Computer Supported Cooperative Work (CSCW)
DOI 10.1007/s10606-015-9232-7

Design and Implementation of TeleAdvisor: a Projection-Based Augmented Reality System for Remote Collaboration

Pavel Gurevich[1], Joel Lanir[2] & Benjamin Cohen[1]
[1]*IBM Haifa Research Laboratory, Mt. Carmel, Haifa, 31905, Israel;* [2]*University of Haifa, Mt. Carmel, Haifa, 31905, Israel (E-mail: joel.lanir@gmail.com)*

Abstract. TeleAdvisor is a versatile projection-based augmented reality system designed for remote collaboration. It allows a remote expert to naturally guide a local user in need of assistance in carrying out physical tasks around real-world objects. The system consists of a small projector and two cameras mounted on top of a tele-operated robotic arm at the worker's side, and an interface to view the camera stream, control the point-of-view and gesture using projected annotations at the remote expert's side. TeleAdvisor provides a hands-free, mobile, low-cost solution that supports gesturing by the remote expert while minimizing the cognitive overhead of the local worker. We describe the challenges, design considerations and implementation details of the two phases of the TeleAdvisor prototype, as well as its evaluation and deployment at an industrial manufacturing center. We summarize our understandings from our experiences during the project and discuss the general implications for design of augmented reality remote collaboration systems.

Keywords: Remote collaboration, Remote assistance, Augmented reality, Mobile projector, Mobility

1. Introduction

In many situations, a person needs help in figuring out how to perform a task that involves working with a physical object. However, there is not always someone with the right expertise available nearby. In a remote physical collaboration task, a remote expert (the *helper*) provides instructions to a novice user (the *worker*) on how to complete a task around a physical object. The worker lacks some knowledge pertaining to the object or the operation of the task, and the helper aids the worker in diagnosing the problem and providing instructions on how to perform unfamiliar operations. Examples of such a scenario include a remote expert technician guiding an emergency repair of a machine in a production line, a help desk operator remotely guiding the fixing of a printer, or a junior physician receiving remote instructions from a senior one during an operation. Unlike video conferencing and most online remote collaboration tools, a remote physical collaboration scenario is inherently non-symmetrical. The helper has most of the knowledge on how to complete the task, while the worker has the physical hands and tools as well as a better overall view of the environment.

Published online: 10 September 2015

In the simplest way, a remote helper can assist the worker through a voice call and verbally explain what to do. However, many studies have shown that a shared view of the worker's environment improves the performance of a collaborative task (Fussell et al. 2003; Kraut et al. 2003). A shared visual context of the working environment improves coordination, facilitates common ground, and provides a shared understanding of what is being discussed (Gergle et al. 2013; Ranjan et al. 2007). This can be done by simply showing the helper a video feed of the worker's workspace. Other works have suggested to further improve communication between helper and worker by communicating helpers' gestures onto the worker's environment (Bauer et al. 1999; Fussell et al. 2004). Gesturing can augment the vocabulary of the users by providing a common ground for deictic references. This enables replacing complex referential descriptions with simple pointing, and thus improves performance and communication (Fussell et al. 2004; Kirk et al. 2007).

Several mechanisms have been suggested to enable remote gesturing of the helper to assist or guide a worker through various tasks. These mechanisms include the use of a remote pointer (Kuzuoka et al. 2000), sketches (Fussell et al. 2004), or full images of the workers hands (Kirk and Stanton Fraser 2006; Tang et al. 2006). An important question is how to convey these gestures to the worker. One possibility is to draw the gestures over a video feed of the worker's environment and present this video feed to the worker on a screen placed in the workspace (Fussell et al. 2004; Kuzuoka et al. 2000; Ou et al. 2003). However, placing a video screen in the worker's environment to present the helper's gestures may create a fractured ecology in which the worker needs to split attention between the task and the external display, increasing the cognitive load of the worker and hindering the performance (O'Neill et al. 2011). Furthermore, it may be difficult for the worker to synchronize the information between the video and the real-world views.

A promising approach for enabling gestures in remote collaboration involves using augmented reality (AR). Augmented reality systems enhance the user's perception of the real world through virtual objects that appear to coexist at the same space as the real world (Azuma et al. 2001). Thus, it may be possible to augment the workspace with the remote helper's gestures using AR technologies. One possible direction is to use a camera and a head mounted display (HMD) worn by the worker, sending the camera's video stream to the helper while showing the helper's gestures on the HMD. However, a head mounted camera would provide a very unstable view for the helper, requiring the worker to limit his or her head movements. Furthermore, the view for the helper changes each time the worker moves his or her head which can be highly confusing. In addition, it is still technically challenging to create annotations that would "stick" to objects, and thus current solutions loopback the video of the workspace to the worker, augmented with the helper's gestures (Alem et al. 2011; Huang and Alem 2011). We suggest using AR in a different way. Instead of using an HMD, it is possible to project the annotation directly on top of the objects in the workspace and thus directly augment the workspace with exact annotations. This has the advantage of leaving the worker's hands free to work and not burdening the worker with wearing complex equipment, while leaving the view of the helper stable.

We describe the design considerations and implementation of TeleAdvisor - a novel augmented reality remote collaboration system. TeleAdvisor provides a transportable, hands-free projected AR solution for remote collaboration, composed of a pico-projector and camera mounted on top of a robotic arm on the worker side and an interface to view and control the camera on the helper side (see Figure 1). It can be moved anywhere in the worker's environment leaving the worker's hands free to complete the tasks, while allowing the helper to control the point of view and have a wide view of the worker's environment. It supports a wide range of gesturing options, enabling enhanced communication option between worker and helper. For the implementation of TeleAdvisor, we used of-the-shelf hardware and a combination of open-source and developed software in order to enable easy reproduction and to keep costs low.

1.1. Overview

This paper is an extended version of an earlier report on the first phase of TeleAdvisor (Gurevich et al. 2012). We discuss the evolution of the TeleAdvisor's design and prototype from its first phase to the second, current phase, provide details on the challenges and the design considerations we faced, and provide implementation details of our solution. We also discuss the implications and understandings that stemmed from our work in regards to the general design of remote collaboration systems. We begin with Section 2 that provides the background for the research on remote collaboration. In Section 3, we provide an overview of the system architecture and describe the two phases that we followed in our implementation. Section 4 describes the challenges, design consideration and the implementation details of the

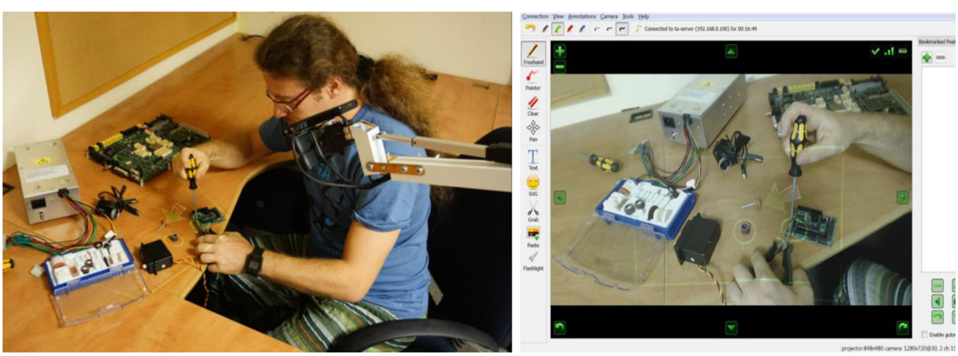


Figure 1. An example of the usage of TeleAdvisor. On the left side, the worker sees the annotations made by the helper, while carrying on the verbal conversation. His hands are free to perform the work. On the right side, the helper sees the physical space and user's hands via the interface, and can communicate with worker using various annotations while talking to the worker.

system, while Section 5 describes the user evaluations we performed. Finally, we discuss the implications of our work and possible future works in Section 6.

2. Background

We first describe the importance of a shared visual space and remote gesturing capabilities in remote collaboration. We then provide details on how various augmented reality solutions were used to enable remote gesturing. Finally, we discuss the importance of supporting workspace flexibility of movement for the worker.

2.1. The importance of gestures

Various studies have shown that a shared visual space of the working environment, including the task objects and supporting tools, enhances communication between helper and worker and improves performance (Kraut et al. 2003). A shared visual context is also crucial for the grounding of communication. Grounding of communication refers to the interactive process in which communicators exchange evidence concerning their mutual understanding. This entails identifying what the partner is attending to at any given time, establishing a joint focus of attention, monitoring the partner's level of comprehension, and establishing conversational efficiency when referring to task objects (Kraut et al. 2003). Fussell et al., looking at these dimensions, emphasized that when interacting with physical objects it is not enough to focus on the work area, or on head movements or facial expressions, but rather it is more important to focus on the interaction with the objects using pointing and gesturing (Fussell et al. 2004). Indeed, remote gesturing was shown to support the grounding of deictic and instructional references, enabling the replacements of complex referential descriptions with simple pointing (Kirk et al. 2007), minimizing the collaborative effort expended in the conversation. Supporting remote helper's gestures, has shown to increase effectiveness, reduce errors and improve the overall communication between the worker and the helper over a solution that includes only a shared view (Fussell et al. 2004).

Various methods have been suggested to represent the remote gestures at the workspace. The simplest way to point at objects is using a remote pointer (Kurata et al. 2004; Kuzuoka et al. 2000). However, a remote pointer's vocabulary is limited to showing a location. Free-hand sketches drawn by the helper) can substantially increase the gesturing language. Finally, full images of the workers hands (Huang et al. 2013; Kirk and Stanton Fraser 2006; Tang et al. 2006) is another method to enable gesturing that might be more natural for the helper and intuitive for the communication. Another important question is how to convey the gestures to the worker. One possibility is to render the gestures over a video feed of the worker's environment and present this video feed to the worker on a screen placed in the environment (Fussell et al. 2004; Kuzuoka et al. 2000; Ou et al. 2003) However, placing a video screen in the worker's environment to present the helper's gestures may require the worker to split attention between the task and the external display requiring extra cognitive effort (O'Neill et al. 2011). Thus, in order to combine between the virtually-represented gestures and the real-world objects the use of augmented reality technologies shows much promise.

2.2. Using augmented reality in remote collaboration

With the advent of display technologies, augmented reality (AR) has gained much media attention lately. The promise of being able to view the real world while augmenting it with digital information has ignited the imagination of many. Nevertheless, AR has been an active field of research since the early 1990's (Azuma 1997). AR has been used in many domains, and has been looked at for gaming, manufacturing, military purposes and more. In particular, it can be useful for training and demonstrating how to perform a variety of tasks such as playing the guitar, repair of a printer, teaching painting or manufacturing (Baird and Barfield 1999; Flagg and Rehg 2006; Motokawa and Saito 2007; Neumann and Majoros 1998). For example, Henderson and Feiner explored how to use AR technologies in automatic instructions of maintenance and repair tasks (Henderson and Feiner 2011; Henderson and Feiner 2009). They augmented the physical view of the user with various types of 3D graphical information that would guide and assist the user during the maintenance task, demonstrating their system in repair tasks of a complex military armored personnel carrier and a Rolls-Royce engine. A remote collaboration scenario is very similar in nature. Here too, a user needs assistance and guidance in term of explanation and demonstration of how to complete a physical task (often also in the field of maintenance and repair). However, rather than having automatic guidance, the guidance is provided by a remote human expert that can better react and provide immediate feedback to the users' actions. Thus, the promise of using augmented reality tools in remote collaboration tasks lies with the potential to provide remote spatial cues that blend together with the actual environment (Billinghurst and Kato 2002).

AR displays for viewing the merged virtual and real environments can be classified into three major categories: see through handheld displays, head-worn displays and projection displays (Azuma et al. 2001). In handheld displays, the camera's view of a smartphone or Tablet computer is augmented with digital information and computer-generated graphics. Gauglitz et al. (2012, 2014) used this approach to enable digital annotations in a remote collaboration scenario. The worker holds the Tablet device in one hand, seeing the helper's annotations through the Tablet device. A different approach suggests using Head Worn Displays (HWD) worn by the worker. Kraut et al., in a set of studies, examined the use of a head-mounted conferencing system on communication and performance in a bicycle repair task (Kraut et al. 2003). While their system did not support helper's gestures, and therefore did not actually blend real and virtual objects (and thus should not be considered as an AR system), it did examine the effect of equipping a worker with a HWD that showed the helper's face as well as an electronic manual. The results from a set of user studies did not find that adding a video channel to the worker (thorough the HWD) improved performance over an audio-only channel. However, they found that the head-mounted video channel influenced the way people talked about the task. They concluded that while workers used the video technology, helpers had difficulties to perceive the workspace due to small view of the worker's hands, camera slippage and the limited view of the surrounding area. They also commented that the lack of performance improvement might be due to the fact that no gesturing options were offered in these experiments. Aiming at enabling gesturing and referencing of objects in a HWD solution, Bauer et al. (Bauer et al. 1999) used a pointer controlled by the helper that was added to the video feed of the workspace that was seen both by the helper and the worker. They showed that a simple pointer tool can enhance communication by effectively guiding and directing the worker's activities. HandsOnVideo (Alem et al. 2011; Huang and Alem 2011), further enhanced the gesturing capabilities. In their work, an HWD AR system for remote collaboration was built with a mining site as a motivating use case. The video stream taken from a camera mounted on top of the HWD was shown to the helper on a large tabletop display. A video of the helper's hand gestures on top of this video was sent back to the worker and shown on a near-eye display. Thus, the worker was able to see the helper use his or her hands to gesture on top of the video of his environment. While surely being beneficial, using a HWD does have some inherent problems. A head mounted camera's view provides an unstable view for the helper, requiring the worker to consciously limit head movements. Furthermore, the camera's focus changes every time the worker moves his or her head. This can create many rapid or irrelevant changes of the point of view (POV), which can be highly distracting for the helper (Fussell et al. 2003). In addition, the helper is constrained to look at a specific location – the area the worker is currently looking at, and cannot work or examine the area in parallel to the worker, a practice that has been shown to be useful (Lanir et al. 2013).

In the third AR approach, gestures are projected directly on top of the worker's environment. Early works used physical pointers and laser pointers for gesturing at the worker's environment (Bauer et al. 1999; Kurata et al. 2004; Kuzuoka et al. 2000). Still, laser pointers are transient and limited to showing referential gestures. A projector may provide a larger gesturing vocabulary that can further enhance communication by enabling persistent representations and representational gestures (Fussell, et al. 2004). Many works examined the use of a camera in combination with projector technologies, mainly for interacting with augmented information (Ishii et al. 2002; Junuzovic et al. 2012; Wellner 1993). In remote collaboration it is possible to use projected AR to project the helper's annotations directly on top of the user's environment. Kirk et al. (Kirk et al. 2007; Kirk and Stanton Fraser 2006) designed a system that projected full gestures onto a desk which constituted the worker's workspace using a top-mounted projector. A camera was placed on top of the worker's workspace and the video stream was sent to the helper and shown to the helper on a vertical display. The helper's view of hands, as well as annotations made by the helper, were videotaped and sent to be projected in the worker's workspace. We take a similar approach of using a combined camera-projector setting to view the environment and project annotations on top of it. However, our approach is more flexible using a transportable device that can be used in an ad-hoc matter any place at the worker's environment.

2.3. Supporting workspace flexibility

Top-mounted or wall-mounted projectors, as were used in (Kirk et al. 2007; Kirk and Stanton Fraser 2006), may have a large projection area and can support high-resolution projection, however, they are stationary and are limited to a prearranged work area, on which the projector was mounted prior to the task. In many scenarios the workspace cannot be set ahead of time and more flexibility is needed. Furthermore, a fixed-view scene camera may not be able to cover the entire workspace in sufficient detail (Fussell et al. 2003). Thus, some works have looked at how to enable flexible control of the direction and movement of the camera and gesturing device. Ranjan et al., suggested to automatically track and follow the user's hand movements as an indicator of where and what the user is currently working on (Birnholtz et al. 2010; Ranjan et al. 2007). Similarly, In Lightguide (Sodhi et al. 2012), user's hands were tracked and automatic instructions were projected on top of the hands, mainly for the purpose of movement guidance such as in exercise or physical therapy. However, automatically following the worker's point of view or hand movements may not always be the right strategy. Often, the worker may wish to look elsewhere, and most workers and helpers prefer to leave the control of the point-of-view in the hands of the helper (Lanir et al. 2013).

The WACL system (Kurata et al. 2004) aimed at combining helpers' control of the point of view with the workspace flexibility and mobility afforded by a HWD device. A steerable camera and laser device was mounted on top of the workers head, yet was controlled by the helper. The helper could follow the worker's point of view, but could also look independently into the workspace. In addition, the helper could also use the laser to point at various objects in the worker's environment. Nevertheless, in this solution, while the helper has control of the point of view, he or she is still constrained by the location and general direction of the worker's gaze, and the view is still jittery because of the worker's constant movements. Another solution designed to enable the movement of the point of view, but provides higher independence and control for the helper, is using a robot controlled by the helper at the workspace. GestureMan (Kuzuoka et al. 2000) used a mobile robot as a communication aid in remote collaboration. The remote helper could control the robot's movements and view the local environment using cameras located on the robot. A laser pointer placed at the front of the robot as well as a pointing stick were used to enable gesturing.

Due to advances in projection technology it is possible to replace the laser pointer with a small portable projector. This enables the option to project annotations in order to further enhance gesturing, while still allowing for projector mobility. A mobile projector can be combined with a video camera to view and augment a workspace (Junuzovic et al. 2012; Linder and Maes 2010; Mistry et al. 2009). Machino et al. (2006) implemented a system in which using a robot, a helper can see and move around in the workspace and project instructions using a projector mounted on the robot. The camera and projector are aligned so that the helper and worker have a shared field of view. In their solution the camera and projector were aligned using a half mirror constellation to try to bring the focal point of the camera to be near that of the projector. In our solution, we took a different approach and used real-time 3D image data processing to recognize the projected scene. Furthermore, in their solution, the helper could not project dynamic annotations or gestures to aid the worker in their task, but was limited to a static projected instruction. In our work, we focus on enabling both real-time gesturing options using a mobile projector, and on enabling both fine-grained movement of the gesturing device (by mounting the camera and projector on a robotic arm) and coarse-grained movement of the entire device in the workspace (by enabling robotic-type of movements for the entire device).

3. TeleAdvisor

Our goal was to design a versatile remote collaboration system that would enable a remote expert to easily provide relevant help to the worker. We wanted to emulate a situation in which a remote helper is looking over the shoulder of the worker, sees what the user is seeing, what the user is holding and how the equipment reacts, while being able to point to the physical object, and annotate it with relevant information to aid the local user's understanding while carrying on normal conversation. In addition, our main assumption was that the worker should see the technology as part of his or her working environment with minimal cognitive overhead. That is, the worker does not need to wear or operate the device. The worker's hands must remain free to do the work, and the device should be portable and versatile. Last, we wanted to use off-the-shelf equipment in order to keep the costs low. In this section we describe the general architecture of the system that we designed, following a summary description of the two phases of our prototype. In the next section, we elaborate on the challenges and design considerations we faced and on the implementation details of the prototypes.

3.1. System architecture

The system spans two areas: the workspace (local environment) where the worker is performing the task and the TeleAdvisor device is situated, and the remote environment where the expert is assisting. Gestures are made by annotations and graphics that are overlaid on the video stream using the remote advisor's interface. The annotations are sent back to the projector via the network and projected on top of the objects at the worker's location.

Design and Implementation of TeleAdvisor

The basic system architecture (Figure 2) remained the same throughout the development. In the local environment, the physical device includes the camera/s, projector and robotic arm, while the computation device includes the software that controls the physical device and communicates with the remote advisor's device. The local computation device includes a configuration module, video transmission, renderer and the robotic arm control engine. The configuration module starts when we open the device. It initiates a handshake with the remote advisor's interface sending information such as the camera and projector resolution, their coordinate system mapping, mechanical description of the robotic arm and desired frame rate. Additionally, available bandwidth testing is performed and video transmission quality is devised. The video transmission module is in charge of receiving the video stream from the camera, compressing it and sending it to the remote computer. The Renderer is in charge of accepting the annotations from the remote advisor's device, rendering the image to be projected and sending the image to the projector. Finally, the robotic arm control engine is responsible for accepting the movement commands from the remote computer and translating them to robotic arm movements.

On the remote advisor's side, the tracking module receives the video transmission from the local device. It first decompresses the video and then is responsible for tracking and understanding the projected image. The user interface module is responsible for showing the video stream of the remote environment to the user and for accepting user annotations and commands. Finally, the mapping engine accepts input from the tracking module and the user interface. It translates the tracking information and the annotations given by the user to annotations sent to the renderer in the local environment.

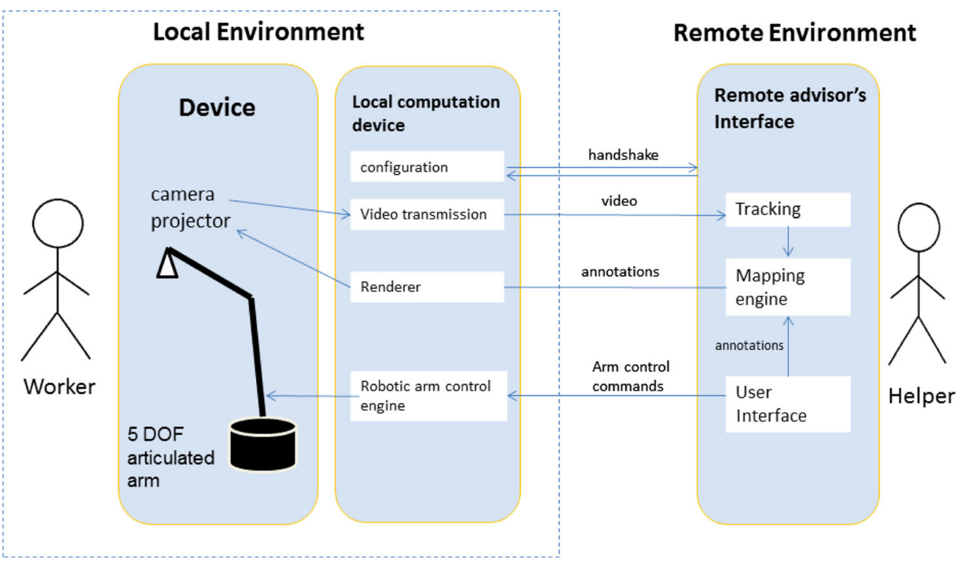


Figure 2. System architecture.

3.2. First prototype

The purpose of the initial prototype was to examine the feasibility of our solution and experiment with the notion of using projected AR as a tool for remote collaboration. For the end-device at the local environment, we used a single camera and pico-projector mounted on top of a custom-made robotic arm with 5° of freedom (Figure 3). Thus, the device can be placed by the worker anywhere in the workspace, and then the helper can use fine-tune movements of the arm to change the point of view. In order to synchronize between the camera's view and what was projected, we used active tracking of the projection area (by projecting on the boundaries and then tracking it) as described in Section 4.

3.3. Second prototype

The second prototype (Figure 4) aimed at improving and addressing several issues and weaknesses that we identified when experimenting with first prototype. The most important improvement is with the tracking and mapping between the camera and projector views. In order to improve tracking, especially for non-planar objects with depth variations, we changed the active tracking mechanism that was used in the first prototype. Instead, in the second prototype, we used stereo vision from two cameras in order to better recognize and be able to accurately project on any object at the scene. We also wanted to improve the mobility and reach of the device. We thus built a larger and more flexible robotic arm that could be more effectively used in manufacturing environments. To improve mobility, the device was placed on a robotic mobility base that moves under remote helper control. Finally, we have added a battery in order to ensure the autonomy of the end device. Changes and improvements were also made in the remote advisor's user interface.

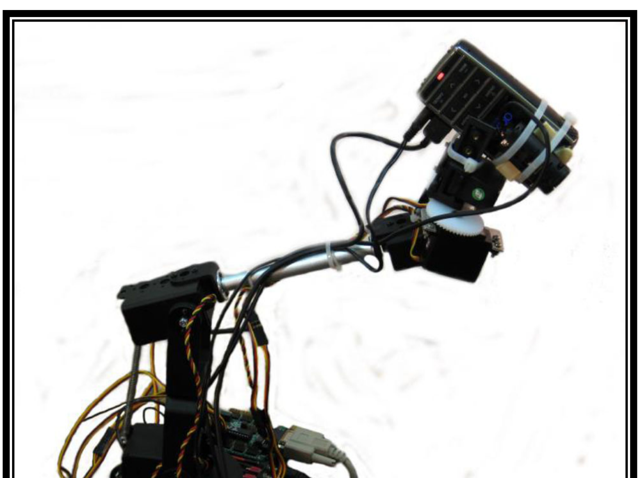


Figure 3. First version prototype end device.

4. Challenges, design considerations and implementation details

In this section, we describe the various challenges and design considerations that we encountered during our work, and provide a detailed description of the solution. The goal is both practical and academic. On the practical side, it is aimed to enable reproducing our work. On the academic side it aims to help understand the various considerations and tradeoffs when designing projected remote collaboration systems.

4.1. Mapping between camera view and projection

One of the main challenges in using a camera-projector solution is the need to synchronize between the camera and the projection views. Previous solutions using projected AR for remote collaboration assumed a known workspace in which the projector is top-mounted at a set distance from the objects (Kirk et al. 2007; Kirk and Stanton Fraser 2006). In such a solution, it is possible to pre-calculate and calibrate the camera and projector views to work for the specific workspace. However, our solution is mobile, and the device can be placed anywhere at the worker's environment which is not known in advance. Furthermore, the camera-projector device can be moved during the work, thus the environment is constantly changing. In order to project annotations at the right place, TeleAdvisor needs to have a mapping, which is updated in real-time, between each point in the camera view to its counterpart in the projector coordinate system.

4.1.1. *Motivation*

A major challenge in mapping between the coordinate systems of the camera and the projector is that it is not constant but rather is a function of the distance to the projected point. Figure 5 explains this in more detail. Figure 5a depicts 2 identical cubes at a different distance from the camera and a ray corresponding to a single point in the camera view. This virtual ray meets both objects at two different points

Figure 4. Second prototype end device.

(Figure 5b). If we apply a (fixed) mapping that corresponds which is correct for the closer cube (Figure 5c), we will be missing for the distant cube (Figure 5d). The mapping, or correction, for the distant cube is different (Figure 5e). This difference can be significant (Figure 5f). Thus, in our solution, we need dynamic mapping that can take into account the distance of the object to be projected on, and furthermore, can account to real-time changes in the distance between the object to be projected on and the camera/projector device. To determine the correction, we should measure the distance to the object first for each pixel in the camera frame (or more precisely, for each pixel in the camera frame that corresponds to a pixel in the projector) and then devise correct mappings for a set of points of interest – a different correction per point. More formally, we apply a view transformation between camera and projector. While this mapping is static, as long as camera and projector do not move, this kind of transformation requires 3D coordinates of the source points as input. Camera only provides 2D projections of those points, so we need to recover the missing coordinate – the distance, or depth which changes as a dependence of the objects in the view.

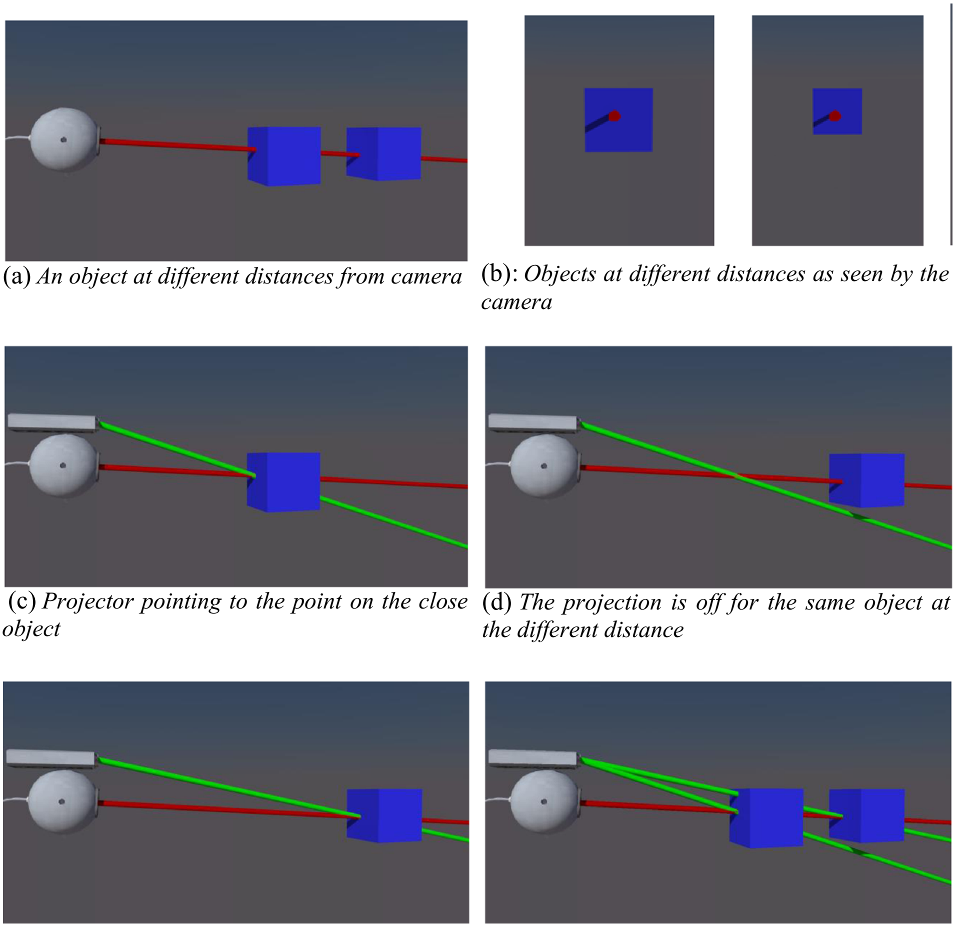
4.1.2. *Early version: projection read-back*

Our first approach to solve the mapping problem, which was implemented and tested with our first prototype, was to project a green thick rectangle to denote the boundaries of the projection area, and to read it back by the (single) camera. This rectangle was then tracked by image analysis means (i.e. computer vision). The projected rectangle also served an additional purpose of hinting the worker where the annotations could be placed. An implicit assumption was made that the surface to project on can be treated as a planar surface. This approach worked relatively well even on non-planar surfaces, as long as the projector was close to the camera and surfaces were nearly perpendicular to the projector/camera optical axis, or far away relatively to the projector camera distance.

To track the projected rectangle (which, in general, does not appear to be continuous and even not rectangular), we used several facts and assumptions:

- *The Rectangle is green*, hence the additional edges will be most strong in the green channel, while other channels can be used to filter out the edges which come from the objects' shapes;
- *Rectangle sides are nearly horizontal/vertical and long*, hence edges which disobey length or angle constraints can be filtered out;
- *Lines comprising the rectangle are significantly separated*, hence lines that are too close cannot come from the projected rectangle and must be filtered out;
- *Lines composing the rectangle have certain thickness*, hence lines that are too thin or too thick can be filtered out.

After the potential candidate lines were identified, their intersection was calculated to find the rectangle corners (the actual rectangle corners could even be obscured, but since we did not do corner detection, but rather computed the corner positions, that did not matter). Then, additional filtering was applied if more than 4 intersection



(a) *An object at different distances from camera* (b): *Objects at different distances as seen by the camera*

(c) *Projector pointing to the point on the close object* (d) *The projection is off for the same object at the different distance*

(e) *Different projector ray is required for different distance* (f) *Putting all together*

Figure 5. Mapping problem between camera and projection spaces. **a** An object at different distances from camera. **b** Objects at different distances as seen by the camera. **c** Projector pointing to the point on the close object. **d** The projection is off for the same object at the different distance. **e** Different projector ray is required for different distance. **f** Putting all together.

places were found, up to a tolerance. Finally, Kalman's filter and averaging filters were applied for smoothing corner movement and stabilizing the overall tracking mechanism. Fine-tuning of algorithm parameters allowed us to run multiple tasks in our laboratory without being limited by the tracking accuracy.

4.1.3. *Second version: stereo vision*

The early version performed fairly well as long as its basic assumptions were not violated. However, in many real-world applications these assumptions do not hold. For example, in machinery applications the surfaces are highly non-planar, the depth variation is big and sometimes there is no place to project the green frame on (e.g. cables in the air, a tip of a drill). To enable TeleAdvisor's usage in these environments, we had to examine a completely different approach that will resolve all the drawbacks of the single-camera solution. The biggest disadvantage of single camera is that one cannot easily and reliably determine the distance of the object (point) of interest to the camera, which is needed for constructing the correct mapping between camera and projector coordinate systems (see Figure 4). Thus, we employed a stereo vision solution using two cameras (see Figure 3) to create a real-time depth map of the environment. The core algorithm for depth map computation we use is Semi-Global Block Matching found in OpenCV which is based on (Hirschmüller 2008) with subpixel metric from (Birchfield and Tomasi 1998).

While there are several means today of creating depth maps in hardware (e.g., Microsoft Kinect, Intel RealSense) which have the advantage of being pre-calibrated and doing most of the work in hardware, we decided to use a custom stereo-pair vision based approach. The main reasons were that:

- With a readymade depth camera, an additional video camera for video transmission is required, while using a stereo pair inherently has a video stream to be presented to the helper;
- The depth camera to be used will not physically be co-located with the video camera, thus video to depth map correspondence is needed which is essentially the same problem as we're solving in camera-projector geometrical correction. To stress, when using Kinect-like solutions, for a point in camera view, it is not clear what is the correct point to look at the depth sensor, because of difference in camera and depth sensor locations – which is significant for short distances.
- Using a stereo pair allows to adjust correction algorithms in the future while depth camera hardware cannot be usually updated;
- Video camera resolutions are typically higher than depth camera resolutions;
- We used a camera with hardware encoding support up to Full HD resolution which allows for very low latency in round-trip while retaining the very high quality video streaming

To conclude, a stereo-vision solution provides higher flexibility, and since in TeleAdvisor all the processing is done on the helper side, the processing algorithms can be updated at any time, as well as helper workstation hardware, without any need to intervene into the deployed TeleAdvisor system.

4.2. Calibration

Before a complex system like TeleAdvisor can operate online, it needs preparatory steps, or *offline calibration*. Such a calibration is done once and then reused during every online operation. For example, before the system can operate, the exact distance between the camera and the projector should be measured (or, in general, view transformation should be determined). Before distance to surrounding objects can be determined, the attached cameras need to establish a high precision relation between each other, and in particular, to correct even minor distortions introduced by optics and their manufacturing technology. All these parameters are part of the model used by TeleAdvisor, and are to be fixed for a particular setup. Some of them, such as robotic arm length, can be measured directly, and some have to be derived from a special set of calibration data. The latter is usually an iterative process which includes step setup and feedback validation until convergence is reached.

Finding out all the unknowns can be expensive in terms of computation and time, but it is usually done once and does not influence algorithm performance in the run-time. A set of computationally cheap self-calibrations can be run during initialization or periodically in run-time to validate the stored calibration is valid or to adapt to changing conditions.

4.2.1. *Stereo-pair calibration*

Generating a depth map from a pair of cameras requires special calibration of the two cameras before they can be referred to as a *stereo-pair*. First of all, various kinds of distortions often observed even in high-end cameras, such as radial and tangential distortions, should be compensated. We model the camera as a perfect device with eight coefficients to allow for imperfections, as found in OpenCV (Bradski and Kaehler 2008). This process is standard and is performed once per camera. A set of images of a known object (we used a checkerboard image) is taken and then distortion coefficients are calculated and stored for later use. We used an asymmetric 8×11 checker-board to have a defined ordering of the internal corners for unambiguous detection. Undistorted images from the stereo-pair are then *rectified*. This is the process of converting the cameras into a pair of perfectly aligned cameras which have row-for-row (horizontal case) or column-for-column correspondence (vertical case) between pixels coming from the same objects. This allows for efficient depth map calculations. Our system uses vertical stereo-pair setup.

4.2.2. *Projector calibration*

The next stage of calibration results in a mathematical model of the system which provides offline mapping between camera (the one that is closest to the projector) and projector coordinates for any given depth. Our calibration process differs from the common approaches found in multi-projector systems calibration where camera-projector mapping is static (Raskar et al. 1999), and structured-light technique can be used for fast recovering of view transformation and projector distortions (Brown et al. 2005). It also differs from hand-eye calibration problems (Tsai et al. 1988) since we are not recovering camera location with respect to the robotic arm, but looking at the generic mapping between the camera and the projector. Furthermore, since the world around the arm is not static, and the arm itself can be repositioned, there is no special meaning for mapping camera coordinates to the world coordinate frame.

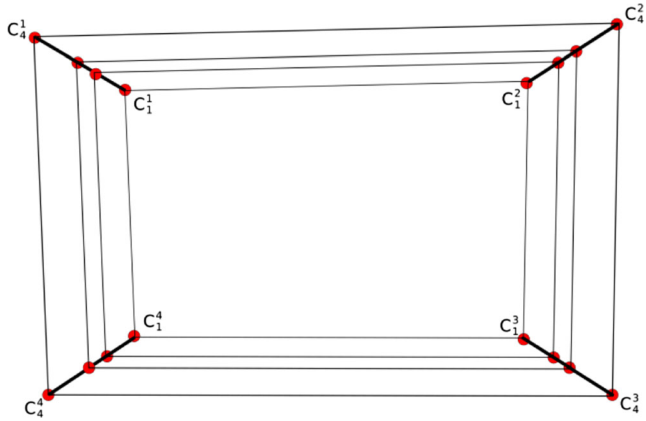


Figure 6. Calibration example. Projector corner positions as seen by the top camera. Four corner positions form a fragment of the corresponding epipolar line corresponding to this point. For example, $C_1^{1} - C_4^{1}$ correspond to the epipolar line of the upper left projector corner C_1. When the distance from camera to the point is also known, a view transformation can be calculated. Alternatively, we can devise a function C_i(d) for each corner by solving equation system which, in turn allows finding correct homography for a given depth d.

Contrary to some systems, we do not use external calibrated cameras at the known locations and/or patterns but rather use facilities built into the device: projector and calibrated (see. 4.2.1) stereo-pair. By doing so, we achieve the ability to perform field re-calibration of the system.

To calibrate, a known pattern (we used specially generated checkerboard with small additions solely for human handiness) is projected to a plane and is observed by the cameras from several different distances $\{D_1 \ldots D_n\}$ from the surface. Known features in the pattern are detected (we use corner positions to reduce numerical errors) in each of the frames F^T_{Di}, F^B_{Di} (top and bottom camera frames). As image quality has an immediate influence on the results, averaging and filtering were applied before detection which allow for sub-pixel feature detection accuracy. An image of each corner point C_i forms a portion of corresponding *epipolar line* (Zhang et al. 1995) in camera frame (see Figure 6). Additionally, the distance to each such point is known from the stereo-pair. At this point we can either recover view transformation as described in (Hartley and Zisserman 2003), or directly solve for each point without explicitly recovering view transformation matrices R and T (view rotation and translation) between frames of reference, and projection matrices K_c and K_p for camera and projector, respectively. These two approaches are equivalent though we used the latter.

This process does not compensate for projector distortions which are sometimes noticeable.[1] The distortions can be compensated by applying full-frame warping of

[1] For MicroVision ShowWX+HDMI these are especially noticeable in top left and right corners.

the image that goes to the projector at a later time. Warp parameters can be deduced as described in (Majumder et al. 2000; Raskar et al. 1999).

4.2.3. *Real-time correction*
After settling the model, the correction process is pretty straightforward. For each point to be mapped (e.g. each point in annotation), depth (disparity) is extracted from the current depth map. Then projector corners' locations are calculated for that depth (in camera coordinate system). A homography between these points in camera coordinate system and projector coordinate system is calculated. That gives the correction for the plane at the point's distance from the camera. This process is repeated for each query point. Below describes the algorithm for this process.

for each point $p_i \in P$

1. find disparity d_i
2. find projector corner coordinates c_i^j,j=1..4
3. find homography $H\colon \left(c_i^j\right) \rightarrow \left(C_i^j\right)$
4. $P_i=\mathrm{H}(p_i)$

end for

4.2.4. *Real-time depth map generation*
The process described above requires an accurate and smooth depth map, or more precisely, a disparity map which is the inverse of the depth map. This map is continuously (asynchronously) generated. We use SGBM (Hirschmüller 2008) algorithm from OpenCV for this purpose though other algorithms such as BM (Medioni and Nevatia 1985) and feature-based algorithms produce good results, too. The resulted disparity map is then filtered to remove outliers and false positives, as well as points with disparities that are out of the allowed range.

4.3. Network and video transmission

For a remote collaboration system to be effective, it is critical to have low-latency video transmission. It has been shown that even a short visual delay can hinder both situation awareness and conversational grounding, both important parts of a remote collaboration session (Gergle et al. 2013). We used an in-house low-latency network transport library to provide real-time reliable video stream transmission.

Currently, the helper locates the worker's device by feeding the worker's IP address or host name, in the address bar of the helper's UI. Directory services, as well as automatic discovery can be added to the system in future. The system is designed to support various types of TA devices. At the connection time, a handshake occurs during which the TA device configuration is transmitted to the helper, and the system reconfigures itself to match this particular TeleAdvisor's instance. The configuration includes the robotic arm configuration and mechanical sizes, video frame size and frame rate, camera-to-projector correspondence, and cameras details. Network speed is also tested at this stage to make sure there is enough bandwidth for fruitful co-operation.

4.3.1. *Encoding and decoding*

The system has two video transmission options, either two full-scale video streams[2] are transmitted from the TA device to the Helper's workstation (full processing option), or the main stream is transmitted full-scale, while the secondary stream (which is solely dedicated for depth map computation) is scaled down and transmitted in quarter-size to save bandwidth (network optimized option). Full processing mode provides slightly more accurate depth map, while requiring more bandwidth and more computing power on the helper's station. In our evaluations, we used the full processing option with no apparent delay problems.

For compression in the full processing option, we used in-camera compression hardware which produces a H264 encoded stream which is then transmitted. In the network-optimized option the compression is done in software because of the different cropping that is used by camera hardware for compression. The software-encoded stream uses VP8 real-time encoding. The hardware encoding introduces about 2 frame latency (~60 ms) according to our measurements, which is less than the frame time (30 ms). The helper side uses the FFMPEG library for decoding the video stream, hence, it can accommodate to wide range of video formats. Our streaming library makes it possible to encrypt the video stream to protect sensitive information. While we did not use it in our experiments, commercial users might find this option important.

4.4. Robotic arm and device movement

It is important that the helper have control over the point-of-view of the device at the workspace (Lanir et al. 2013). To attain fine-level control, the camera and projector were put on the top of a robotic arm controlled by the helper, enabling the helper to move the device's head and thus control the point-of-view.

4.4.1. *First prototype*

The first robotic arm we used (see Figure 2) was a commercially available hobbyist arm from Lynxmotion (AL5D with heavy-duty wrist) which was then customized to have a larger reach while still being able to hold a projector and a camera (camera had some parts removed to make it lighter). This arm has 5° of freedom (DoF) and is

[2] The second video stream is transmitted for the sake of depth map calculation at the Helper's workstation.

Figure 7. TeleAdvisor's second prototype robotic arm has a wide range reach.

controlled solely by servo motors which should constantly be powered on during the operation.[3] For autonomy purposes, the arm had an attached power source made from an adapted UPS with battery for several hours of operation.

Despite employing a five DoF arm, we only used 2° of freedom in actual experiments: base rotation and tilt, while other DoF options could be pre-set for a particular environment in the program settings. The reason for not using the other available DoF movements of the arm was our intention to keep the mental model of the device's control simple and understandable for the helper (see Section 4.6.1). In any case, we only developed the inverse kinematics engine, in the second prototype (see Section 4.4.3). The problem with this setting was the need to pre-set the remaining degrees of freedom before operation to get a generally convenient starting view of the work area.

4.4.2. *Second prototype*

The main limitations of the first prototype were a limited reach of the arm and the need for continuous power to be supplied to the device, even when steady. Besides drawing excess power while holding, servo motors produce an audible noise at 50 Hz (analog) or 500 Hz (digital) and are prone to resonance. To address these limitations, in the second prototype, we designed and built a custom robotic arm. We replaced the servo motors with different type of actuators, namely, linear actuators (LACs). This type of motor addresses both of the issues stated above as it is very compact, strong and does not draw power or produce noise while holding. In addition, we used longer joints in order to have a significantly extended arm reach (See Figure 7). The prototype itself was crafted from coated aluminum hollow rectangular tubes to reduce the weight of the arms. The second generation arm also has hardware switches that allow to operate each motor manually if the helper is absent and the worker

[3] This caused several servos to burn out.

needs to move the device, or in emergency situations. The arm's firmware also stores its parking position to which the arm moves back when not connected or shut down (to avoid having the device crash when connection is shutdown).

To further enhance the mobility of the device at the local's site, the second prototype was built on top of a mobile robotic base (see Figure 3, left). We used the Pioneer P3-DX which is a heavy duty industrial grade mobile base, capable of carrying up to 23 kg. The base can carry up to three high-capacity batteries which are also used to power the robotic arm and serve for lowering the overall center of mass and improving stability of the device. The mobile base is controlled via a serial connection using the supplied Aria software package. Its movements could be controlled by the helper using the helper's interface so that the entire device moves in 4 directions (2° of freedom movement).

4.4.3. *Inverse kinematics*

As there is no widely accepted and intuitive way of providing control for multiple degrees of freedom with a robotic arm, we used easily understandable abstractions: closer/farther, up/down pan, and left/right pan, that would be easily understandable to the helper when seeing the workspace from the robotic arm's head point of view. In order to implement these abstractions, the system should be able to know its position and camera direction in terms of motor positions as well as know its next position in the same terms. As the computation is done on the helper side, there is an additional requirement to perform computations in real time for various mechanical implementations of the robotic arm.

We used a parameterized model for arm description which covers a broad set of specific arm implementations. Then an *inverse kinematics* algorithm is able to compute how motors should be engaged to reach the desired position and camera direction. The system is also able to detect when the desired movement exceeds the physical range of the arm and notify the helper without an attempt to execute exceeding commands on the arm controller.

4.5. Autonomy

Because the TA end device needs to be moved anywhere in the environment, it is important that it would have full autonomous capabilities, not being attached by cable to any specific location. The second prototype of TeleAdvisor is designed to be fully autonomous for several hours and is equipped with a set of powerful batteries. These are used to operate the attached computer and to move the arm. The arm itself employs motor types (linear actuators) that only draw energy while changing their position and are able to hold the position without consuming energy. The robotic base has its own set of batteries which can also supply power for the rest of the system. For communication, the system is equipped with a wireless receiver/transmitter that is able to operate wherever there is a wireless network with sufficient bandwidth. Fourth-generation cellular networks can be used as a network provider using an extra modem.

4.6. Human factors aspects and user interface design at the helper's side

In this section we describe the human factors considerations and the design of the user interface on the helper side. We draw our design from studies conducted within the larger research field of teleoperated robots (Chen et al. 2007). Human performance issues involved in teleoperation generally fall into two categories: remote perception and remote manipulation. Remote perception deals with understanding the task and the remote environment. Poor perception has a detrimental effect on situation awareness which in turn, degrades performance when operating a remote device (Murphy 2004). In our case, the remote helper perceives the workspace using the view provided by the camera mounted on top of the end-device, providing an egocentric view of the environment (as opposed to an exocentric view in which the camera is placed on the body of the device). For a wide-view perception of the workspace, the remote helper can control the device's POV using control command movements of the robotic arm. Remote manipulation, on the other hand, deals with the manipulation of objects at the remote environment. In our case, these are the annotations that are projected on top of objects at the workspace and which are initiated by the helper. Next, we describe the remote perception and the remote manipulation aspects of the helper's user interface in our solution.

4.6.1. *Remote perception and control of the device's POV*

The helper's perception of the worker's environment relies on the video feed captured by the camera. It is important to have a high quality video communication since situation awareness relies on small environmental cues that help to build the helper's mental model of the environment (Darken and Peterson 2001). Factors such as reduced frame rate, reduced resolution of the display, lower grayscale, and most notably, latency of the transmission, were shown to reduce operator's performance (Chen et al. 2007). It is thus important to provide a high bandwidth and low latency communications that support high values with these factors (See Section 4.3, on how we deal with network and video transmission). Another issue that affects the remote perception is the field of view (FOV). It is important to have a wide as possible FOV when operating a remote device in order to avoid the "keyhole" effect (Woods et al. 2004). A limited FOV hinders peripheral vision and thus hinders target detection, location identification and situation awareness. In our case, we used a wide lens, high-end camera (Logitech HD Pro C920) to deliver Full HD (1080p) resolution at 30 frames per second. Still, a camera can only provide a limited view of the remote environment. We considered mounting another camera on the body of the device to provide an additional exocentric view, however, integrating information from two different perspectives might be difficult for the operator, and often switching between multiple views can be confusing (Chen et al. 2007). We thus used a robotic arm to allow view flexibility and control of the point of view by the helper. Finally, the camera's view does not usually align with the worker's viewing perspective. This creates a situation where the helper is looking at the scene from a different angle than the worker. From our experimentations (See Section 5) this was not deemed a major problem. The reason might be that the annotations, which are seen both by the helper and the worker act as a common frame of reference. However, in a more mobile scenario, or with the device located at possibly a wider angle when looking at the object at hand, the different viewing perspectives might cause some incoherencies for the helper.

In designing the user interface, our goal was to enable the remote expert full control of the movements of the device as well as control of various gesturing possibilities while being as simple and intuitive as possible. In addition, we wanted the user interface to be able to be operated from a standard desktop environment rather than use specialized equipment. Figure 8 shows the final version of the user interface. At the center of the interface is the video feed stream of the workspace. The helper can use two buttons (marked as "−"and "+" at the upper left corner of the video stream) to zoom in and out moving the view closer or further. To move the POV of the device, the helper can use the keyboard or the control buttons on the screen. Although the robotic arm allows five DOF, its movements are mapped into two dimensions on the Controller interface using inverse kinematics (See 4.4.3). We decided to narrow down the movement of the arm to two dimensions in order to match the helper's 2D view, and thus reduce the helpers' cognitive load. From our experimentations, 2 DOF with the addition of zoom were enough to reach a wide area of workspace including views of several planes. To increase the usability, we added a feature that we call *positional bookmarks*. At any time, the helper can

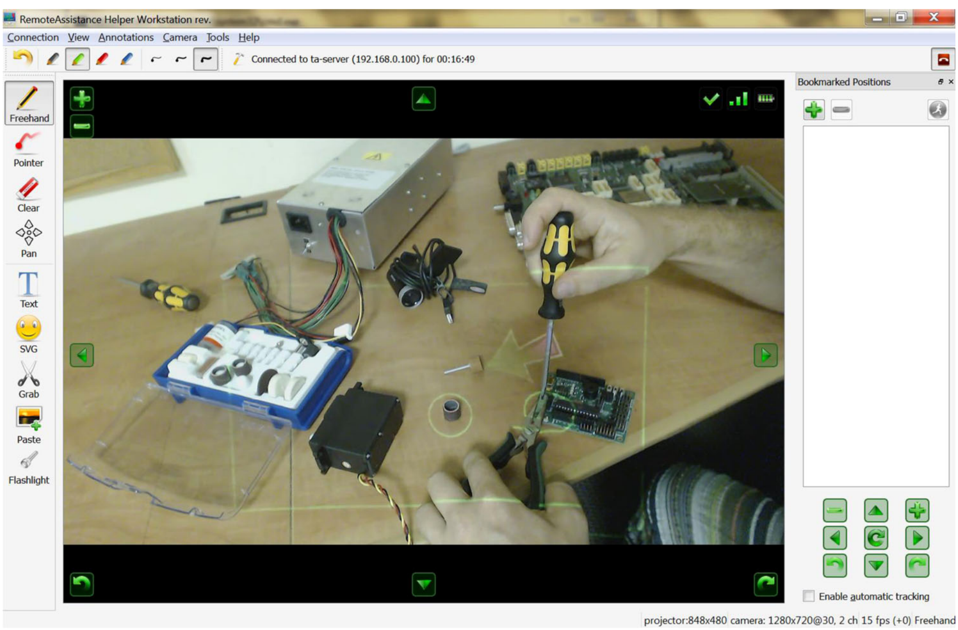


Figure 8. TeleAdvisor's user interface.

bookmark a view. Later, selecting this bookmark will move the arm back to the exact same position that was bookmarked. This can help save and return the view to important locations at the workspace. Finally, the local computational device continuously notifies controller's interface about its status: battery life, hardware command completion status, network channel quality – which are reflected in the user interface in the form of intuitively understood icons. This feedback integrates with the local feedback provided by inverse kinematics, for example, when the arm reaches its physical limit it notifies this in the UI.

4.6.2. *Remote manipulation (annotations)*

The helper can use a wide range of annotation tools in order to send the gestures to be projected at the workspace. The most basic tool for annotating is using free sketched ink. The helper can choose from four possible colors (default is black) and three possible line sizes. The helper can also choose to use a transient pointer that is only displayed while the mouse is pressed (see toolbars at the top and left side of Figure 8). In addition, the helper can choose a set image (or icon) to be placed at a specific location. For example, the helper can attach a pre-set "rotate clockwise" icon to a screw, or a pre-set warning icon in the user's environment. The helper can also add a "bubble" in which it is possible to enter text and attach it to an object. For example, the helper can mark a certain object with a bubble with the words "don't touch this". Another available option is the crop and paste feature. The user can crop part of the view and later paste it anywhere to show it back to the worker. This can be useful if the helper wishes to copy an image or scene and then project it on a white plane (a wall) and annotate on top of copied image. In the evaluations (see next section), undoubtedly, the most common use was of free sketching. Helpers used free sketching to circle objects, draw arrows (pointing at objects), draw lines and more. Most annotations were used as support to verbal explanations to show the location, orientation and action to be performed on objects. Very few helpers changed the size or color of the annotations. The pointer feature was also used, mostly to show transient gestures. That is, to point at various objects. The other features (the icon, bubble or crop and paste) were rarely used.

Stroke annotations that are overlaid on top of the helpers' view are erased from the helper's view after a pen-up event. After that, the real projected annotations are seen by the helper through the video captured view of the worker's workspace. Finally, in order to fine-tune the placement of annotations, it is possible to move sketch, image or text annotations by a few millimeters by pressing the arrow keys just after drawing each annotation. We considered enhancing the helpers' view with the annotation drawings to provide a type of predictive display (Sheridan 2002), but after several experimentations decided against it. The main reasons being that firstly, the actual latency we managed to achieve was low enough to not notice the delay in the projection of the annotations, and second, the annotations did not always align pixel-to-pixel with the real-world projected annotation causing visual clutter and misunderstanding. In addition, we felt it diminished helpers' understanding of what exactly the worker sees and how the annotations are perceived by the worker.

4.6.3. *Other possible user interfaces for the helper*
In the current implementation, the system is operated using a standard mouse-based GUI that can be run on any Windows-based computer. This has the advantage of easy and flexible deployment anywhere (e.g., in the helper's house). Since annotations are drawn on top of the video feed, using pen or touch gestures to create the annotations can be more natural. A large touch display can be thus used similar to (Alem et al. 2011; Tecchia et al. 2012) to allow easier annotation drawing. This is supportable by the current version with single-touch gestures. Also, a hand-controller (i.e., a joystick) can be used instead of the mouse to control the movements of the device and the robotic arm as is common in many remote Teleoperation interfaces (Fong and Thorpe 2001). A solution of a joystick for device movements can be combined with touch, pen or mouse for annotation, since annotations are only made when the device is not moving. Finally, we also considered a master–slave constellation for the control of the robotic arm in which a small model of the end device's arm would be moved by the helper in order to move the arm at the workspace. However, such a constellation is preferable with an exocentric view of the environment that views the end-device from faraway.

5. Evaluation

We evaluated TeleAdvisor in several phases. The initial prototype was first evaluated with users in our laboratory to examine the feasibility of the solution and to test usage of features and user acceptance. We then employed a more formal, quantitative evaluation focusing on the question of who should control the point-of-view (POV), the helper or the worker. Feedback for the second prototype was gathered from two deployments: (1) a young scientist visitor's day in which many students experimented with the system functions, and at a more formal deployment at the Advanced Manufacturing Research Center (AMRC) located in Sheffield UK. The AMRC works with major advanced manufacturing companies like Boeing, Rolls-Royce, and others, to research and develop new technologies aimed to meet challenging manufacturing problems.

For the first evaluation, three recruited participants were trained to serve as helpers, while ten recruited participants served as workers. Two tasks were given. The first was a Lego assembly task which is common in evaluations of remote collaboration systems (Fussell et al. 2004; Kirk et al. 2007; Kirk and Stanton Fraser 2006). The task consisted of connecting small pieces of Lego (see Figure 9) where the helper held the manual for the given model while the worker constructed the model using the helpers' instructions. The second task consisted of connecting various cables between a TV set, a DVD and an AV receiver box. Both tasks were designed to simulate a 3D scenario in three planes. Results indicated that helpers were able to effectively use TeleAdvisor to complete the tasks. Helpers repeatedly used annotations, accompanying them with phrases such as: "put it here" and "move it there" to establish common ground using deictic references. Helpers mostly used the free sketch tool for annotation. Pointer was used a few times while other features were seldom used. Helpers explained that sketches were most useful for deictic gestures, highlighting a specific object or location (usually by circling it), while pointers were useful for more accurate and transient gestures. The type of annotations made varied and mostly included pointing to objects and their target locations. Few annotations were used to draw sketches or indicate orientations of pieces. This is in line with the usage of annotations reported in (Fussell et al. 2004). Helpers heavily relied on the movement of the robotic arm and on the use of position bookmarks since both tasks required a wider point of view of the workspace than the camera provided. Both workers and helpers reported that TeleAdvisor was intuitive, very useful, and saw much value in having projected annotations in such a setting.

In the second evaluation, a more formal evaluation was conducted (Lanir et al. 2013). Twenty four participants took part in the experiment. Four were assigned the roles of helpers, and 20 participants were assigned the role of workers. The same tasks of constructing a Lego model and wiring a TV set were used. The study focused on the issue of control of the gesture device, comparing whether and when it should be in the hands of the worker or the helper. Two versions of the end-device were used, one that is controlled by the helper (the first prototype described in this paper) and a camera-projector set that was mounted on a lamp-like device, which movements could only be controlled by the worker (See Figure 10). Thus, a within-study design was performed with the main variable being *helper-control* vs. *worker-control*. Results of the study suggests that, in general, the control of the POV is better left at the hands of the helper. When comparing task performance, results in the Lego task did not show a significant difference. However, results of the wiring task showed faster completion time in the helper control condition compared to the

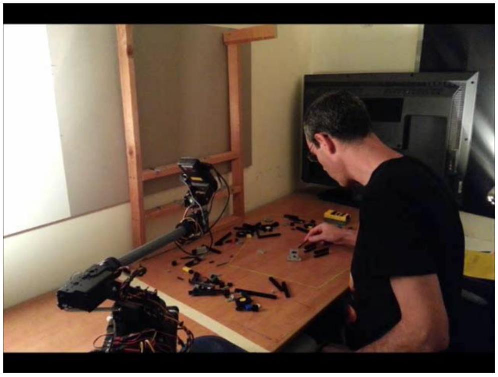


Figure 9. Lego construction task in evaluation.

worker control condition. We also compared the number of times the POV was moved in each condition. Results show that in both tasks, there were significantly more movements in the helper-control condition compared to the worker-control condition (average of 24.5 movements per Lego task in the helper-control condition vs. 8.2 movements in the worker-control condition). Finally, we also compared the number of gestures performed by the remote helpers. Results indicated that in Lego task, helpers employed more annotations in the helper-control condition. Looking at user behavior, we noticed that when the helper had control, helpers were more actively involved and had better task awareness. This is reflected in the quantitative results that show more device movements and more gestures in the helper-control condition. Another interesting observation was that the helper condition afforded parallel work in which the helper performed searching tasks in parallel to the worker. We concluded that workers need to focus on the task at hand and thus, controlling the POV of the device adds excessive cognitive efforts to their work which is not recommended. On the other hand, helper's task is to follow and assist the worker, and thus controlling the POV of the device is a natural and even advisable subtask as it helps their situation awareness. These results support our notion of using a robotic arm controlled by the helper to enable the helper flexibility and control of the POV. Subjective opinions of both workers and helpers confirm this, indicating that both helpers and workers thought that control should be left in the hands of the helper. Furthermore, workers rated the helper-condition significantly better on ease to complete the task, and on their perceived performance.

Following these studies, many improvements were made to the design of the robotic arm and depth sensing capability was added to improve the accuracy of the projected AR overlaid graphics on real-world 3D objects. A much larger prototype (version 2) was built that had a much longer reach for the robotic arm and could easily deal with any item of interest from floor level to ceiling level. This version 2 prototype was first experimented at our laboratory. During a young scientist day,

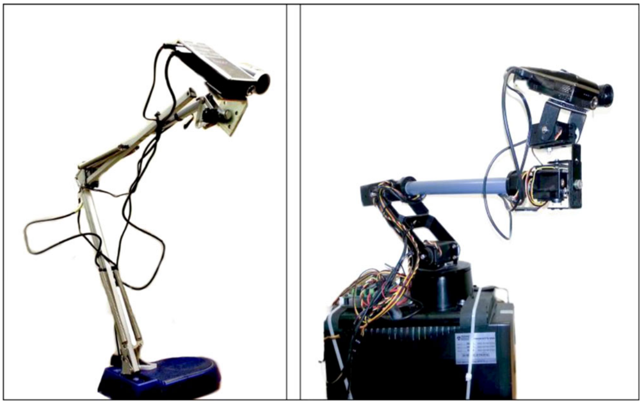


Figure 10. The camera and projector configurations for the second evaluation. On the *left*, the worker controls the device by manually moving it. On the *right side*, the helper remotely controls the device using the robotic arm.

many kids and students experimented with the prototype providing many comments on the usability and usefulness of the device. Most reactions were very enthusiastic, and most users, both acting as helpers and workers were able to easily understand how to use the system. Following this initial trial, we deployed the device at the Advanced Manufacturing Research Center (AMRC) in Sheffield UK, as part of an integrated demonstration system developed by IBM to support manufacturing and repair operations. The TeleAdvisor prototype at AMRC is demonstrated to interested manufacturing companies who visit the AMRC. Since the TeleAdvisor system was installed at the AMRC is was demonstrated to key technical people from several engineering and manufacturing companies. Feedback was very positive and the prototype was said to be an ideal solution for remote technician support, remote inspection, training and mentoring. For example, one CTO of a very large manufacturing company said that TeleAdvisor provides a solution for a real problem they have in which older, expert engineers, who prefer not to travel so much, need to support manufacturing and repair at remote overseas sites.

6. Discussion and future work

During our work, we have encountered several important issues that we would like to layout and discuss here. These points are common in most remote collaboration systems and thus, may need to be considered when designing any type of system that uses AR for remote collaboration. We discuss these issues following the limitations of our solution and possible future work.

6.1. Supporting mobility

In order for a remote collaboration system to be flexible, it must support the mobility of the worker within the workspace. The worker should be able to freely move around, look at the objects from different directions and work at locations that are not necessarily determined ahead of time. While initial works examined fixed, pre-defined workspaces in order to focus on other issues that accompany remote collaboration (Kirk et al. 2007; Kirk and Stanton Fraser 2006; Tang et al. 2006; Junuzovic et al. 2012), it is clear that a complete solution must also consider mobility. Our solution implements *fine-grained mobility* of the device's POV in order to be able to better see a single workspace area (similar to (Birnholtz et al. 2010)), and *higher level mobility* of the device itself to be moved from one workspace area to another. The assumption made is that when focusing on a task, the device is set in one location, and there is only need for fine-grained mobility of the device to achieve better resolution or better reach (e.g., look at different angles). The fine-grained mobility is implemented using the movement of the robotic arm. We have shown that it is preferable that the helper control the POV of the device, and thus, the movements of the robotic arm are controlled only by the helper. Higher level mobility in the first prototype was achieved by designing the prototype to be self-contained and transportable so it could be placed by the worker anywhere at the workspace. That is, it is designed to be initially placed or moved at the workspace by the worker, and after being placed, the helper can employ fine-level movements. In the second prototype, we improved the mobility of the system by also enabling the helper control of the high-level movements of the device, placing the entire device on a robotic base.

A different way to support workspace mobility is to use an HMD solution (Alem et al. 2011; Fussell et al. 2003; Kraut et al. 2003). An HMD solution has the advantage of being worn by the worker and thus being naturally portable anywhere at the workspace. It also has the advantage of implicitly showing the helper where the attention of the worker is at any given time. However, an HMD provides an unstable, jittery view for the helper, and may require the worker to unnaturally limit his or her head movements. Furthermore, worker's head movements also change the view for the helper (e.g., the worker might look aside when she hears someone walk by), which may be confusing and difficult to resume focus when looking back at the workspace. Our solution combines the advantages of portable AR along with the advantages of a stable view and the helper's control of the POV.

6.2. Human factors aspects

At the worker's side, the worker needs to focus on performing the work, and thus there should be minimum cognitive overhead in operating the device and understanding the instructions. A projected AR solution has the advantage of keeping the worker's hands free to work on the task. In addition, instructions from the helper are directly overlaid on top of the objects. This enables natural gesturing and reduces the cognitive effort that would be needed in switching between displays with a screen-based solution (either a fixed screen placed in the workspace or an HMD). From the helper's side, it is the helper's main task to follow and assist the worker. Helpers should maintain an ongoing awareness of what the worker is doing, the status of the task, and the environment. This is often referred to as *situation* or *task awareness* (Kraut et al. 2003). When the helper is in control of the POV, the helper is more actively involved. This involvement, although not necessarily improving performance, improves helper's task awareness by monitoring the task status, which in turn improves the overall communication between helper and worker (Lanir et al. 2013). It is important that the helper's interface for controlling the POV be as intuitive as possible to reduce extra effort of the helper. In the current solution we tried to accommodate this by mapping the movements of the robotic arm (which had 7 DOF) to natural user representations.

We have supported several simple controls for creating annotations. Indeed, we saw in our experimentations that users mostly used free sketch annotations (mostly to circle and mark objects or locations) as well as the pointer device (for transient gesturing). It is possible to support other kinds of gesturing such as rendering of the hands, or more sophisticated gesturing. However, we noticed that users were quite happy with the available features and did not require other types of gesturing. Control of annotations was done using the mouse on a standard GUI interface. We believe that using a large touch screen would probably improve the ease and accuracy of the gesturing. Finally, it is also possible to improve the control of the device's movements by using a joystick or another type of physical controller.

6.3. Comparison with other approaches

In this section, we compare our system with other existing approaches, discussing the advantages and disadvantages of each approach. We first compare our system with other camera-projection systems, followed by a comparison of our system with other non-projection AR approaches for remote collaboration.

Several previous camera-projector systems used a fixed projection in which a camera and projector pair is mounted in a fixed position. This includes works by Kirk and colleagues (Kirk and Stanton Fraser 2006; Kirk et al. 2007) who used a camera-projector system mounted on the ceiling to examine projection-based remote gestures, and the Illumishare project (Junuzovic et al. 2012) that used two stationary devices to create a symmetric shared workspace in which each user can share and annotate physical or digital objects. A fixed camera-projector system is easier to implement and does not need continuous mapping between the camera and the projector view (see Section 4.1). However, it is limited in its reach and versatility. Since TeleAdvisor is easily transportable, it can be placed (or in our second prototype, can reach by itself) anywhere in the environment. For example, a worker can take and place TeleAdvisor anywhere in a large manufacturing plant, to help fix some unexpected problem. This is especially useful in ad-hoc situations when help is needed in unforeseen or large areas that cannot be instrumented ahead of times. For large industrial plants, the cost of setting up many (possibly thousands of) fixed camera projector pairs in every location and orientation that might 1 day facilitate remote assistance would be impractical to both install and maintain. Furthermore, the articulated arm based TeleAdvisor system can see and project at the required angles and from any direction on stationary items. For example, if a large, complex jet engine is being fixed, the device can be placed in different positions beside, under or even inside the engine. A stationary constellation is limited to a single point-of-view. This versatility of reach is illustrated in Figure 7. Other approaches, such as Kuzuoka et al. (2000), enabled mobility by using a mobile robot device, but used a pointer-only system as a gesturing device. A simple pointer is less expressive in its ability than the complete annotation tools available with a projector. One difference is that a pointer is transient in nature. It only allows focus of a single point in time. This supports deictic gesturing, but fails to support more complex types of communication. With projection, it is possible to have persistent representations that can support a wide-range of activities and explanations (Anderson et al. 2004). For example, the helper can draw individual circles and numerals (1–5) around five screws that need to be removed in a certain order. The worker sees these circles and numerals as persistent representations that help guide the work and reduce errors. A second difference is that pen-based annotations can support representational gesturing (e.g., showing angle or rotations for describing how to insert or manipulate objects) in addition to deictic gestures. Fussell et al. (2004) have shown that a pointer alone showed no benefit over video-only connection, while a pen-based drawing tool led to significant improvements in performance time over video alone. They noted that while less frequent, representational gestures are a crucial component of conversational grounding. In our experiments, we saw pen-based gestures used to mark multiple items, annotate directions and more complex sketches, and even project ready-made images.

Our solution uses a projected-AR approach in which instructions are projected over the actual objects. An alternative solution uses see-through AR, in which the user holds a mobile device (i.e., a Tablet computer), and can see the helper's gestures and augmentations on top of the device's camera feed (Gauglitz et al. 2012, 2014). While a Tablet-based system is simpler and possibly more accessible, there are several advantages to our solution. First, seeing the helper's annotations through a mediator (i.e., the Tablet device) creates a fractured or mixed ecology in which the worker needs to split attention between the task and the external display, increasing the cognitive load of the worker and hindering the performance (O'Neill et al. 2011). The 'mixed ecologies' perspective argues that the design of remote gesturing tools should attempt to recreate gesturing in a format closely aligned to normal co-present gesturing as possible (Kirk et al. 2005). With a Tablet, the worker needs to continuously shift his or her attention from the Tablet to the real-world object. In our solution, the user sees the helper's annotations as a natural part of his or her own working environment. It does not require the user to wear or hold anything and it naturally combines the instructions with the objects allowing the worker to focus on the task at hand. Second, using a Tablet device, the helper is constrained to look at the specific location that the worker aims the Tablet at. The worker controls what and where the focus of attention is at any given time. If the worker wants to see a different area, he or she needs to specifically ask the worker to move the orientation of the device. With a robotic arm solution, the helper has control over the point-of-view and can decide where to look at. We have shown that providing helpers with control of the point-of-view increased their active involvement, and in some cases improved the overall task performance (Lanir et al. 2013). We have also shown that when the helper has control of the point-of-view, the helper can perform parallel work and has increased task awareness. The workers need to focus on completing the task, and having the worker also control the camera's view (such as in a Tablet solution) adds to the cognitive load of the worker and is difficult for the worker. Third, using a projected-AR approach, both worker's hands are free to perform the required operations. With a see-through Tablet solution, the user holds the Tablet device in one hand and can only perform operations using the other hand. This may be bulky and ineffective, especially for more complex actions which may require the use of both hands (for example holding a screwdriver in one hand and securing the object in the second hand). However, this can be mediated by placing the Tablet on the work surface or using some kind of holder to free worker's hands.

Finally, another approach is an HMD-AR solution. In this solution, the worker wears an HMD on which the annotations are drawn. This way, annotations are seen on top of the objects following the mixed ecology perspective. However, control is again left solely to the hands of the worker, completely following the worker's point-of-view. The view for the helper might be jittery because of the worker's small head movements. Furthermore, if the worker moves his or her head to the side, following some external noise, the helper's view follows the worker. This may cause difficulties on the helper in retaining his or her situation awareness. In addition, most current HMD-AR devices are still a bit cumbersome, with low resolution and may feel somewhat unnatural. However, this will surely improve with the advent of technology.

In summary, each solution has its advantages and disadvantages. We argue that our solution provides much versatility supporting both flexibility of movements, representational gestures, a mixed ecology perspective, and control of both worker and helper. Albeit, a see-through Tablet solution might be cheaper and easier to deploy as it only requires a Tablet device and a software implementation. An HMD solution might be advantageous when one wants to completely follow the worker's point-of-view. We believe our solution is better fitted to support complex operations in which the task may be composed of several steps and may require complex physical manuvers. For these kind of tasks (e.g., fixing a complex jet engine), not needing to shift attention between an electronic device and the object, and having the helper be able to control the point-of-view may be advantagous. On the other hand, for simple tasks, with a possibly wider deployment (e.g., fixing a sink's plumbing at home), a Tablet see-through solution might be preferred. Nevertheless, it is clear that at this point, all three solutions should be further examined. Further studies will directly compare between HMD, Tablet and projected AR solutions, as well as examine these solutions in real-world scenarios.

6.4. Limitations

While projected annotations can provide useful inline augmentations for remote collaboration, there are several limitations with the suggested approach. First, the lighting condition in the room may affect the way the projections are seen by the worker. It may be difficult to see the projections in a highly lit room or if operating outdoors. In our prototype, we used a 20 lumens pico-projector. The lighting conditions in the evaluation were of a slightly dimmed office room and participants were able to easily see and recognize all annotations and perceived the projector to be sufficient for this use. Still, we saw that in direct and high lighting conditions, it can be difficult to see the annotations. With the advent in projector power, with 200-lumens pico-projectors already available, we believe that at least for indoor situations, this limitation can already be addressed. Second, the condition of the surface may also affect the way the projections are seen on it. If the surface is composed of varied materials with different heights, it may be difficult to see the projected annotations. For example, we experimented with the task of repairing a desktop computer. Because of the many different type and height of components inside the computer, it was often difficult to see exact projections. Finally, a projected solution requires specialized hardware and thus might be more expensive than an alternative Tablet solution. Nevertheless, in our prototype we used off-the-shelf components that are not too expensive on their own.

6.5. Future work

A promising direction for future work is combining human-guided remote assistance with automatic computer-guided assistance. Many studies examine how automated solutions using AR technologies can aid in various instruction-focused or training type tasks (Baird and Barfield 1999; Henderson and Feiner 2011, 2009; Neumann and Majoros 1998). However, most of these automated assistance solutions are limited both to pre-existing workspaces or tasks and in the scope of their explanations. It is clear that a human operator is still much more flexible in both understanding the current worker's task context and problem, and in adapting his or her knowledge to unfamiliar workspaces. Still, a human expert resource is often scarce, and experts are not always available. Thus, a combination of automatic computer-based guidance, with a single human expert that can monitor several remote locations and can intervene when needed can be very useful. Such a solution should include ways for the human expert to see and augment the automatic solutions and ways for the worker to ask for the expert's assistance. A similar direction would consider a constellation in which only human-to-human assistance is available, but one helper may assist several workers. An interface that would allow continuous monitoring of several locations and easy switches of focus between workers (assuming voice and gestures are only available at one location at a time) is critical for the success of such a system.

A possible improvement of TeleAdvisor would look at anchoring annotations to specific locations using computer vision techniques allowing the object or the device to move while keeping the annotations in the same relative location. This has been explored in (Gauglitz et al. 2012) who showed user preference and slight improved performance for anchored annotations when using an AR see-through solution (using a Tablet) in remote collaboration. It seems that in our solution as well, anchored annotations can be helpful. In addition, there is room for improvements in the user interface of the helper. It is likely that annotations would be better drawn using a touch or a pen interface. The control interface for the device's fine-grained POV as well as the high level movements of the device should be closely examined, and other solutions such as physical control should be explored. Finally, as we mentioned before, future work would directly compare see-through (Tablet), HMD and projected AR solutions to quantify and examine the benefits and disadvantages of each solution.

References

Alem, Leila, Franco Tecchia, and Weidong Huang, (2011). HandsOnVideo: Towards a Gesture based Mobile AR System for Remote Collaboration. In *Recent Trends of Mobile Collaborative Augmented Reality Systems*. New York: Springer, pp. 135–148. doi:10.1007/978-1-4419-9845-3

Anderson, Richard J., Crystal Hoyer, Steven A. Wolfman, and Ruth Anderson. (2004). A study of digital ink in lecture presentation. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. New York: ACM Press, pp. 567–574. doi: 10.1145/985692.985764

Azuma, Ronald. (1997). A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, vol. 6, no. 4, pp. 355–385. doi:10.1.1.30.4999

Azuma, Ronald, Yohan Baillot, Reinhold Behringer, Steven Feiner, Simon Julier, and Blair MacIntyre (2001). Recent advances in augmented reality. *IEEE Computer Graphics and Applications*, vol. 21, no. 6, pp. 34–47. doi:10.1109/38.963459

Baird, Kevin M., and Woodrow Barfield (1999). Evaluating the effectiveness of augmented reality displays for a manual assembly task. *Virtual Reality*. vol 4, no. 4, pp. 250–259. doi:10.1007/BF01421808

Bauer, Martin, Gerd Kortuem, and Zary Segall. (1999). Where are you pointing at? A study of remote collaboration in a wearable videoconference system. In *Third International Symposium on Wearable Computers*, *San-Fransisco, CA, 18–19 October, 1999*. Washington, DC: IEEE computer society, pp. 151–158. doi:10.1109/ISWC.1999.806696

Billinghurst, Mark, and Hirokazu Kato (2002). Collaborative augmented reality. *Communications of the ACM*. vol. 45, no. 7, pp. 64–70. doi:10.1145/514236.514265

Birchfield, Stan, and Carlo Tomasi (1998). A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 20, no. 4, pp. 401–406. doi:10.1109/34.677269

Birnholtz, Jeremy, Abhishek Ranjan, and Ravin Balakrishnan (2010). Providing Dynamic Visual Information for Collaborative Tasks: Experiments With Automatic Camera Control. *Human-Computer Interaction*. vol. 25, no. 4, pp. 261–287. doi:10.1080/07370024.2010.500146

Bradski, Gary, and Adrian Kaehler (2008). *Learning OpenCV: Computer Vision with the OpenCV Library*. Sebastopol, CA: OReilly Media Inc. doi:10.1109/MRA.2009.933612

Brown, Michael, Aditi Majumder, and Ruigang Yang (2005). Camera-based calibration techniques for seamless multiprojector displays. *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 2, pp. 193–206

Chen, Jessie YC, Ellen C. Haas, and Michael J. Barnes (2007). Human performance issues and user interface design for teleoperated robots. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on,* vol. 37, no. 6, pp. 1231–1245. doi:10.1109/TSMCC.2007.905819

Darken, Rudolph P., and Barry Peterson (2001). Spatial Orientation, wayfinding and representation. In K. M. Stanney (Ed.): *Handbook of Virtual Environment Technology*. Mahwah, New Jersey: Lawrence Erlbaum Associates, pp. 493–518

Flagg, Matthew, and James M. Rehg (2006). Projector-Guided Painting. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology (UIST 2006)*. New York: ACM Press. pp. 235–244. doi:10.1145/1166253.1166290

Fong, Terrence, and Charles Thorpe. (2001). Vehicle teleoperation interfaces. *Autonomous Robots*, vol. 11, no. 1., pp 9–18. doi:10.1023/A:1011295826834

Fussell, Susan R., Leslie D. Setlock, and Robert E. Kraut. (2003). Effects of head-mounted and scene-oriented video systems on remote collaboration on physical tasks. In *Proceedings of the Conference* *on Human Factors in Computing Systems - CHI'03*, pp. 513–520. New-York: ACM Press. doi:10.1145/642611.642701

Fussell, Susan R., 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*. vol. 19, no. 3, pp. 273–309. doi:10.1207/s15327051hci1903_3

Gauglitz, Steffen, Cha Lee, Matthew Turk, and Tobias Höllerer. (2012). Integrating the physical environment into mobile remote collaboration. *Proceedings of the 14th International Conference on Human-Computer Interaction with Mobile Devices and Services - MobileHCI'12*. New-York: ACM Press. pp. 241–250. doi:10.1145/2371574.2371610

Gauglitz, Steffen, Benjamin Nuernberger, Matthew Turk, and Tobias Höllerer (2014). World-stabilized annotations and virtual scene navigation for remote collaboration. *Proceedings of the 27th annual ACM symposium on User interface software and technology.* New-York: ACM Press. pp. 449–459.

Gergle, Darren, Robert E. Kraut, and Susan R. Fussell. (2013). Using Visual Information for Grounding and Awareness in Collaborative Tasks. *Human-Computer Interaction*, vol. 28, no. 1, pp. 1–39. doi:10.1080/07370024.2012.678246

Gurevich, Pavel, Joel Lanir, Benjamin Cohen, and Ran Stone. (2012). TeleAdvisor: a versatile augmented reality tool for remote assistance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New-York: ACM Press. pp. 619–622. doi:10.1145/2207676.2207763

Hartley, Richard, and Andrew Zisserman (2003). *Multiple view geometry in computer vision*. New York: Cambridge university press.

Henderson, Steven J., and Steven Feiner. (2009). Evaluating the benefits of augmented reality for task localization in maintenance of an armored personnel carrier turret. In *Science and Technology Proceedings - IEEE 2009 International Symposium on Mixed and Augmented Reality, ISMAR 2009*. Washington DC: IEEE Computing Society, pp. 135–144. doi:10.1109/ISMAR.2009.5336486

Henderson, Steven J., and Steven Feiner (2011). Exploring the benefits of augmented reality documentation for maintenance and repair. *IEEE Transactions on Visualization and Computer Graphics*, vol 17, no. 10, pp. 1355–1368. doi:10.1109/TVCG.2010.245

Hirschmüller, Heiko (2008). Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2., pp. 328–341. doi:10.1109/TPAMI.2007.1166

Huang, Weidong, and Leila Alem (2011). Supporting hand gestures in mobile remote collaboration: a usability evaluation. In *Proceedings of the 25th BCS Conference on Human-Computer Interaction*. Swinton, UK: British Computer Society. pp. 211–216.

Huang, Weidong, Leila Alem, and Franco Tecchia. (2013). HandsIn3D: supporting remote guidance with immersive virtual environments. In *Human-Computer Interaction–INTERACT 2013 lecture notes in computer science*. vol. 8117, pp. 70–77. Berlin: Springer

Ishii, Hiroshi, Eran Ben-Joseph, John Underkoffler, Luke Yeung, Dan Chak, Zahra Kanji, and Ben Piper. (2002). Augmented urban planning workbench: overlaying drawings, physical models and digital simulation. In *Proceedings of Ismar'02.The first International Symposium on Mixed and Augmented Reality*. Washington DC: IEEE Computer Society, pp. 203–211 doi:10.1109/ISMAR.2002.1115090

Junuzovic, Sasa, Kori Inkpen, Tom Blank, and Anoop Gupta. (2012). IllumiShare: sharing any surface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* New York: ACM Press, pp. 1919–1928. doi:10.1145/2207676.2208333

Kirk, David, and Danae Stanton Fraser (2006). Comparing remote gesture technologies for supporting collaborative physical tasks. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*. New York: ACM Press, pp. 1191–1200.

Kirk, David, Andy Crabtree, and Tom Rodden (2005). Ways of the Hands. In *Proceedings of the ninth european conference on Computer-Supported Cooperative Work 18–22 September, Paris, France*. pp. 1–22. Springer Netherlands.

Kirk, David, Tom Rodden, and Danaë Stanton Fraser (2007). Turn it THIS way: grounding collaborative action with remote gestures. In *Proceedings of ACM CHI 2007 Conference on Human Factors in Computing Systems*. New York: ACM Press, pp. 1039–1048. doi:10.1145/1240624.1240782

Kraut, Robert E., Susan R. Fussell, and Jane Siegel (2003). Visual Information as a Conversational Resource in Collaborative Physical Tasks. *Human-Computer Interaction*. vol. 18, no. 1, pp. 13–59. doi:10.1207/S15327051HCI1812_2

Kurata, Takeshi, Nobuchika Sakata, Masakatsu Kourogi, Hideaki Kuzuoka, and Mark Billinghurst. (2004). Remote collaboration using a shoulder-worn active camera/laser. *Eighth International Symposium on Wearable Computers*, vol 1. Washington DC: IEEE Computer Society, pp. 62–69. doi:10.1109/ISWC.2004.37

Kuzuoka, Hideaki, Shinya Oyama, Keiichi Yamazaki, Kenji Suzuki, and Mamoru Mitsuishi. (2000). GestureMan: a mobile robot that embodies a remote instructor's actions. *Proceedings of the ACM conference on Computer Supprted Cooperative Work*. New York: ACM Press, pp. 155–162. doi:10.1145/358916.358986

Lanir, Joel, Ran Stone, Benjamin Cohen, and Pavel Gurevich (2013). Ownership and Control of Point of View in Remote Assistance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI'13*. New York: ACM Press, pp. 2243–2252. doi:10.1145/2470654.2481309

Linder, Natan, and Pattie Maes. (2010). LuminAR: Portable Robotic Augmented Reality Interface Design and Prototype. In *Adjunct proceedings of the 23nd annual ACM symposium on User interface software and technology*. New York: ACM Press, pp. 395–396. doi:10.1145/1866218.1866237

Machino, Tamotsu, Satoshi Iwaki, Hiroaki Kawata, Yoshimasa Yanagihara, Yoshito Nanjo, and Ken-ichiro Shimokura (2006). Remote-collaboration system using mobile robot with camera and projector. In *Proceedings - IEEE International Conference on Robotics and Automation 2006*. Washington DC: IEEE Computer Society, pp. 4063–4068. doi:10.1109/ROBOT.2006.1642326

Majumder, Aditi, Zhu He, Herman Towles, and Greg Welch. (2000). Achieving color uniformity across multi-projector displays. *Proceedings Visualization 2000. VIS 2000.* Washington DC: IEEE Computer Society, pp 117–124. doi:10.1109/VISUAL.2000.885684

Medioni, Gerard, and Ramakant Nevatia (1985). Segment-based stereo matching. *Computer Vision, Graphics, and Image Processing*. vol 31, no 1., pp. 2–18. doi:10.1016/S0734-189X(85)80073-6

Mistry, Pranav, Pattie Maes, and Liyan Chang (2009). WUW - Wear Ur World: A Wearable Gestural Interface. In *CHI'09 Extended Abstracts on Human Factors in Computing Systems*. New York: ACM Press, pp. 4111–4116. doi:10.1145/1520340.1520626

Motokawa, Yoichi, and Hideo Saito (2007). Support system for guitar playing using augmented reality display. In *Proceedings - ISMAR 2006: Fifth IEEE and ACM International Symposium on Mixed and Augmented Reality*. Washington DC: IEEE Computer Society, pp. 243–244. doi:10.1109/ISMAR.2006.297825

Murphy, Robin Roberson (2004). Human-robot interaction in rescue robotics. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*. vol 34. no. 2, pp 138–153. doi:10.1109/TSMCC.2004.826267

Neumann, Ulrich, and Anthony Majoros (1998). Cognitive, performance, and systems issues for augmented reality applications in manufacturing and maintenance. *Proceedings. IEEE 1998 Virtual Reality Annual International Symposium*. Washington DC: IEEE Computer Society, pp. 4–11. doi:10.1109/VRAIS.1998.658416

O'Neill, Jacki, Stefania Castellani, Frederic Roulland, Nicolas Hairon, Cornell Juliano, and Liwei Dai (2011). From ethnographic study to mixed reality: a remote collaborative troubleshooting system. In *Proceedings of the ACM 2011 conference on Computer supported cooperative work*. New York: ACM Press, pp. 225–234. doi:10.1145/1958824.1958859

Ou, Jiazhi, Susan R. Fussell, Xilin Chen, Leslie D. Setlock, and Jie Yang (2003). Gestural communication over video stream: supporting multimodal interaction for remote collaborative physical tasks. In *Proceedings of the 5th international conference on Multimodal interfaces*. New York: ACM Press, pp. 242–249.

Ranjan, Abhishek, Jeremy P. Birnholtz, and Ravin Balakrishnan (2007). Dynamic shared visual spaces: experimenting with automatic camera control in a remote repair task. In *Proceedings of ACM CHI 2007 Conference on Human Factors in Computing Systems*. New York: ACM Press, pp. 1177–1186. doi:10.1145/1240624.1240802

Raskar, Ramesh, Michael S. Brown, Ruigang Yang, Wei-Chao Chen, Greg Welch, Herman Towles, Brent Scales, and Henry Fuchs (1999). Multi-projector displays using camera-based registration. *In Visualization'99. Proceedings.* Washington DC: IEEE Computer Society, pp. 161–522.

Sheridan, Thomas B. (2002). *Humans and Automation: System Design and Research Issues.* New York: John Wiley & Sons.

Sodhi, Rajinder, Hrvoje Benko, and Andrew Wilson (2012). Lightguide: projected visualizations for hand movement guidance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York: ACM Press, pp. 179–188. doi:10.1145/2207676.2207702

Tang, Anthony, Carman Neustaedter, and Saul Greenberg (2006). VideoArms: Embodiments for Mixed Presence Groupware. In *People and Computers XX - Engage*. London: Springer, pp. 85–102. doi:10.1007/978-1-84628-664-3_8

Tecchia, Franco, Leila Alem, and Weidong Huang (2012). 3D helping hands: a gesture based MR system for remote collaboration. *Proceedings of the 11th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry - VRCAI'12*. New York: ACM Press, pp. 323–328. doi:10.1145/2407516.2407590

Tsai, Roger Y., and Reimar K. Lenz. (1988). Real time versatile robotics hand/eye calibration using 3D machine vision. In *Robotics and Automation, 1988 Proceedings., IEEE International Conference on.* Washington DC: IEEE Computer Society, pp. 554–561.

Wellner, Pierre (1993). Interacting with paper on the DigitalDesk. *Communications of the ACM*. vol. 36, no. 7, pp. 87–96. doi:10.1145/159544.159630

Woods, David D., James Tittle, Magnus Feil, and Axel Roesler (2004). Envisioning human-robot coordination in future operations. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 34, no. 2, pp. 210–218. doi:10.1109/TSMCC.2004.826272

Zhang, Zhengyou, Rachid Deriche, Olivier Faugeras, and Quang-Tuan Luong (1995). A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial intelligence*, vol. 78, no 1, pp. 87–119.