboration. Journal of Visual Communication and Image Representation 58 (2019), 428–438.

[25] Brennan Jones, Anna Witcraft, Scott Bateman, Carman Neustaedter, and Anthony Tang. 2015. Mechanics of Camera Work in Mobile Video Collaboration. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15). Association for Computing Machinery, New York, NY, USA, 957–966. https://doi.org/10.1145/2702123.2702345

[26] Seungwon Kim, Gun Lee, Mark Billinghurst, and Weidong Huang. 2020. The combination of visual communication cues in mixed reality remote collaboration. Journal on Multimodal User Interfaces 14, 4 (2020), 321–335.

[27] Seungwon Kim, Gun Lee, Nobuchika Sakata, and Mark Billinghurst. 2014. Improving co-presence with augmented visual communication cues for sharing experience through video conference. In 2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR). IEEE Computer Society, USA, 83–92. https://doi.org/10.1109/ISMAR.2014.6948412

[28] Seungwon Kim, Gun A. Lee, Sangtae Ha, Nobuchika Sakata, and Mark Billinghurst. 2015. Automatically Freezing Live Video for Annotation during Remote Collaboration. In Proceedings of the 33rd Annual ACM Conference Extended

Abstracts on Human Factors in Computing Systems (CHI EA '15). Association for Computing Machinery, New York, NY, USA, 1669–1674. https://doi.org/10.1145/2702613.2732838

[29] Seungwon Kim, Gun A. Lee, and Nobuchika Sakata. 2013. Comparing pointing and drawing for remote collaboration. In 2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR). IEEE Computer Society, USA, 1–6. https://doi.org/10.1109/ISMAR.2013.6671833

[30] David Kirk, Andy Crabtree, and Tom Rodden. 2005. Ways of the Hands. In ECSCW 2005, Hans Gellersen, Kjeld Schmidt, Michel Beaudouin-Lafon, and Wendy Mackay (Eds.). Springer Netherlands, Dordrecht, 1–21. https://doi.org/10.1007/1-4020-4023-7_1

[31] Robert E. Kraut, Susan R. Fussell, and Jane Siegel. 2003. Visual Information As a Conversational Resource in Collaborative Physical Tasks. Hum.-Comput. Interact. 18, 1 (June 2003), 13–49. https://doi.org/10.1207/S15327051HCI1812_2

[32] Morgan Le Chénéchal, Thierry Duval, Valérie Gouranton, Jérôme Royan, and Bruno Arnaldi. 2019. Help! I Need a Remote Guide in my Mixed Reality Collaborative Environment. Frontiers in Robotics and AI 6 (2019), 106.

[33] L. Lingard, S. Espin, S. Whyte, G. Regehr, G. R. Baker, R. Reznick, J. Bohnen, B. Orser, D. Doran, and E. Grober. 2004. Communication failures in the operating room: an observational classification of recurrent types and effects. BMJ Quality & Safety 13, 5 (Oct. 2004), 330–334. https://doi.org/10.1136/qshc.2003.008425

[34] Stephan Lukosch, Heide Lukosch, Dragoş Datcu, and Marina Cidota. 2015. Providing information on the spot: Using augmented reality for situational awareness in the security domain. Computer Supported Cooperative Work (CSCW) 24, 6 (2015), 613–664.

[35] Sophie Maria, Solène Lambert, and Ignacio Avellino. 2022. From Déjà vu to Déjà vécu: Reliving Surgery in Post-Operative Debriefing. In 2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW). IEEE Computer Society, Los Alamitos, CA, USA, 462–465. https://doi.org/10.1109/VRW55335.2022.00102

[36] Helena M. Mentis, Ignacio Avellino, and Jwawon Seo. 2022. AR HMD for Remote Instruction in Healthcare. In 2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW). IEEE Computer Society, Los Alamitos, CA, USA, 437–440. https://doi.org/10.1109/VRW55335.2022.00096

[37] Helena M. Mentis, Yuanyuan Feng, Azin Semsar, and Todd A. Ponsky. 2020. Remotely Shaping the View in Surgical Telementoring. In Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20). Association for Computing Machinery, Honolulu, HI, USA, 1–14. https://doi.org/10.1145/3313831.3376622

[38] Thammathip Piumsomboon, Arindam Dey, Barrett Ens, Gun Lee, and Mark Billinghurst. 2019. The effects of sharing awareness cues in collaborative mixed reality. Frontiers in Robotics and AI 6 (2019), 5.

[39] Aaron Stafford, Wayne Piekarski, and Bruce Thomas. 2006. Implementation of god-like interaction techniques for supporting collaboration between outdoor AR and indoor tabletop users. In Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR '06). IEEE Computer Society, USA, 165–172. https://doi.org/10.1109/ISMAR.2006.297809

[40] Peng Wang, Shusheng Zhang, Xiaoliang Bai, Mark Billinghurst, Weiping He, Mengmeng Sun, Yongxing Chen, Hao Lv, and Hongyu Ji. 2019. 2.5 DHANDS: a gesture-based MR remote collaborative platform. The International Journal of Advanced Manufacturing Technology 102, 5 (2019), 1339–1353.

[41] Douglas A. Wiegmann, Andrew W. ElBardissi, Joseph A. Dearani, Richard C. Daly, and Thoralf M. Sundt. 2007. Disruptions in surgical flow and their relationship to surgical errors: An exploratory investigation. Surgery 142, 5 (Nov. 2007), 658–665. https://doi.org/10.1016/j.surg.2007.07.034

A PRE-QUESTIONNAIRES

	Questions	Options
Demographic	Age	[ ] 18 - 20 years [ ] 21 - 25 years [ ] 26 - 30 years [ ] 31 - 35 years [ ] 36 - 40 years [ ] 41 and above
	Gender	[ ] Male [ ] Female [ ] Prefer not to say
Education	What is the highest degree or level of education you have completed? (If you are currently enrolled in school, please indicate the degree you are pursuing)	[ ] Less than a high school diploma [ ] High school degree or equivalent (e.g. GED) [ ] Some college, no degree [ ] Associate degree (e.g. AA, AS) [ ] Bachelor's degree (e.g. BA, BS) [ ] Master's degree (e.g. MA, MS, MEd) [ ] Professional degree (e.g. MD, DDS, DVM) [ ] Doctorate (e.g. PhD, EdD)
	Specify your major:	
	Are you a part-time or a full-time student?	
Familiarity with Technology	How many times a week do you use a computer?	[ ] Never [ ] Once in a week [ ] 2-3 times a week [ ] More than 2-3 times a week [ ] Everyday
	Have you used any of the Augmented Reality (AR) devices (e.g., Google Glass, Magic Leap, Hololens 2) before?	[ ] No [ ] Microsoft Hololens [ ] Magic Leap [ ] Epson Moverio [ ] Google Glass [ ] Other ______________
	If you answered yes in your previous question, how many times do you use an AR device?	[ ] Once in my life [ ] Once a year [ ] Once a month [ ] Once a week

Table 1. Pre-questionnaire that included demographic questions and information about participants' previous experience

B POST-QUESTIONNAIRES

	Questions	Likert-Scales
Q1_R	I found the __________ function easy to use: (Pointing, Drawing, Zooming, Rotation)	1 (strongly disagree) to 5 (strongly agree)
Q2_R	I found the __________ function useful: (Pointing, Drawing, Zooming, Rotation)	1 (strongly disagree) to 5 (strongly agree)
Q3_R	How did you use each function to provide instructions to the local worker? Please describe how you used each function to support your instructions, for example when asking to stack blocks, move a block, rotate a block, stick together two blocks, separate blocks, etc. Pointing: __________ Drawing: __________ Zooming: __________ Rotation: __________	
Q4_R	I could easily convey:	
	(i) Where I was looking at: Through which function(s)?: ____	1 (strongly disagree) to 5 (strongly agree)
	(ii) Where I was pointing at: Through which function(s)?: ____	1 (strongly disagree) to 5 (strongly agree)
	(iii) When I articulated a spatial measure (e.g., place the lego five studs away): Through which function(s)?: ____	1 (strongly disagree) to 5 (strongly agree)
	(iv) When I articulated motion (e.g., rotate the lego clockwise): Through which function(s)?: ____	1 (strongly disagree) to 5 (strongly agree)
	(v) When I articulated a specific object (e.g., describe abstract shapes): Through which function(s)?: ____	1 (strongly disagree) to 5 (strongly agree)
	(vi) Did you use another type of instruction? __________ Through which function(s)?: ____	1 (strongly disagree) to 5 (strongly agree)
Q5_R	Reflection on remote collaboration: (i) How did augmented reality features of HoloMentor such as the shared workspace, the pointer or drawing impact your collaboration with your partner? Please describe it in a few sentences. (ii) Describe in detail one moment when you could not understand/give instructions to your partner. (Include details such as what was your intention, what happened, how do you wish things could have gone) (iii) Did you wish to have another function (apart from the ones provided) while collaborating with your partner? If yes, please explain. (iv) Describe in detail one moment when you could understand/convey instructions to your partner very easily. (Include details such as what happened, which functionality was used, your reaction and your partner's reaction)	

Table 2. Post-questionnaire for the remote helper

	Questions	Likert-Scales
Q1_L	I found the __________ function easy to understand: (Pointing, Drawing, Zooming, Rotation)	1 (strongly disagree) to 5 (strongly agree)
Q2_L	I found the __________ function useful: (Pointing, Drawing, Zooming, Rotation)	1 (strongly disagree) to 5 (strongly agree)
Q3_L	How did the remote helper use each function to support instructions? Please describe how the remote helper used each function to support their instructions, for example when asking to stack legos, move a lego, rotate a lego, stick together two blocks, etc. Pointing: __________ Drawing: __________ Zooming: __________ Rotation: __________	
Q4_L	I could easily understand:	
	(i) Where the remote helper was looking at: Through which function(s)?: ____	1 (strongly disagree) to 5 (strongly agree)
	(ii) Where the remote helper was pointing at: Through which function(s)?: ____	1 (strongly disagree) to 5 (strongly agree)
	(iii) When the remote helper talked about a spatial measure (e.g., place the lego five studs away): Through which function(s)?: ____	1 (strongly disagree) to 5 (strongly agree)
	(iv) When the remote helper talked about motion (e.g., rotate the lego clockwise): Through which function(s)?: ____	1 (strongly disagree) to 5 (strongly agree)
	(v) When the remote helper talked about a specific object (e.g., describe abstract shapes): Through which function(s)?: ____	1 (strongly disagree) to 5 (strongly agree)
	(vi) Did your remote helper use another type of instruction? __________ Through which function(s)?: ____	1 (strongly disagree) to 5 (strongly agree)

Table 3. Post-questionnaire for the local worker

C SEMI-STRUCTURED INTERVIEW QUESTIONS

Category	Questions
Collaboration	(i) What did you think of the system?
	(ii) Recall one instance when giving an instruction was particularly hard (breakdown)?
Territoriality	(i) Why did you put the shared workspace there? Were there any agreements about putting it in that position?
	(ii) [To the worker] How did you feel about the remote helper drawing/pointing on your view? (Did you feel that the remote helper is interfering with your workspace?)
	(iii) [To the helper] How did you feel about drawing/pointing on the worker's view? (Don't you feel like you're interfering with the worker's workspace?)
	(iv) Whose working area do you think the shared workspace is?
	(v) [To the worker] Did you ever ask the remote helper to erase their drawings?
	(vi) [To the helper] Did you ever ask the worker if you could draw/point on their view or ask them if you can erase something?
	(vii) Did you ever have a miscommunication that arose from referring to the base instead of the block or vice versa?
Awareness	(i) What did you think of real-time drawing directly on the workers view?
	(ii) [To the helper] How did you know the worker was aware of your instruction?
	(iii) [To the worker] when the remote helper used the zoom or rotation, did you understand the intention of the remote helper's action by seeing the change of the blue viewport?
	(iv) [To the worker] Did the viewport affect your collaboration or communication?
	(v) [To the helper] Did you realize that you use rotation instead of tilting your head?
	(vi) Do you think being able to recognize your partner's activities helps smooth communication?
	(vii) [To the helper] How was it to control the pointer of the workspace with a tablet?

Table 4. Post-questionnaire for the semi-structured interview

Received May 2022; revised August 2022; accepted September 2022