MAUI: Tele-Assistance for Maintenance of Cyber-Physical Systems

Philipp Fleck[1] , Fernando Reyes-Aviles[1] , Christian Pirchheim[1] ,
Clemens Arth[1,2] and Dieter Schmalstieg[1]
[1]*ICG, Graz University of Technology, Inffeldgasse 16/2, 8010 Graz, Austria*
[2]*AR4 GmbH, Strauchergasse 13, 8020 Graz, Austria*
{*philipp.fleck, fernando.reyes-aviles, pirchheim, dieter, arth*}*@icg.tugraz.at*

Keywords: Remote Collaboration, Telepresence, Augmented Reality

Abstract: In this paper, we present the maintenance assistance user interface (MAUI), a novel approach for providing tele-assistance to a worker charged with maintenance of a cyber-physical system. Such a system comprises both physical and digital interfaces, making it challenging for a worker to understand the required steps and to assess work progress. A remote expert can access the digital interfaces and provide the worker with timely information and advice in an augmented reality display. The remote expert has full control over the user interface of the worker in a manner comparable to remote desktop systems. The worker needs to perform all physical operations and retrieve physical information, such as reading physical labels or meters. Thus, worker and remote expert collaborate not only via shared audio, video or pointing, but also share control of the digital interface presented in the augmented reality space. We report results on two studies: The first study evaluates the benefits of our system against a condition with the same cyber-physical interface, but without tele-assistance. Results indicate significant benefits concerning speed, cognitive load and subjective comfort of the worker. The second study explores how interface designers use our system, leading to initial design guidelines for tele-presence interfaces like ours.

1 INTRODUCTION

Technical facilities increasingly incorporate cyber-physical systems, which must be operated and maintained using a mixture of physical and digital interfaces. This evolution is not limited to industrial settings: For example, many components of a modern car can no longer be repaired with physical tools without access to diagnostic devices that access the car's software-controlled sensors and actuators. In industrial environments, workers must learn to operate a mixture of conventional physical controls, such as levers or buttons, and digital interfaces, such as touch panels placed next to machines or desktop computer interfaces in a control station. For maintenance and repair, workers must access sensor telemetry data through diagnostic interfaces, while cross-referencing the telemetry with sources on paper. Mastering this multitude of information sources can be difficult. For cases where the local worker's competence is exceeded, equipment manufacturers operate call-centers or send experts to customers at high traveling costs.

Consequently, tele-assistance has been proposed as a relief for workers and experts. Most tele-assistance solutions establish a shared presence of remote expert and worker via some form of audio/video link combined with tele-pointing and on-screen annotation. One approach is to combine tele-presence with augmented reality (AR) by letting the worker wear a head-mounted display (HMD) with built-in camera and microphone. The worker has their hands free and does not have to switch attention to a stationary screen in order to access digital data. The remote expert can see what the worker sees, provide spoken instructions, and point out relevant areas in the shared video.

These collaborations primarily target supporting the mutual spatial understanding of worker and expert. They are sufficient in situations where only physical interfaces or facilities need to be manipulated. However, in a cyber-physical system, digital interfaces must be manipulated as well, requiring shared understanding of non-spatial aspects. Collaboration in digital space is common practice for office workers, who combine video conferencing with screen sharing and remote desktop interfaces. The expert will commonly operate the office worker's desktop computer remotely. In such a tele-assistance situation, the physical interface (*e.g.,* , feeding paper into a printer) must be operated by the worker, while the digital interface is subject to mixed initiative (both users take turns at operating the mouse). Oftentimes the digital interface will almost exclusively be operated by the expert, in particular, if the worker lacks technical understanding.

In this work, we combine AR tele-assistance with shared operations via a remote desktop-like user interface into a framework called *maintenance assistance user interface* (*MAUI*). Our system is able to display web-based interfaces as overlays in the HMD. These

Figure 1: Facility for production. (left) An expert in the control center dispatching (right) a worker on the shop floor, wearing a HoloLens, to fix machine downtime problems.

digital interfaces can contain any mixture of telemetry information, instructions and interactive widgets used to control the digital aspects of a cyber-physical system. Since the worker is busy with the physical tasks, may lack experience with the digital interface, or is simply overwhelmed by the amount of information displayed in the relatively small field of view of the HMD, the primary responsibility for configuring and operating the digital interface is deferred to the remote expert. The expert helps the worker to interact with the virtual and real world at the same time, thus reducing the risk of possible failures or mistakes. MAUI was designed with the intention of more efficient subdivision of work between expert and worker, providing the following contributions:

- We analyze the requirements of AR tele-assistance for industrial jobs, where remote operation and configuration of a user interface by a remote expert is required.
- We present implementation details of MAUI, which address the aforementioned requirements with its robust abilities for sharing audio, video, digital content and control state in harsh industrial environments.
- We discuss results of a user study demonstrating lower task completion times, reduced cognitive load, and better subjective comfort on performing a maintenance task.
- We discuss the results of an exploratory study, orthogonal to the first one, analyzing how web developers perform when creating user interface content in the MAUI framework.

2 RELATED WORK

AR can help a worker by purely displaying digital information. Henderson *et al.* (Henderson and Feiner, 2011) have shown the benefits of AR in the maintenance and repair domain. However, such pre-configured information sources are often unavailable. In this case, a good alternative is to link the worker to a remote expert providing live support. Dealing with a cyber-physical system adds the dimension of a digital interface, which can be controlled locally by the worker, or alternatively by the remote expert. Thus, our work is at the nexus of remote collaboration, interaction with cyber-physical systems and remote desktop user interfaces. We provide background to each of these topics in the remainder of this section.

2.1 Remote collaboration

Video transmission has been the enabling technology for tele-assistance, since the pioneering work of Kruger (Krueger et al., 1985). Early work in this space (Barakonyi et al., 2004; Wellner and Freemann, 1993) was mostly constrained to desktop computers, due to technical limitations. Recent progress in mobile and wearable computing has brought video conferences abilities to the factory floor.

However, establishing a shared spatial presence at the task location still proves challenging. Experts need to visually experience the worker's environment. The video stream from a camera worn by the worker will only show what the worker is seeing (Huang and Alem, 2013; Bauer et al., 1999; Kim et al., 2013; Chastine et al., 2008; Kurata et al., 2004). Giving the remote expert independent control of a robotic camera (Kuzuoka et al., 2000; Gurevich et al., 2012) is usually not economically feasible.

Apart from spoken instructions, most tele-assistance solutions let the expert provide visual-spatial references, either via hand gestures (Oda et al., 2013; Huang and Alem, 2013; Kirk and Stanton Fraser, 2006), remote pointing (Bauer et al., 1999; Fussell et al., 2004; Kim et al., 2013; Chastine et al., 2008), or hand-drawn annotation on the video (Fussell et al., 2004; Gurevich et al., 2012; Chen et al., 2013; Kim et al., 2013; Ou et al., 2003). Hand-drawing is either restricted to images from a stationary camera viewpoint (Kim et al., 2013; Gurevich et al., 2012; Huang and Alem, 2013; Bauer et al., 1999; Chen et al., 2013; Fussell et al., 2004; Kirk and Stanton Fraser, 2006; Kurata et al., 2004) or requires real-time 3D reconstruction and tracking (Chastine et al., 2008; Lee and Hollerer, 2006; Gauglitz et al.,

2014b; Gauglitz et al., 2014a).

2.2 Cyber-physical system Interaction

None of the remote collaboration systems mentioned in the last section takes into account the special requirements of a task that must be performed on a cyber-physical system. The dual nature of a cyber-physical system implies that each task will commonly consist of a physical task (*e.g.,* physical part mounted with screws) and a virtual task (*e.g.,* re-initializing a device after repair). It is crucial for the worker to receive support on both aspects of cyber-physical tasks.

Recent work in interaction design is starting to consider such interactions with cyber-physical systems. Rambach *et al.* (Rambach et al., 2017) propose that every cyber-physical system serves its own data (*e.g.,* sensor information, control interface) to enable new ways of interaction. Alce *et al.* (Alce et al., 2017) experimentally verify different methods of device interaction while increasing the number of devices. A common design are device-specific controls embedded in the AR user interface (Feiner et al., 1993; Ens et al., 2014).

Other recent work investigates AR in production and manufacturing industries. Kollatsch *et al.* (Kollatsch et al., 2017) show how to control an industrial press simulator within an AR application by leveraging its numerical interface, *i.e.,* a device interface allowing to run simulations which are partially executed on the machine. Han *et al.* (Han et al., 2017) concentrate on automated situation detection and task generation to give better and more responsive instructions, *e.g.,* for fixing paper jams in printers.

Cognitive conditions during task performance are a key element to success in many industrial situations. Maintenance workers must frequently perform multiple repair tasks during one shift, requiring a high level of flexibility and concentration. Therefore, recent research has considered how reducing factors like frustration, stress and mental load can improve overall performance. For instance, Baumeister *et al.* (Baumeister et al., 2017) investigate mental workload when using an AR HMD. Funk *et al.* (Funk et al., 2016; Funk et al., 2017) compare instructions delivered via HMD to tablet computers and plain paper. Recently, Tzimas *et al.* (Tzimas et al., 2018) reported findings on creating setup instructions in smart factories.

2.3 Remote desktop user interfaces

Remote desktop tools, such as Skype[1] or TeamViewer[2], combine video conferencing with remote operation. In theory, these tools have the features required for worker-expert collaboration and can be made to run on AR headsets such as the HoloLens. However, a closer inspection reveals that the similarities to the desired solution are shallow. Desktop user interfaces are operated using mouse and keyboard. They do not work very well when one user has reduced resources (*e.g.,* when using a phone with a small screen) or when network connectivity is unstable. Workers do not want to retrieve files and navigate them manually, while they are tending to a task. Moreover, shared spatial presence between worker and expert is not considered at all in desktop tools. Even re-using parts of desktop tool implementation in an AR applications turns out to be hard because of the differences between desktop and mobile operating systems.

Perhaps closest to our approach in this respect is the work of O'Neill *et al.* (O'Neill et al., 2011) and Roulland *et al.* (Roulland et al., 2011). Like us, they present a concept for remote assistance, focused on office printer maintenance. However, unlike ours, their work relies on schematic 3D rendering of a printer device, delivered on the printer's built-in screen, and very few details are provided on the implementation and extensibility of the system. In contrast, MAUI is a comprehensive tele-assistance framework. We describe details about its implementation, and evaluate the system's development and use.

3 DESIGN GOALS

In this section, we discuss insights from discussions with an industrial collaborator and give an overview of our system design, including the user interface components.

Our industrial collaborator operates manufacturing facilities for mass-produced goods using large-scale machines. Workers have to change parts, fix jams and adjust machinery. Most of the tasks involve cyber-physical systems; *e.g.,* in an adjustment task, the worker needs to adjust a physical valve and then restore a setting on a digital interface.

In the maintenance procedures we considered, we found significant motivation for using AR and tele-assistance. In most cases, live readings of machine telemetry are not accessible at the task location. A common situation is that a second person has to keep an eye on a physical display or analog gauges, while the worker is performing a repair. Providing real-time readings within a HMD drastically improves the workflow. Furthermore, step-by-step instructions increase the confidence of the worker and create a clear reference frame for the expert.

One recurring statement in the discussions was

[1]Skype: `https://www.skype.com`

[2]TeamViewer: `https://www.teamviewer.com`

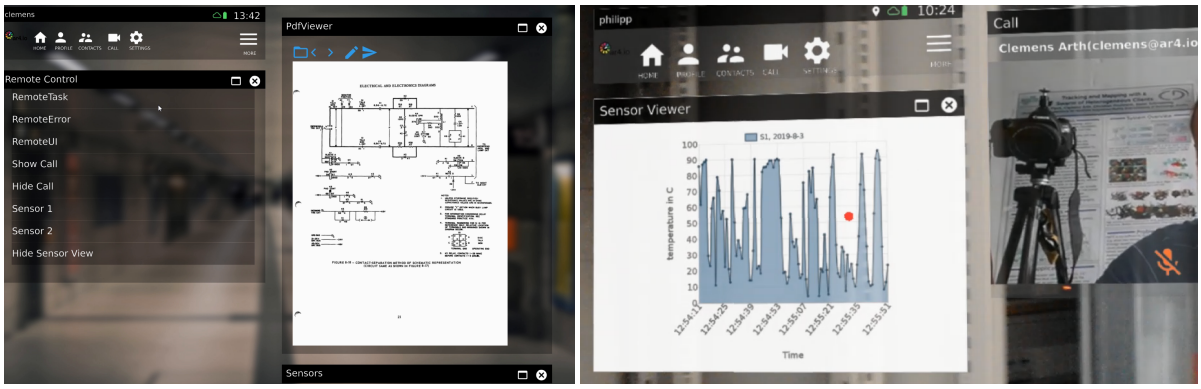

Figure 2: Example of a remote maintenance demo application. (left) Expert interface with an open document and shortcuts of remote commands, (right) worker UI within a call (A/V, data, commands) and live view of telemetry data shown as a graph.

that the pace of technology development is too high, and workers tend to be overwhelmed, when new features – in particular, digital ones – are introduced too quickly. The workers have a wide range in terms of age and come from a diverse educational background. Not surprisingly, young "digital natives" seem to have an easier time adjusting to cyber-physical interfaces than seasoned workers close to retirement. To cater to all these user groups in a flexible way, we adopted a strategy of introducing levels of expertise into the digital interfaces, supporting the progressive disclosure of new interface features as workers are learning.

Interaction techniques found in the standard user interface design of a device such as the HoloLens, *e.g.,* , a gaze-contingent cursor and a handheld Bluetooth "clicker" device, are not easily accepted by the workers. Our objective was to provide a very gentle learning curve, so we limited our initial design explorations to presenting 2D windows in the HMD in a style comparable to desktop interface, without any use of 3D geometry. Often, a detailed illustration is all it takes an experienced worker to solve a problem, so rather straight-forward features can provide good value and serve as motivating examples for workers asked to adopt the new technology.

We also heard from workers who already felt sufficiently comfortable with the AR system that registration of a CAD model with the real machine in 3D could help them in locating sensors or other components more quickly. However, we did not explore any use of 3D registration in the work presented in this paper.

In contrast to the workers, experts are generally well-trained specialists, such as construction engineers. They often have extensive knowledge in CAD models creation and a good understanding of cyber-physical systems. Therefore, the expert can be entrusted with more detailed control of the tele-assistance system. The tele-presence system should also give the expert the ability to judge the worker's abilities and decide on an appropriate course of action.

The tele-assistance system described in this paper was designed to provide remote expert support to workers directly on the production floor. Using a see-through HMD keeps the worker's hands free, so all visual elements must be embedded in the HMD view, and the need for explicit interaction must be reduced compared to a typical desktop interface in order to not distract the worker from the physical task.

Consequently, we formulated concrete requirements concerning telepresence, multimedia and remote control features, to address the requirements of potential applications. Worker features include:

- Initiating audio/video connections with easy connection management (phonebook, user discovery).
- A 3D mesh transfer function enables the worker to send a scanned 3D model of the current environment to the expert, including the worker's current position (Leveraging device capabilities *e.g.,* accessing the HoloLens mesh, but also allowing to plugin other 3D-reconstruction systems).

Expert features include:

- A screen capture module allows the expert to share screenshots of running applications (*e.g.,* CAD model viewer showing a cross section of a broken machinery part) with the worker.
- Multimedia content in a representation-agnostic form (PDF, HTML, images, links) can be transferred and displayed.
- Multi-page documents, such as PDF files, afford synchronized navigation between expert and worker. Content can be enriched with shared annotations.
- The expert must be able to control the worker's UI, including web content, but also triggering native interfaces of the cyber-physical system.

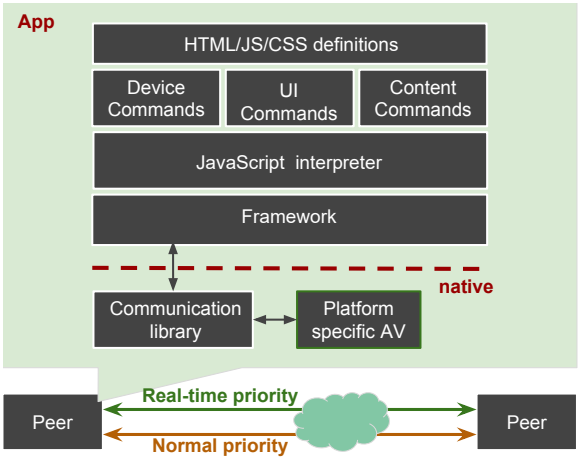

Figure 3: The application software stack of MAUI with several native and interpreted software layers.

The last feature, remote control, implies that a custom user interface must be dynamically embedded in the tele-assistance application, alleviating the worker from switching apps or configuring the user experience. The ability to change the user interface without touching the underlying application makes it also easy to embed tutorial functions into the user interface itself, which can be step-by-step enabled, facilitating a gentle learning curve for the worker. Once the worker is sufficiently familiar with the user interface, more features can be enabled, *e.g.,* transitioning from 2D to 3D visualizations or enabling additional controls for physical devices.

Finally, the industrial partners in this research project also required to use established software technologies. We chose a web-centric approach, inspired by Argon (MacIntyre et al., 2011), which leverages existing developer skills and allows for run-time extensibility.

4 IMPLEMENTATION

MAUI uses Unity3D and runs on the Microsoft HoloLens (for the worker) and Windows 10 (for the expert). A native audio/video component component enables real-time recording, playback, encoding and decoding of audio/video streams. Audio/video support can produce a high system load in particular on mobile platforms, so using hardware accelerated audio/video features on each target system was essential. Therefore, we bypass the regular communication structure of Unity3D in favor of calling device drivers on the target platform directly.

The communication component builds on a modified version of the RakNet[3] library for the lower levels of the network stack. We chose RakNet over alternatives provided in .NET, since it is relatively lightweight and enables unrestricted cross-platform support. To ensure high performance, these features use a native, multi-threaded implementation with static memory allocation. Furthermore, we can gradually control data-flow-rates and change the video and audio compression accordingly to adapt for any network-bandwidth.

We rely on the ability of Unity3D to load and register third-party libraries as components to support platform-specific features as well as dynamic user interfaces. Asynchronous coordination is enabled through event passing between components. Native libraries are developed in C++, but can be scripted using C# and Javascript.

The resulting framework has only minimal functionality to call native device functions, while all application code and user interface code is loaded dynamically. Such code is written in C#, Javascript and HTML and therefore interpreted at runtime. This allows arbitrary code changes at runtime, useful for continuous delivery and, in particular, for remote control.

In the remainder of this section, we describe details of the individual system components, as shown in Figure 3.

4.1 User interface component

MAUI uses a custom HTML engine with JavaScript interpreter[4] to create user interfaces in HTML5, CSS and JavaScript. User interfaces are rendered into texture maps and displayed on arbitrary polygonal surfaces inside the Unity3D view. Using web technologies lets a designer easily create responsive user interfaces and application layouts.

Even though, technically, the JavaScript engine is just a plug-in to the Unity3D game engine, JavaScript is promoted to the central facility for UI development. However, both worlds are not separated: An important property of the web-centric user interfaces is its native integration into the Unity3D scene. Both the document object model of the web content and regular game objects in Unity3D are exposed to Javascript. Thus, 2D web content and 3D game objects can interact. Apart from handling 2D and 3D events inside the Unity3D process, JavaScript can also handle external events. Such external events can be generated by web services (*e.g.,* REST) or cyber-physical systems.

4.2 Remote user interface orchestration

The interpreted nature of MAUI facilitates remote orchestration of the user interface experienced by the worker. All data transmitted between communicating nodes in MAUI is either an audio/video stream or

[3] https://github.com/facebookarchive/RakNet

[4] https://powerui.kulestar.com/

Figure 4: *The worker's view within the HoloLens*: (leftmost image) An error is reported, (right images) repair instructions are shown to the workers.

a message in a JSON envelope. The JavaScript interpreter unwraps the envelope and interprets the message. Since the message can contain JavaScript code, MAUI does not have to enforce a distinction of data and code. This implies that messages are directly executable. For instance, instead of just sending an image, we can pack the image into a command to create a new window on the recipient side and display the sent image. Therefore, the recipient system does not require an implementation of such functionality. If the shipped command is given a unique name, it can be stored as data and later invoked by that name to extend the set of user interface features. Direct execution of messages allows the expert to quite literally control the worker's user interface. The expert can invoke every function that that the worker could invoke, both inside Unity3D and outside (*i.e.,* addressing the cyber-physical system). Note that this remote control operates on the level of framework activities and not on the level of user interface events: The expert does not have to remotely "click a button" in the worker's user interface, but can invoke functions directly by name, or install new functions to extend the command set.

UI commands let the expert manipulate the user interface. This includes remotely opening, closing and re-arranging windows, or modifying the user interface widgets to assume an arbitrary state.

Device commands wrap native functions on the peer device, such as activating a flashlight or a sensor in a (non-extensible) JavaScript library. Configuration options of the peer device, such as location or WiFi SSID, can be queried dynamically to make the UI adaptive and context-sensitive. For example, the expert may query the device for the current location, and retrieve instructions for the facilities at the given location from a building information system.

Content commands are used to transfer multimedia data (PDF, HTML, images, links etc.) in a representation-agnostic binary form. Within an industrial scenario one would transmit part documentations, repair guides, images of broken parts or CAD drawings. Annotations and multi-page synchronization are also supported.

5 EXAMPLE SCENARIO

In this section, we describe a repair process in order to illustrate a possible workflow between worker and expert, as could happen in MAUI.

The worker faces an issue with a personal computer, which fails to boot. Therefore, the worker calls the expert by selecting the expert's id in the phonebook.

An audio-video call is established. The expert sees the worker's outfacing camera view, while the worker sees a live image of the expert. After identifying the malfunctioning PC, the expert is able to obtain a diagnostic message from the PC over the network: "the fan stalled". The expert raises the error to the worker by selecting the corresponding command in the remote commands menu. The remote command is shipped to the worker and displayed.

The worker is unfamiliar with this error and asks for help and further guidance. The expert replaces the error message with step-by-step-instructions, depicted in Figure 4.

While going through the steps the worker requires more detailed instructions. The expert decides to take a screenshot of the worker's view and annotate the important areas where the fan-mounts are placed. By pressing the *Send* button, the annotation is send and displayed in the workers view.

6 USER STUDY

In the following, we describe a user study designed to evaluate how well our system can support a worker in a task of repairing a cyber-physical system. The results of the study are given in subsection 6.4.

6.1 Conditions

We compared two conditions. In both conditions, the worker experienced the same user interface as part of a HoloLens application, but the availability of the expert varied: In the first condition, *self-navigation* (SN), no expert was available. In the second condition, *expert help* (EH) through the MAUI software was available.

Self navigation In the SN scenario, the worker controls and navigates the user interface alone. All decisions need to be made without external support, requiring to draw on one's own knowledge and experi-

Figure 5: (left) User study equipment: faulty device (electronics box with light-bulb, with device id label on top, HoloLens and Bluetooth keyboard), (center) Power cord with labeled outlets Labeled outlets, (right) The worker wearing the HoloLens has completed the task, and the light-bulb turns green.

ence. SN allows the worker to have the full control over the user interface with all its features, but the user does not have the ability to add new elements or exchange the UI with a different one. For instance, a new control window for an additional device cannot be added. Only the window layout can be altered before starting the procedure.

Expert help In the EH scenario, the worker can ask for expert help, and the expert can take care of the user interface. After establishing the connection, the expert begins supporting the worker via the features offered by MAUI. Meta-information about the worker's system (*e.g.,* OS-version, device-type, local environment, etc.) is conveyed to the expert, guiding the experts in providing support. During UI operation, the expert decides which content or windows are shown or hidden, so that the worker can better focus on the task at hand. This relieves the worker from having to go through potentially deeply nested and complex menu hierarchies to retrieve required information.

We were especially interested in the preparation time before the worker starts executing a particular task. This time encompasses the effort needed to get all necessary instructions and instruments ready to successfully perform the task. Based on preliminary testing, we hypothesized that EH would cause less mental workload, have a lower time to start (i.e., a reduced time between receiving instructions and the first step being carried out), and create higher comfort for the worker.

6.2 Task description

We designed our evaluation to investigate mental workload and effort, while performing a machine maintenance task. A module to control networked home appliances was added to our base system. A smart light-bulb (Philips Hue) was introduced as the target device (Figure 5). Timing data as well as other relevant statistical data inside the application was captured with a user-study component added to our framework. Finally, we created two distinct worker apps, the SN app and the EH app.

The task itself consisted of a combination of physical and virtual procedures. The physical procedures included unplugging and re-plugging a power cord, pressing power buttons and typing on a physical keyboard. The virtual procedures included interactions with the UI, reading of instruction reading and triggering functions on the light-bulb.

The task had the following structure divided in five steps within a step-by-step instruction list:

1. The faulty device lights up red.
2. In both conditions, the worker has to open the instructions within the UI (a list with nine step-by-step instructions) for the task.
3. The worker has to open the "light-bulb" widget to control the smart light-bulb. In SN, the worker must do this by searching through the menu. In EH, the list is opened by the remote expert.
4. While following the instruction, the worker has to perform a sequence of switching off power, unplugging, re-plugging, and switching power on again. Power outlets are labeled by id, and the instructions refer to particular outlet id for the re-plugging step.
5. The worker has to press the start button in the device-interface widget. The task finishes when the device lights up green.

6.3 Experimental procedure

We used a within-subjects design, where the conditions were SN and EH. To balance the tasks, we alternated the order of conditions between participants, and we alternated the plug position on the power-chord. Each participant had to perform a training task first, followed by either of the two methods. The training task consisted of a subset of the main UI, but widgets had different names and the training task involved only the smart light-bulb, and not the power cord. Between each task, a NASA TLX (Hart and Staveland, 1988) questionnaire was administered.

Participants completed a pre- and a post-study questionnaire to gather basic personal information, experience with AR, preferred method, and subjective difficulty. Before starting the user study, participant

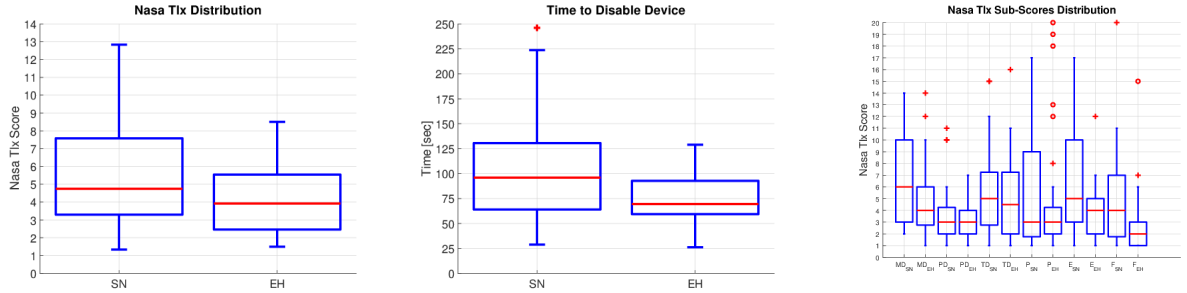

Figure 6: (left) Boxplot of NASA TLX scores SN vs EH, where **EH** reached lower scores with a p-value < 0.05 ($F_{1,62}, = 4.55$) showing a significance towards less mental workload. (center) Boxplot of Timings until the device got disabled. Timing SN vs EH, where **EH** reached shorter time with a p-value < 0.05 ($F_{1,44} = 5.84$) showing a significance towards EH needing less to disable the device. (right) Scores for NASA TLX categories. From left to right: (MD) mental demand; (PD) physical demand; (TD) temporal demand; (P) performance; (E) effort, (F) frustration. We found main effects on MD ($p < 0.05$, $F_{1,62}, = 4.61$), P ($p < 0.05$, $F_{1,62}, = 4.61$) and F ($p < 0.05$, $F_{1,62}, = 4.61$) preferring **EH** over **SN**, but no effects on categories related to the sitting position of the task (PD), the high completion rate (P), and short time it took to perform the task (TD).

were asked to put on the HoloLens and adjust it to comfort. They were explained that they were workers and had to complete a maintenance task. The study began with the app starting and the faulty device lighting up red.

The setup is depicted in Figure 5, showing the faulty device (light-bulb in box, with a non-working, unconnected and disabled power-supply on top) with the id on it, the labeled outlets and the HoloLens with the Bluetooth keyboard. Furthermore, a participant is shown successful completion the task with remote expert support, as indicated by the smart light-bulb lighting up in green.

The expert role was played by one of the authors, since we were primarily interested in the performance of the workers. Using a volunteer as expert would require training the volunteer to become an expert first. Interaction of volunteer expert and volunteer worker would then depend on the training success, which we wanted to rule out as a confounding factor.

The expert acted in a passive way, letting the participant decide how much help was needed. In all cases, the expert asked for the faulty device id to perform the search on behalf of the worker, and then opened the light-bulb widget for the worker. The expert also pointed out to carefully read through the instructions and to ask for help if needed.

6.4 Results

We tested 32 participants (4f), aged 21-40 (avg 28.9, median 28.5). The age distribution does not directly match the current age distributions of workers in such industries (Workforce Age Distribution, $\sim$10 years higher), but is a good representation for the next and further waves of workers. Half of the participants had never used a HoloLens before, and all of the participants had used step-by-step instructions before. All participants use a computer on a regular basis; 12.5% of the participants had never repaired, build or assembled a computer before. All participants felt that the training was enough to get familiar with the user interface.

A one-way repeated measures ANOVA found main effects on NASA TLX score supporting our hypothesis ($p < 0.05$, $F_{1,62}, = 4.55$) that the EH interaction method causes less mental workload while performing a task. Figure 6 shows the NASA TLX score distribution side-by-side with the time-to-start timings distribution. In both cases, EH has lower scores, which supports our hypothesis. By comparing the scores of each category with one-way repeated measures ANOVA, we found main effects on mental demand ($p < 0.05$, $F_{1,62}, = 4.61$), effort ($p < 0.05$, $F_{1,62}, = 4.61$) and frustration (($p < 0.05$, $F_{1,62}, = 4.61$)), favoring EH over SN. No significant

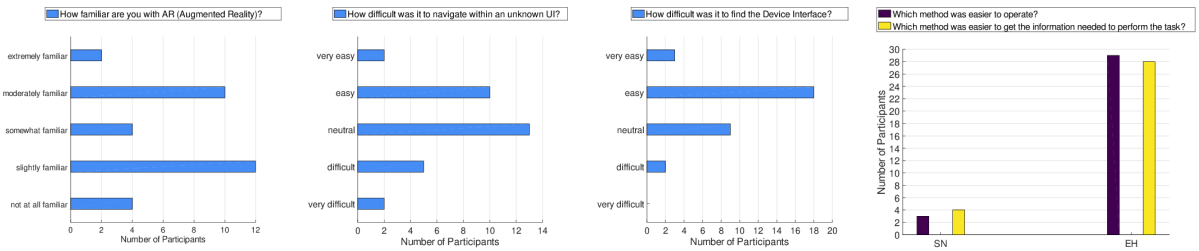

Figure 7: *left-to-right*; (i) Most of the participants were between moderately and slightly familiar with AR. (ii) The ease of navigation within the UI was mainly easy to neutral. (iii) It was also easy for the participants to find the *device-interface*. (iv) Around 90% of the participants preferred the **EH** interaction method over SN and also found it easier to operate.

effects were found on physical demand, temporal demand and performance. Reasons might be that physical demand is low in our scenario involving a seated position and lightweight objects. Since the overall task duration was short, temporal demand was low overall. Concerning performance, we observed high completion rates, which are later discussed in this section. Figure 6 shows the NASA TLX category scores and their pairwise distribution starting with mental demand (MD) in alternating order SN, EH.

We measured time-to-start as the time until the worker disabled the lightbulb. In case of SN, we started measuring the time when the worker performed the first action, until the worker pressed the disable device button. In case of EH, we started measuring when the participant sees the expert (to avoid hardware related delays from camera and microphone initialization).

Note how the tasks differed depending on the condition: In SN, the participant had to perform actions like reading through menu entries to open the right task, clicking, opening the lightbulb widget and entering the device id into the search box. In EH, the worker had to talk to the expert and read the device id aloud. Afterwards, the expert would raise the lightbulb widget in the worker's display.

Figure 6 shows overall shorter timings of EH. A one-way repeated measures ANOVA shows main effects on the time measurements, supporting our hypothesis ($p < 0.05$, $F_{1,62}, = 4.55$) that EH is faster with high significance. In real-life maintenance and repair scenarios, conditions might deteriorate while performing a task. Therefore, it can be very important to start acting swiftly.

In SN, 25% of the participants made an error, while, in EH, only 15% made an error. Overall, 8/32 missed the first step, either in SN or EH. This gives us 85% task completion rate for EH and 75% task completion rate for SN.

Questionnaire results After the experiment, we asked each participant *which method was easier to operate*. An overwhelming 90.6% preferred EH over SN. In response to the question *which method was easier to get the information needed to perform the task*, 87.5% preferred EH over SN. Figure 7 (d) summarizes these numbers. Some of the participants preferred SN over EH, which seems to be related to a possible low trust placed in the expert. We consider this a problem that must be addressed in the human resources structure of a company.

Additionally participants had to answer questions on a Likert scale concerning UI navigation. Most participants found it easy to navigate the UI and find specific widgets such as the one for the lightbulb. Interestingly, some participants who answered that it was easy to find the lightbulb widget actually had troubles finding it. One comment was *oh, it is written right there, i just have to read*, implying that the participant did not read the window title. These contradicting observations happened only in the SN condition and show that at least some of the participants did not spend full attention on the task. If a lack of concentration is a common cause of problems, the cost of expert support could be better justified.

The familiarity with AR across all participants is bi-modal, split up between *slightly familiar* and *moderately familiar*. Figure 7 shows the corresponding Likert questions. Few participants answered they were *somewhat familiar* with AR. We did not note any specific correlation of AR experience and performance.

User feedback The perceived behavior of the participants while performing the tasks was very homogeneous. Half of the participants had never used a HoloLens before. Users generally liked our UI design, but did not like the interactions prescribed by the Microsoft HoloLens SDK, which uses a head-locked crosshair and clicking via the "air-tap" gesture. A more sophisticated hand tracking (Xiao et al., 2018) may provide a good alternative.

Some users mentioned that buttons were too small and too closely spaced. We found that this criticism is strongly correlated with one's experience in using the HoloLens. Our informal observation is that small widgets are usable for seated users, but standing or moving users require larger widgets. Given the limited display real estate of the HoloLens, this implies the need for a radically uncluttered user interface. In MAUI, the expert can take care of this requirement.

7 DEVELOPER STUDY

In addition to the previous study concentrating on how the worker perceives the expert instructions, we were also interested to assess the characteristics of MAUI as a development platform. How easy or hard is it to create user interfaces in MAUI? Easy development would not only allow to quickly integrate new use cases into MAUI, but would also potentially enable an expert to modify the user interface on the fly.

Thus, we performed a qualitative user study were we asked two developers (*A* and *B*) with a web-development background (knowledge of HTML, CSS, JavaScript) to create exemplary applications and user-interfaces based on MAUI. The developers were instructed to apply workflows and leveraging the strengths of the provided framework. They were free to choose what features to implement and how to realize it. Questions were encouraged at any time, and

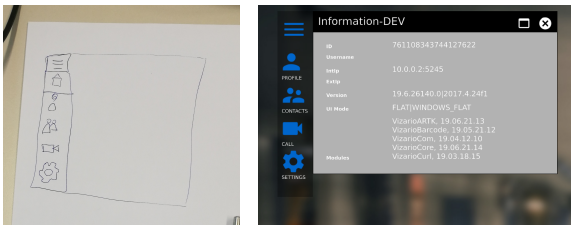

Figure 8: *New layout of Developer A.* (left) Draft on paper of the attempted layout. (right) Functional result after 2.5h.

thinking-aloud was encouraged. Each developer participated in a workshop-style introduction and immediately afterwards entered a design and implementation phase, while being accompanied and observed by one of the authors.

7.1 Procedure

Participants took part in a one-hour face-to-face workshop provided as an introduction to the MAUI framework. After the introduction, they filled in an introductory questionnaire and started the design and implementation phase (up to three hours). One author accompanied the developer and answered API questions, since we do not have extensive written documentation at this point. To conclude, a second questionnaire was administered, followed by an semi-structured interview. Finally a NASA-TLX assessment was done.

7.2 Developers and Results

In this section we give details about our two participants, *A* and *B*, and discuss their experiences. *A* is a 36 years old male researcher at Graz University of Technology. He writes code daily, but uses HTML/CSS/JS only occasionally. He is slightly familiar with Electron [5], AngularJS [6] and other JS frameworks.

The basic idea of *A* was to improve the layout of a UI example presented to him, giving the example a style more akin to contemporary mobile apps. The re-worked application, depicted in Figure 8, shows the new layout, which offers more screen-space compared to the original one. Since *A* has a strong C++ background, he was less skilled with the design procedure, but highly productive with the provided framework. Two bugs were found and immediately fixed. Most of the time was spent on getting the layout to work appropriate. As he said, *the provided functions are very useful* and helped with the implementation.

A felt neutral about how easy it was to create the experience and that the presented way was slightly different to the traditional ways of implementing web content. Furthermore, he felt that it was easy to use

[5]Electron: `https://electronjs.org/`

[6]AngularJs: `https://angularjs.org`

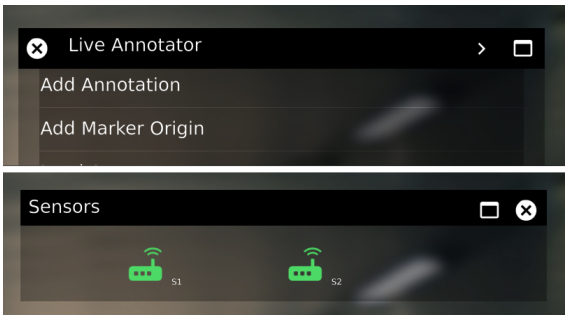

Figure 9: *Sample App Developer B.* The improved version with the *close* button on the left and the *toggle* and *maximize* controls on the right in the *top view*. The original version with the controls grouped together in the *bottom view*.

the given material. This developer also mentioned that, with more time and more thoughts on the design, better results where easily possible. He was happy with the capabilities and how easily it was to integrate web content with the existing Unity 3D application.

B is a 28 years old male researcher at Graz University of Technology. He writes code daily and often uses HTML/CSS/JS. Furthermore, he is moderately familiar with Electron and somewhat familiar with AngularJS and similar frameworks. This developer has a lot of experience with jQuery [7].

Right after going through the provided examples, *B* immediately pointed out usabilty issues. He made small, but impactful layout changes. Having the window-control buttons *toggle*, *maximize* and *close* next to each other encourages mistakes to happen. Pressing button in close proximity is easy for desktop users with a mouse, whereas in AR (on the HoloLens) selection is more brittle, especially if a mistake closes the window (see Figure 9). Code changes were done to the Javascript and JSON templates and therefore integrated seamlessly with the application.

Another usability problem was identified within the PC repair instructions sample. Having the controls below the content always leads to an extended head movement in AR. Placing the controls next to the content allows for easier navigation without distraction (see Figure 10).

It was easy for *B* to modify and improve selected samples. The developer stated that MAUI did not diff much from conventional web development. Overall, both developers created presentable and working results, one tackling the display real estate and one fixing usability problems relevant for AR presentation. Within the short period of time, the developers were able to utilize the framework and performed meaningful changes to selected samples. *B* mentioned *to not forget to check the implementation*

[7]jQuery`https://jquery.com/`

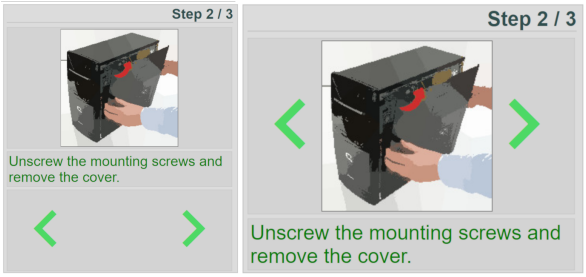

Figure 10: *Changes made by developer B to the PC repair task.* The green controls of the original version (left) moved next to the center image (right) increasing usability, especially in AR.

within Unity3D-Application, to avoid incompatibility with unsupported features *e.g., css display: flex.*

Nasa TLX shows nothing unexpected. The mental demand was higher than normal for *A*, because of his limited use of web technologies. Both developers reported normal to good performance in self-assessment, possibly being more critical than an average evaluator would. Frustration and effort where near "high" for *A*, whereas, for *B*, it was near "low".

7.3 Discussion

The developer study shows that developers with basic web knowledge can perform meaningful changes to existing applications and, given enough time, could rather easily create whole applications themselves. Within the industrial context, this aspect turns out to be of key value. Today, an AR developer is expected to have special knowledge, such as Unity 3D or C# to write an application for a commercial AR device, such as the HoloLens. Lowering the entry gap by removing the requirement to have profound knowledge in 3D and C++ out allows to tap into a larger pool of possible developers, as also observed by MacIntyre *et al*. (MacIntyre et al., 2011).

In a near future, where AR is commonly used, web content can be generated easily, or it can be generated automatially by web service portals to cyber-physical systems and the internet of things.

8 CONCLUSION

We have presented MAUI, a collaborative platform that lets a worker wearing an AR headset call a remote expert to help with operating a cyber-physical system. MAUI combines spatial AR tele-presence through shared audio/video with shared control of a web-based user interface displayed in the AR headset. The expert can take full control over the UI, relieving the worker of handling the digital interface and letting the worker concentrate on the physical interface instead.

We performed a quantitative user study in order to compare both interaction methods in terms of time and comfort benefits. The results show that expert help was overwhelmingly preferred by participants over working alone. Our user study results show that support from an expert can reduce cognitive load and increase performance. In particular, the time until the first physical action can be performed is decreased, allowing a quick response in critical real-world scenarios (Huang et al., 2013; Huang and Alem, 2011).

Furthermore, two developers touching the presented work for the first time were able to come up with meaningful results in a short time-frame of up to three hours. One improved the layout of the application towards more screen-estate and one improved the usability especially in AR (HMD).

A future study could also compare pure audio/video help with remote UI help, but this study would involve more complicated aspects. A study comparing different feature sets of remote help for AR would also include whether the remote expert can observe all relevant physical activities around the expert, if the expert can control IoT objects in the worker's environment, and so on.

In the future, we plan to extend our tests to real workers in a production environment and improve the aesthetics of the user interface to fit modern design standards[8]. Long-term evaluations will show the effectiveness on educating workers on problem solving. We plan to improve the widget placement system over the standard solution[9] to reliably avoid situations where the UI blocks the worker's view of the physical objects. Moreover, we plan a user interface management system for delivering tailored user interfaces to workers based on a formal task description.

ACKNOWLEDGEMENTS

The authors wish to thank Denis Kalkofen. This work was supported by FFG grant 859208.

REFERENCES

Alce, G., Roszko, M., Edlund, H., Olsson, S., Svedberg, J., and Wallergård, M. (2017). [poster] ar as a user interface for the internet of things - comparing three interaction models. In *ISMAR-adj.*, pages 81–86.

Barakonyi, I., Fahmy, T., and Schmalstieg, D. (2004). Remote collaboration using augmented reality videoconferencing. In *Proc. of Graphics Interface (GI)*, pages 89–96. Canadian Human-Computer Comm. Society.

Bauer, M., Kortuem, G., and Segall, Z. (1999). "where are you pointing at?" a study of remote collab. in a wearable videoconf. system. In *ISWC*, pages 151–158.

Baumeister, J., Ssin, S. Y., ElSayed, N. A. M., Dorrian, J., Webb, D. P., Walsh, J. A., Simon, T. M., Irlitti, A.,

[8]Design Principles

[9]Microsoft Mixed Reality Toolkit: Tag-along

Smith, R. T., Kohler, M., and Thomas, B. H. (2017). Cognitive cost of using augmented reality displays. *TVCG*, 23(11):2378–2388.

Chastine, J. W., Nagel, K., Zhu, Y., and Hudachek-Buswell, M. (2008). Studies on the effectiveness of virtual pointers in collaborative augmented reality. *3DUI*, pages 117–124.

Chen, S., Chen, M., Kunz, A., Yantaç, A. E., Bergmark, M., Sundin, A., and Fjeld, M. (2013). Semarbeta: mobile sketch-gesture-video remote support for car drivers. In *Augmented Human International Conference (AH)*.

Ens, B., Hincapié-Ramos, J. D., and Irani, P. (2014). Ethereal planes: A design framework for 2d information spaces in 3d mixed reality environm. In *SUI*. ACM.

Feiner, S., MacIntyre, B., Haupt, M., and Solomon, E. (1993). Windows on the world: 2d windows for 3d augmented reality. In *UIST*, pages 145–155. ACM.

Funk, M., Bächler, A., Bächler, L., Kosch, T., Heidenreich, T., and Schmidt, A. (2017). Working with ar?: A long-term analysis of in-situ instructions at the assembly workplace. In *PETRA*, pages 222–229. ACM.

Funk, M., Kosch, T., and Schmidt, A. (2016). Interactive worker assistance: Comparing the effects of in-situ projection, head-mounted displays, tablet, and paper instructions. In *UBICOMP*, pages 934–939. ACM.

Fussell, S. R., Setlock, L. D., Yang, J., Ou, J., Mauer, E., and Kramer, A. D. I. (2004). Gestures over video streams to support remote collaboration on physical tasks. *Hum.-Comput. Interact.*, 19(3):273–309.

Gauglitz, S., Nuernberger, B., Turk, M., and Höllerer, T. (2014a). In touch with the remote world: Remote collaboration with augmented reality drawings and virtual navigation. In *VRST*, pages 197–205. ACM.

Gauglitz, S., Nuernberger, B., Turk, M., and Höllerer, T. (2014b). World-stabilized annotations and virtual scene navigation for remote collaboration. In *UIST*, pages 449–459, New York, NY, USA. ACM.

Gurevich, P., Lanir, J., Cohen, B., and Stone, R. (2012). Teleadvisor: A versatile augmented reality tool for remote assistance. In *CHI*, pages 619–622. ACM.

Han, F., Liu, J., Hoff, W., and Zhang, H. (2017). [poster] planning-based workflow modeling for ar-enabled automated task guidance. In *ISMAR-adj.*, pages 58–62.

Hart, S. G. and Staveland, L. E. (1988). Development of nasa-tlx (task load index): Results of empirical and theoretical research. *Human mental workload*, 1(3):139–183.

Henderson, S. and Feiner, S. (2011). Exploring the benefits of augmented reality documentation for maintenance and repair. *TVCG*, 17(10):1355–1368.

Huang, W. and Alem, L. (2011). Supporting hand gestures in mobile remote collaboration: A usability evaluation. In *BCS Conference on Human-Computer Interaction (BCS-HCI)*, pages 211–216.

Huang, W. and Alem, L. (2013). Handsinair: A wearable system for remote collaboration on physical tasks. In *CSCW*, pages 153–156. ACM.

Huang, W., Alem, L., and Tecchia, F. (2013). Handsin3d: Augmenting the shared 3d visual space with unmediated hand gestures. In *SIGGRAPH Asia 2013 Emerging Technologies*, SA '13, pages 10:1–10:3. ACM.

Kim, S., Lee, G. A., and Sakata, N. (2013). Comparing pointing and drawing for remote collaboration. In *ISMAR*, pages 1–6.

Kirk, D. and Stanton Fraser, D. (2006). Comparing remote gesture technologies for supporting collaborative physical tasks. In *CHI*, pages 1191–1200. ACM.

Kollatsch, C., Schumann, M., Klimant, P., and Lorenz, M. (2017). [poster] industrial augmented reality: Transferring a numerical control connected augmented realty system from marketing to maintenance. In *ISMAR-adj.*, pages 39–41.

Krueger, M. W., Gionfriddo, T., and Hinrichsen, K. (1985). Videoplace - an artificial reality. In *CHI*, pages 35–40, New York, NY, USA. ACM.

Kurata, T., Sakata, N., Kourogi, M., Kuzuoka, H., and Billinghurst, M. (2004). Remote collaboration using a shoulder-worn active camera/laser. In *ISWC*, volume 1, pages 62–69.

Kuzuoka, H., Oyama, S., Yamazaki, K., Suzuki, K., and Mitsuishi, M. (2000). Gestureman: A mobile robot that embodies a remote instructor's actions. In *CSCW*, pages 155–162, New York, NY, USA. ACM.

Lee, T. and Hollerer, T. (2006). Viewpoint stabilization for live collaborative video augmentations. In *ISMAR*, pages 241–242. IEEE Computer Society.

MacIntyre, B., Hill, A., Rouzati, H., Gandy, M., and Davidson, B. (2011). The argon ar web browser and standards-based ar application environment. In *ISMAR*, pages 65–74.

Oda, O., Sukan, M., Feiner, S., and Tversky, B. (2013). Poster: 3d referencing for remote task assistance in augmented reality. In *3DUI*, pages 179–180.

O'Neill, J., Castellani, S., Roulland, F., Hairon, N., Juliano, C., and Dai, L. (2011). From ethnographic study to mixed reality: A remote collaborative troubleshooting system. In *CSCW*, pages 225–234. ACM.

Ou, J., Fussell, S. R., Chen, X., Setlock, L. D., and Yang, J. (2003). Gestural communication over video stream: Supporting multimodal interaction for remote collaborative physical tasks. In *ICMI*, pages 242–249, New York, NY, USA. ACM.

Rambach, J., Pagani, A., and Stricker, D. (2017). [poster] augmented things: Enhancing ar applications leveraging the internet of things and universal 3d object tracking. In *ISMAR-adj.*, pages 103–108.

Roulland, F., Castellani, S., Valobra, P., Ciriza, V., O'Neill, J., and Deng, Y. (2011). Mixed reality for supporting office devices troubleshooting. In *VR*, pages 175–178.

Tzimas, E., Vosniakos, G.-C., and Matsas, E. (2018). Machine tool setup instructions in the smart factory using augmented reality: a system construction perspective. *IJIDeM*.

Wellner, P. and Freemann, S. (1993). The Double DigitalDesk: Shared editing of paper documents. Technical Report EPC-93-108, Xerox Research.

Xiao, R., Schwarz, J., Throm, N., Wilson, A. D., and Benko, H. (2018). Mrtouch: Adding touch input to head-mounted mixed reality. *TVCG*, 24(4):1653–1660.