Integrating the Physical Environment into Mobile Remote Collaboration

Steffen Gauglitz **Cha Lee** **Matthew Turk** **Tobias Höllerer**
Department of Computer Science, University of California, Santa Barbara
{sgauglitz, chalee21, mturk, holl}@cs.ucsb.edu

ABSTRACT

We describe a framework and prototype implementation for unobtrusive mobile remote collaboration on tasks that involve the physical environment. Our system uses the Augmented Reality paradigm and model-free, markerless visual tracking to facilitate decoupled, live updated views of the environment and world-stabilized annotations while supporting a moving camera and unknown, unprepared environments. In order to evaluate our concept and prototype, we conducted a user study with 48 participants in which a remote expert instructed a local user to operate a mock-up airplane cockpit. Users performed significantly better with our prototype (40.8 tasks completed on average) as well as with static annotations (37.3) than without annotations (28.9). 79% of the users preferred our prototype despite noticeably imperfect tracking.

Author Keywords

telecollaboration; video-mediated communication;
Augmented Reality; markerless visual tracking; user study

ACM Classification Keywords

H.5.2 Information Interf. and Presentation: User Interfaces

INTRODUCTION

With the widespread deployment of fast data connections and availability of a variety of sensors for different modalities, the potential of remote collaboration has greatly increased. While the now ubiquitous video conferencing applications take advantage of some of these capabilities, the use of video between remote and local users is limited largely to watching disjoint video feeds, leaving much to be desired regarding direct interaction with the remote environment. Thus, teleconference-like applications have been largely successful when the matter at hand can be discussed verbally or with the help of purely digital data (such as presentations slides), but they hit severe limitations when real-world objects or environments are involved.

Gesturing and pointing are very natural and effective parts of human communication, without which communication can be ineffective and frustrating ("If I could just point to it, its right there!" [8]). To incorporate these means of communication, researchers have explored various ways to support remote pointing. However, previous studies use indirect [8] or counter-intuitive [3] means to display the pointers, or require specialized hardware [11, 21] or prepared environments [7]. Most notably, most of these systems assume a static camera as otherwise the pointers lose their referents.

Advances in computer vision facilitate applications which are to some degree able to understand where mobile cameras are pointed and what is being seen. These new capabilities should be exploited to enable the remote user to interact with what she[1] sees, instead of forcing her to passively watch whatever is viewed by the local user's camera. Further, we suggest that mobile Augmented Reality (AR) provides a natural and user-friendly paradigm to communicate spatial information about the scene and to browse an environment remotely. While many enabling technologies for AR still have deficiencies (such as limited robustness of visual tracking), we show that visual tracking has advanced to the point that it can significantly improve telecollaboration systems.

The contributions presented in this paper fall into three areas:

- We describe a system framework for unobtrusive mobile telecollaboration that integrates the physical environment. Our framework is compatible with a wide range of hardware configurations, including systems that are already ubiquitous (e.g., smartphones) as well as more advanced immersive systems.
- We designed a prototype that implements this framework and features several novel interface elements for unobtrusive mobile telecollaboration in unprepared environments. Our system uses *model-free, markerless, expanding visual tracking and modeling* to enable a remote user to provide visual/spatial feedback by means of *world-stabilized annotations* that are displayed to a local user with AR. Further, our system enables *decoupling* of the local user's view from the remote user's view while maintaining live updates. This gives the remote user some control over her viewpoint as well as allowing her to point at objects not currently in the field of view of the local user.
- We conducted a user study with 48 participants to evaluate the benefits of our prototype interface over an interface

[1] For ease of readability, we will assume a *female* remote user and a *male* local user in this text.

without annotations as well as an interface with static annotations, and we discuss both quantitative results and qualitative observations in detail. To our knowledge, our study is among the first formal user studies overall to purely rely on markerless, model-free visual tracking.

RELATED WORK

Video-mediated communication of remote collaborators has been studied in detail, including various video configurations [10, 20] and support for pointing or gesturing [8, 18].

Further research has been done on increasing the level of immersion of teleconferences [22, 30, 33], for example, by transferring live three-dimensional or perspectively corrected imagery of the participants. These setups typically require static hardware installations. Collaboration is possible on purely virtual data [22, 30, 33], but spatial references to the physical world are not supported.

There are several AR frameworks that focus on collaborative work and more mobile infrastructure [4, 5, 6, 31]. However, they also facilitate collaboration on virtual data only and have not been able to reduce the gap between virtual data and the remote physical environment. On the other hand, "VideoDraw" [34] and the "DoubleDigitalDesk" [37] are noteworthy early examples of integration of the remote physical space, but are limited to a static setup.

While many of the above systems focus on symmetric setups (i.e., both participants have the same equipment and share their own environment to the same degree), further work has been done on asymmetric local worker/remote expert scenarios [1, 3, 7, 8, 18, 21, 29]. These systems tend to focus on tasks involving objects in the local worker's physical environment ("collaborative physical tasks" [18]). Various ways to support the visual/spatial referencing have been studied, for example remote pointers or markers [3, 7, 8], laserpointers [21], the ability to draw onto the video stream [17, 29], or directly transferring videos of hand gestures [1, 17, 18].

With respect to the use of remotely controlled pointers or markers in a user study and the local worker/remote expert scenario, our work is most closely related to the studies by Fussell et al. [8], Bauer et al. [3], and Chastine et al. [7]. In [8], the local user had to assemble a toy robot with guidance from the remote user. The remote user saw the local user's workspace by means of a static camera looking over the local user's shoulder, while the local user saw the same view on a separate monitor in front of him. The remote user controlled a cursor which was visible on both screens.

In [3], the local user was wearing a video-see-through head-worn display (HWD) and had to solve a puzzle-like task. The remote expert saw the video and controlled a pointer visible to both of them. To be able to accurately reference objects despite movement of the local camera, the remote expert could freeze the video. However, they did not employ any kind of tracking and thus had to freeze the local worker's view simultaneously to be able to display the pointer. This appears to be a suboptimal user interface especially for an HWD, where the user expects the video to respond to his head movement.

In [7], the local user was asked to build a structure of wooden blocks, for which the remote user sees a virtual model in AR. Fiducial marker-based tracking was used to establish a shared coordinate system for both the virtual model (on the remote user's side) and the physical model (to be built on the local user's side), and the remote user could place pointers by placing additional fiducial markers around the virtual object. (Note that here, the virtual model serves not only as "expert knowledge," but additionally as surrogate for the physical model when placing the markers around it.)

However, all of these systems either assume a static camera, since otherwise virtual annotations lose their referents [1, 3, 8, 11, 17, 18, 21], or require extensive equipment and prepared environments to track and thus maintain the annotations' locations [7]. Furthermore, in all of these systems, the remote user's view into the local environment is either restricted to a static camera [8, 17] or tightly coupled to the local user's head or body movement [1, 3, 7, 21], thus forcing the remote user to constantly re-orient and ask the local user to hold still (or, in the case of [3], enforcing this by freezing both users' views) while pointing at an object.

In contrast, our system is able to support a *moving camera* by leveraging visual tracking. Annotations are displayed in AR, which provides a very natural alternative to Fussell et al.'s indirect third person view. Further, tracking the camera allows to *decouple* the views of local and remote user, thus allowing for accurate pointing and correct "anchoring" of annotations while avoiding Bauer et al.'s frozen HWD view.

The idea of using model-free, markerless visual tracking for the purpose of remote collaboration has previously been conceptualized by Lee and Höllerer [25] and Ladikos et al. [24].

One alternative to a wearable setup is to use a camera and projective device mounted to a robot which is controlled by the remote user [11, 23]. However, this requires specialized hardware which needs to be carried around and put in place, and the range and speed of operation are limited by the robot.

Our work is further related to studies of pointing [16] and collaboration [26] in virtual environments, as well as support for pointing in groupware applications [13, 15], although these systems do not reference the physical environment.

A FRAMEWORK FOR MOBILE TELECOLLABORATION THAT INTEGRATES THE PHYSICAL ENVIRONMENT

Fig. 1 describes the proposed framework which enables the remote user to explore a physical environment by means of live imagery from a camera that the local user holds or wears. The remote user is able to interact with the model fused from these images by creating virtual annotations in it or transferring live imagery (e.g., of gestures) back.

The crucial and, with respect to live mobile telecollaboration systems, novel component of this framework is the tracking and modeling core, which enables the system to (1) synthesize novel views of the environment and thus decouple the remote user's viewpoint from that of the local user, giving her some control over her viewpoint, and (2) correctly register virtual annotations to their real world referents.

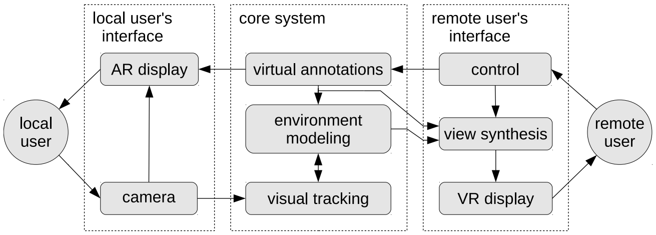


Figure 1. Overview of the proposed framework for unobtrusive, mobile telecollaboration that includes the physical environment. This figure depicts the situation that the user on the left is situated in the physical task environment and thus assumes the role of the local user, while the user on the right assumes the role of the remote user.

This framework is compatible with hardware systems that are already ubiquitous (e.g., smartphones), but scales to more advanced high-end systems as well. In particular, it is compatible with various types of displays for the local user, including projector-based setups [11].

Local User's Interface

The local user is assumed to hold or wear a device that integrates a camera and a display system (e.g., hand-held tablet or HWD with camera), which is used to both sense the environment and display visual/spatial feedback from the remote user correctly registered to the real world. In the case of a hand-held device, it acts as a *magic lens* (i.e., showing the live camera feed plus virtual annotations). Since a collaboration system has to aid the user in his or her actual task rather than distract from it, an interface which is simple and easy to comprehend is essential. It must facilitate an active user who may be looking at and working in multiple areas.

Remote User's Interface

The remote user is presented with a view into the local user's environment, rendered from images obtained by the local user's camera. The remote user can place annotations that will be displayed to both users, correctly registered to their real-world referents from their respective point of views. Annotations may include point-based markers, more complex three-dimensional annotations, drawings, or live imagery (e.g., of hand gestures).

In the simplest case, the remote user's viewpoint may be restricted to being identical to the local user's current camera view. In this case, no further image synthesis is needed. Ideally, however, the remote user should be able to decouple her viewpoint and control it independently, as far as supported by the available imagery of the environment.

If the system allows for decoupled views, it is important that only the *viewpoint* is decoupled; the video is still synthesized and updated from live images in order to enable consistent communication. (In the case of our prototype system, the effect of this can be observed in Fig. 2(c): although the viewpoints are different, the remote user sees how the local user is pointing to a control element on the panel in front of him.)

Visual Tracking & Environment Modeling

We assume that the environment may be completely unknown prior to the start of the system, that is, we do not require any model information. While using model information bears the potential to make the system more robust, any kind of model information has to be collected in some way prior to the task, which either severely limits the generality of the system or puts a burden on the user.

Instead, we assume that the system starts with no prior knowledge about the scene and builds up an internal representation on the fly, which automatically expands to include new areas as the camera moves. Our framework is compatible with environment modeling systems of different levels of flexibility and generality, including panorama mappers [35] and Simultaneous Localization and Mapping (SLAM) [19]. If admissible for the application, other sensors such as active depth cameras could also be used [28].

PROTOTYPE IMPLEMENTATION

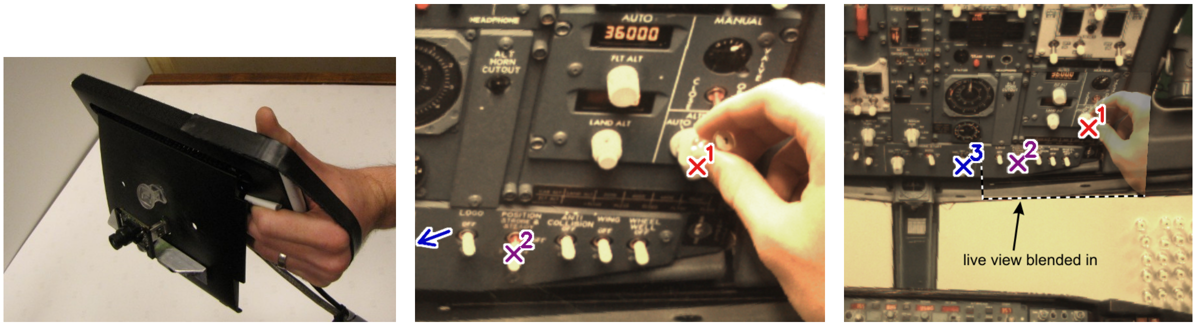
We designed a prototype system which implements the framework described in the previous section. Our prototype is fully functional, but in several ways the simplest or least immersive implementation of our framework. With the user study discussed in the next section, we show that it nevertheless provides significant benefits in a telecollaboration task.

Our current prototype is limited in generality in one particular aspect: its tracking and modeling capabilities are restricted to homographic warps, that is, it is compatible with planar scenes or rotation-only movements, but not with general camera motion in general environments. (While state-of-the-art SLAM-like systems [19, 27] have demonstrated impressive robustness, they still require the user to move the camera in a certain way especially during initialization. We thus consciously decided to not use a SLAM system for this study.) This is flexible enough for our study setup and serves as proof of concept for both the framework and the interfaces described in the following. We plan to address this restriction in future work.

User Interfaces

As hardware interface for the local user, we decided to use a hand-held tablet screen, since we wanted to show that benefits can be achieved using hardware setups that are already in widespread deployment. In the current prototype, for ease of implementation and flexibility with respect to the hardware components to be used, we use a USB-driven screen with a camera mounted on the back (Fig. 2(a)) instead of a stand-alone tablet computer, and all computations are executed on the connected PC. However, we do not make use of a GPU and the computation-intensive part (i.e., the tracking with the template matching core, cf. details below) is very similar to the tracking system by Wagner et al. [36], which operates in real time on a 2009 smart phone. Thus, our system could be implemented on a mobile device (such as a light-weight tablet or smartphone) given appropriate optimizations.

When tracking is lost and needs to be recovered, the local user sees a large red 'X' across the screen to indicate that tracking is lost and hence virtual annotations cannot be drawn. By pointing back to a previously seen location, the user can help the system to recover tracking (cf. section on recovery below).



(a) hand-held device for local user (b) view for local user (c) view for remote user

Figure 2. (a) Hand-held device for local user: 10" USB tablet screen with camera mounted on back; (b,c) view for local and remote user. The remote user's viewpoint is frozen, but the live video frame is – correctly registered with the frozen frame – blended in, such that the remote user can still observe the local user's actions. The remote user has set three markers in her view. Two are inside the local user's current field of view, the third lies outside his view on the left, as indicated by the (accordingly colored) arrow on the left. Note that the visual tracking works fine (as apparent from the correctly registered frames and markers) despite significant occlusion from the local user's hand.

Even if tracking is lost, the live video feed is not interrupted and the local user may continue to work.

As virtual annotations, our prototype supports point-based markers which the remote user controls. They are displayed to both users in their respective views as an 'X' anchored to the real world, with a number attached to it and additionally color coded to disambiguate between multiple markers. When a marker is set outside the current view or moves outside the current view, a correspondingly colored arrow appears on the border of the screen pointing towards the marker's location (see Fig. 2(b)). (Cf. [2, 12] for other visualizations of off-screen objects.)

The remote user is presented with a view of the local user's environment and can place markers by clicking into this view (Fig. 2(c)).

We provided one particular feature that allows control over the remote user's viewpoint: the remote user can "freeze" (and un-freeze) her viewpoint at any time. Despite its simplicity, this feature allows decoupling of the remote view from local movement, and thus enables for example precise clicking on objects despite the movement of the camera on the local user's side. (Cf. Güven et al.'s "Frame & Freeze" [14]; the difference is that they developed their technique for a mobile user to interact with his own view.)

In contrast to the setup by Bauer et al. [3], the local user's screen is *not* affected by this and remains "live" at all times. Using visual tracking, the remote user's view is still registered to the local user's live view. Therefore, when markers are set, they immediately appear at the correct position with respect to the world on both the local and remote views. The remote user can return to the live view at any time by right-clicking again, and the view "zooms back" to the live view, animated by interpolating a few frames between the two viewpoints.

Note that not the remote user's *frame* is frozen — this would have the crucial disadvantage that the remote user would not receive any visual updates and could not observe the local user's action. Instead, we only freeze the *viewpoint* of the remote user and display a transparent image of the live stream on top of the remote user's frozen view, correctly registered with the frozen viewpoint. This creates the effect that, when the local user points to something within his camera's field of view, the remote user sees a half-transparent hand correctly indicating the object of interest. Blending the two frames rather than displaying only the warped live frame has the advantage that the initial frozen frame remains visible and stable even if the warped view becomes jittery or blurry (due to jittery tracking, motion blur, or extreme warping angles), and it reduces artifacts along the border of the live frame. The blended view with the local user's half-transparent hand (Fig. 2(c)) bears noteworthy resemblance to the blended video feeds in [17, 34, 37], but these systems require a static setup with known camera pose.

Visual Tracking & Environment Modeling

We used a multi-level, active search patch tracker with normalized cross-correlation (NCC)-based template matching and keyframe-based recovery, inspired by the systems of Wagner et al. [36] and Klein and Murray [19]. In preliminary investigations, this algorithm was found to perform favorably in terms of speed/robustness trade-off compared to several image alignment- and other feature-based algorithms.

As postulated by our framework, we do not use any model information; that is, the environment is completely unknown prior to the start of the system. Instead, our system builds up an internal representation on the fly which automatically expands to include new areas as the camera moves.

The tracking system could enable further features that may improve the collaboration. Most notably, it could effectively increase the field of view of the remote user by displaying the entire map collected by a panning camera. However, we decided to make only minimal use of the tracking by only world-referencing annotations and the remote user's viewpoint, to allow evaluation of these features in particular.

Details of the Tracking Algorithm

From the first frame, the tracker creates an image pyramid by half-sampling the image twice. On each level, keypoints are detected with the FAST corner detector [32], and a subset that is spatially well-distributed across the image is selected as described in [9]. Those keypoints constitute the features that are tracked and based on which the camera viewpoint (modeled as homography) is estimated for each frame.

Each incoming frame is projected back to the initial viewpoint, taking the previous frame's estimated homography as prior estimate (such that the patch tracker has to be robust to the distortion between two subsequent frames only). Then, each feature's new position is established by NCC-based template matching of an 11×11 pixel image patch. This is done for the top-most (smallest) image level first, then the homography is re-estimated from the putative feature correspondences using RANSAC and used to project the features into the next level. This ensures robustness to relatively large inter-frame movements despite a small template matching area. We used a radius of 5 pixels on the top-most level, resulting in tolerable movements of 20 pixels between two frames.

As new areas come into view, features are detected in those areas as well and added to the internal map by storing their position and NCC template with respect to the initial frame's coordinate system.

Tracking Loss & Recovery

We implemented a recovery algorithm similar to that of Klein and Murray [19]. In certain intervals and if tracking quality is deemed good (as determined by the fraction of inliers found by RANSAC), the system creates a keyframe by downsampling the current frame to 80×60 pixels, blurring it with a Gaussian kernel ($\sigma = 1.5$), and storing it together with the current pose information.

The system declares tracking to be lost if RANSAC fails to find a certain fraction of visible features (25%) that agree on one pose estimate. If tracking is lost, each new frame is also downsampled and blurred, and the NCC score between this frame and all stored keyframes is computed. The keyframe with the highest score is then aligned to the downsampled current frame. During this image alignment step, the homography is restricted to be affine, which increases both the speed and the convergence rate. The refined pose of the stored keyframe is then fed back into the tracking algorithm. If RANSAC is able to again find a sufficiently large fraction of inliers among the feature correspondences, tracking is assumed to be restored successfully; otherwise, the recovery algorithm is run again with the next frame.

When It Is Not Necessary to Recover Tracking

Unless tracking recovery succeeds quasi-instantaneously and thus fully automatically, the user has to help out by pointing towards a previously seen location. This is a distraction from his actual task, and thus should only be done if necessary. Therefore, it is important to understand when recovery is dispensable and design the system appropriately.

In our system, tracking is needed in two cases: when annotations (markers) are present, and when the remote user has frozen her viewpoint. If neither is the case, tracking is not currently needed and hence the user should not be bothered with requests to recover it. In this case, recovery is attempted only for a maximum of 10 frames. If unsuccessful, tracking is simply reset and re-initialized with the current frame. If, however, tracking is needed to maintain currently active markers or the registration with the remote user's frozen viewpoint, recovery is attempted for much longer, asking the user to help with recovery by displaying the red 'X' if needed. If recovery is not successful after 240 frames, it is assumed that the user does not want or is unable to recover tracking, and the tracking is reset.

local user remote user	female female	female male	male female	male male
# of teams	6	4	5	9

Table 1. Gender distribution and participant teams.

Real-time Performance

Overall, this system is fast enough to operate in real time — the framerate is limited by the camera (30 Hz) and not by the tracking algorithms — and robust enough to be used by our study participants without any specific instructions, while coping with (or successfully recovering from) free camera movement, motion blur, specular reflections, and significant occlusion (e.g., due to the user's hand, cf. Figs. 2(c) and 4(b)).

EVALUATING THE EFFECTIVENESS OF LIVE ANNOTATIONS IN A COLLABORATIVE TASK

To evaluate the effectiveness of world-stabilized markers, we designed a user study comparing our interface using world-stabilized markers against one interface without any annotations and one interface with static markers. The study's scenario was that of a remote expert instructing and directing a novice local user in operating an airplane.

Various parameters of the study as reported below were refined during several pilot study trials with a total of 12 users.

Participants

We had a total of 48 participants in the main study, 18 to 39 years old (average 23.3), 27 male and 21 female, who worked together on 24 teams as detailed in Table 1.

All participants had normal or corrected vision. 40 reported they did not know the other participant, 4 had "met before," and 4 "knew each other well." 93% stated they had at least 10 years of experience speaking English. Each user was compensated for their time commitment of about one hour with a nominal amount of US $10. We had two further participant teams whose data we did not include in the analysis. In one team, one individual had a form of color-blindness; in the other team, one participant did not adequately follow the study administrator's instructions during the training period.

Physical Setup

We created a mock-up airplane cockpit by printing a high-resolution image of the interior of an airplane cockpit (Fig. 3) on 3'×4' paper and mounting it to a metal panel on the wall. To simulate a remote user in another location, we placed a room divider next to the poster and placed the remote user's

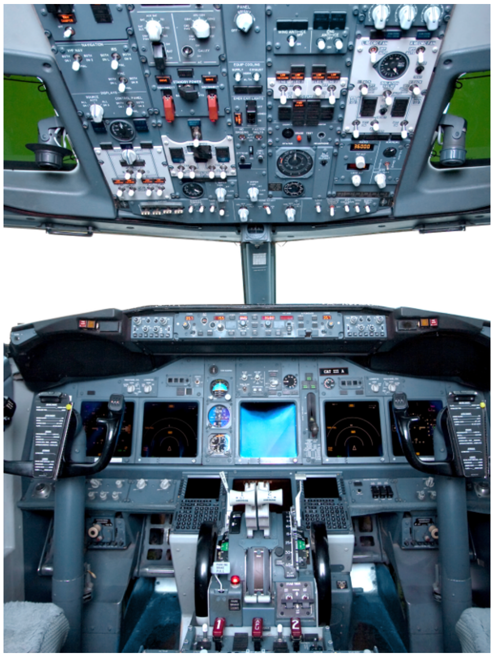


Figure 3. Our testbed: view of a Boeing 737 cockpit. This image was printed in size 3'×4' and mounted to a metal panel on the wall. The resolution is high enough that most of the control element labels are readable. Original image file obtained from iStockphoto.com/Smaglov.

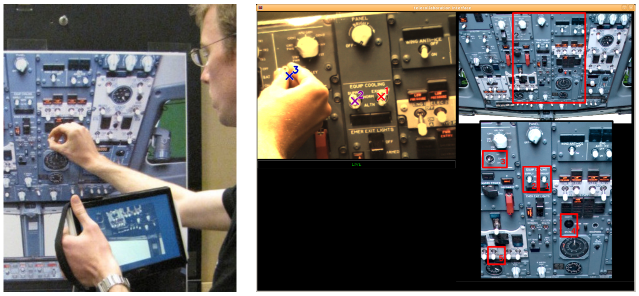
station on the other side. This allowed both users to communicate verbally by simply talking out loud while blocking any direct visual communication.

The device for the local user consisted of a MIMO 10.1" "iMo Monster" Touchscreen (used as display only, touch disabled) with a Point Grey Firefly (34° horizontal field of view) mounted on the back to deliver 640×480 Bayer images at 30 Hz. To make the tablet more comfortable to hold, we added some rubber padding on both sides of the tablet and a strap to go around the user's hand (cf. Fig. 2(a)).

The remote participant used a standard desktop PC interface with monitor, keyboard, and mouse. The system ran on a standard PC with an Intel i7 Core with 4 GB RAM running Ubuntu 10.10. All interfaces and further system components were implemented in C++, making use of the OpenCV and libCVD libraries for the vision system.

Task

The task that each pair of participants performed was to identify and "operate" a series of control elements in our mockup cockpit, in order to, e.g., "safely land the airplane." The local user was assumed to be a novice, not knowing which elements to operate, and had to be instructed by the remote expert. "Operating" an element was simulated by placing a magnetic pin onto it. The pins were numbered so that the study administrator could later verify the correct placement and order. We chose the testbed such that it would neither be too easy nor too difficult to reference the control elements verbally. Many of the elements had readable labels, visually distinguishable features, and/or could be described by their relative position in the cockpit. However, labels were relatively small, and some switches were in a series of alike-looking elements. It would be easy to design a testbed that would be much easier or much harder, but through pilot studies we arrived at this setup as a reasonable and, in particular, realistic compromise between the two extremes.



(a) local user

(b) remote user's interface

Figure 4. (a) Local user with tablet, putting a magnetic pin on one of the "buttons." (b) Overview of the remote user's interface with live video view on the top left and "expert knowledge" information (consisting of overview image on the top and detail on the bottom) on the right.

One participant played the role of the local user. This participant stood in front of the poster with the display device (as seen in Fig. 4(a)) and was instructed by the remote user as to which control elements to "operate," i.e., place a pin onto.

The other participant played the role of the remote expert and was responsible for directing the local user as to which control element to operate. The remote user sat in front of a desktop monitor which showed the camera feed from the local user on the left side. On the right side, the remote user saw a detail of the cockpit with a sequence of buttons clearly marked and, above, an overview image in which the location of the detail was indicated (Fig. 4(b)). These images simulated the "expert knowledge" that the remote user was assumed to possess. The remote user then communicated the locations of each element to the local user verbally and via the interface functions. In addition to directing the local user, she was asked to monitor correct pin placement. (Users were allowed to remove and re-place incorrectly placed pins, and the remote user was instructed to inform the local user accordingly if she observed an error.)

Each page of the expert knowledge information indicated a sequence of five random buttons. When completed, the remote user could press a key to proceed to the next page. After five pages (25 buttons), we introduced a brief break in which the study administrator would take down the pins, comparing them with a control sheet and noting down errors (if any).

Conditions

With all interfaces, the participants were able to talk to each other without restriction.

Interface A: video only

In this interface, the live video from the tablet's camera is streamed to the remote user's screen (i.e., one-way video), but the remote user does not have any means of providing visual feedback. This is similar to using a current tablet PC with rear-facing camera and standard video conferencing software.

Interface B: video + static markers

In this interface, the remote user can click into the video view to create a marker — a colored 'X' — that is visible on both screens (the remote user's screen as well as the local user's tablet screen). However, the marker is static within image coordinates, so it appears to "swim away" from its original position if the tablet's camera is moved. The remote user can set up to five of those markers, and clear them by pressing the space bar at any time. By pressing a number key between 1 and 5, the user can select the next marker to be set; without pressing number keys, the system rotates through markers 1 to 5. The next marker to be set is indicated by a small white number next to the mouse cursor. This condition is similar to the pointing in [8] in that it assumes a stationary camera.

Interface C: video + world-stabilized markers

This interface is using our prototype system as described in the previous section. As with interface B, the remote user can set up to five markers by clicking into the video view (space bar to clear, number keys to select marker as in B). However, now the markers are stored in world-stabilized coordinates and thus stick to their original positions despite movement of the camera. With a right-click, the remote user can freeze her viewpoint as described in the previous section.

Experimental Design

We used a within-subjects design with a single independent variable (interface type A, B, C) and a single dependent variable (total number of tasks completed). We also recorded the number of errors. The order of the interfaces was completely balanced for all six possible orderings, with each possible ordering traversed by four teams, and the tasks were randomly selected for each interface from a set of tasks.

Our hypotheses about the study's outcome were as follows:

H1 Users will complete more tasks with both interface B and C than with interface A.

H2 Users will complete more tasks with C than with B.

H3 Users will prefer interface C over both A and B.

H4 Users will feel more confident about their task performance with interface C.

Procedure

At the beginning of the study, each participant was given a color blind test and filled out a pre-study questionnaire with demographic and background information. The study administrator then verbally explained the scenario and the roles each participant would play. Next, the local user was given the hand-held display and adjustments to the strap were made so that the user was comfortable holding the device in one hand.

For each interface, the administrator explained the respective interface. The administrator was careful to only explain the individual features and to not recommend any particular strategy; instead, the users were explicitly told that it was up to them to determine how exactly they would make use of the features. Next, the users were given a training session of five minutes to get accustomed to the interface and develop a strategy for using it. During this training session, the administrator would correct any mistakes made by either user, and would also encourage the users to try out the different features if they had not already done so. After the training session, a measured session of seven minutes was started. Before starting this session, the administrator gave the stipulation that selecting the correct elements in the correct order was important and that participants should confirm with each other if in doubt, but that within that, users should work as fast as possible.

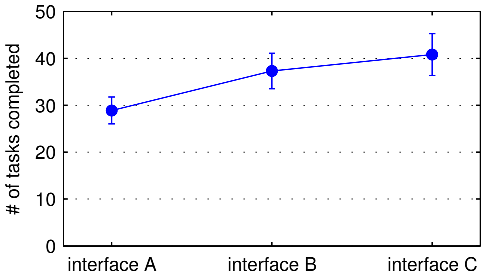


Figure 5. Main quantitative result: number of tasks completed as a function of the three different interfaces, shown are mean and 95% confidence intervals. Users were significantly faster with B than with A, and significantly faster with C than with A (cf. Table 2).

After each interface, users filled out a brief questionnaire asking about their experience with the interface and any physical discomfort they may have felt. While some local users indicated that their arms got tired, all of them explicitly confirmed verbally that they did not have any concerns about continuing the study.

Finally, the participants were asked to fill out a post-study questionnaire, asking them to compare the interfaces and to note any further comments on the study.

RESULTS

Task Performance

The stipulation to not make mistakes and confirm with each other when in doubt worked very well: 16 out of 24 teams completed all three trials without a single error, and only two teams made more than one error in one trial (in which users would select around 20 to 60 buttons). The errors were spread evenly among the three interfaces.

Participants were able to complete the most tasks with Interface C, averaging 40.8 tasks. Participants averaged 37.3 tasks with Interface B and 28.9 tasks with Interface A (Fig. 5). Analyzing our results, Mauchly's test did not show a violation of sphericity against interface ($W(2) = 0.94$, $p = 0.49$). With a one-way repeated measures ANOVA, we found a significant effect of interface on the number of tasks completed with $F(2,46) = 23.45$, $P < 0.001$, and $\eta^2_{\text{partial}} = 0.50$. Using Tukey's posthoc analysis (Table 2), we found that users completed significantly more tasks with both interfaces B and C than with interface A, thus confirming hypothesis H1. We did not find a significant difference between B and C, hence hypothesis H2 could not be confirmed despite the slightly higher average number of completed tasks with interface C.

Interfaces	Difference	Lower	Upper	P Adj.	
B – A	8.42	4.08	12.75	< 0.001	*
C – A	11.92	7.58	16.25	< 0.001	*
C – B	3.50	-0.83	7.83	0.135	

Table 2. Tukey's posthoc analysis for all pairwise comparisons with a 95% family confidence interval. * indicates significant differences.

Questionnaires

In the post-study questionnaire, users were asked to rate and rank the three interfaces. The results are aggregated in Fig. 6. Overall, users clearly preferred interface C: 72% 'strongly agreed' to the statement "Interface C helped me to solve this task," compared to 8.5% for A and 23% for B; 68% 'strongly agreed' to the statement "Interface C made me feel confident that I was doing the task correctly," compared to 17% for A and 11% for B. When asked "Which of the interfaces did you like best/would you choose to use for a real task?," 79% selected interface C as their first choice. Many users further confirmed their preference with comments: "The interface with vision tracking was better and easier." — "I was very impressed with the tracking capabilities. The interface was very easy to understand" — "This interface is an order of magnitude better than the others" — "The tracked markers [were] much eas[ier] to give direction[s] with."

When mapped to a linear scale and analyzed with ANOVA, the ratings for interface C are significantly better than the ratings for both A and B for all of the above questions (in above order: F(2,90) = 32.7, F(2,90) = 14.6, F(2,90) = 32.6, with $P < 0.001$ for all). There were no significant differences in the answers to "I had difficulties using this interface."

These data confirm hypotheses H3 and H4.

Very few users seemed to be distracted by the additional features and the multimodality of the task or commented to that effect: "It was actually easier to just talk it [out] rather than worry about clicking the mouse which left me distracted."

DISCUSSION

Overall, most participants seemed very engaged with the task, and many explicitly commented that they liked the experience. With both interfaces B and C, most teams effectively used multimodal communication, by indicating control elements with markers as well as visually describing them at the same time. However, it was apparent that with interface C, many teams needed very little verbal communication, largely relying on the markers (however, users did add verbal descriptions for "difficult" markers or to confirm if needed). With interface B, more verbal communication and more corrections were needed ("Is it this one?" — "No, no, the one next to it!"). This is a possible explanation for why users felt more confident when using interface C (cf. Fig. 6).

Use of Features

Use of the interface features varied among participants. Almost all used markers for the overwhelming majority of cases; most used the freeze function, though frequency and strategy varied. Only a few people used the number keys to change the sequence of markers.

Task Performance with Interface C vs. Interface B

Given the overall very good user ratings of interface C and few usability problems, the question remains why the study did not show a significant difference between C and B in terms of the number of tasks completed. It is possible that for this task, both interfaces are equivalent. However, we observed indications for other possible explanations:

- One very frequently employed strategy with interface C was to first navigate the local user to the general area, then freeze, mark several buttons, then observe the local user's actions. If the local user started to put the pins down while the remote user would mark the next elements (i.e., they would work in parallel, for which the decoupling of views is essential), this strategy seemed to achieve the best performances, which could not be reached with the other interfaces. However, a few local users waited until all markers would be selected (sometimes encouraged to do so by the remote user, even though her view was frozen and thus not affected by his movement) before putting down pins, which negatively impacted the performance.

- Since the local user had to wait for instructions from the remote expert for each step, it was relatively easy (although, as some users commented, more strenuous) for him to hold the tablet still while she pointed out the elements to be marked. A task that required more autonomous work from the local user would likely show a greater benefit of the decoupling of views.

- Due to our use of a "magic lens" tablet, our AR experience was indirect. Some users explicitly commented on this: "I had to see where the X was on the screen then find it again in 'real life.' That transition added a non-trivial delay." Due to this indirection and since they could hold the tablet in one hand, putting down pins with the other, interfaces B and C are effectively more similar to each other than with a direct view (e.g., with an HWD or projective display).

- Loss of time due to tracking loss: although not perceived as a large problem by most users (61% of the users reported 'no' or 'at most a little' impact on task performance, another 29% 'some' impact), it is clear that occasional or more frequent tracking loss impacted the task performance. One noteworthy circumstance is that tracking got lost if users moved very close to the poster (due to fewer trackable features in the field of view and interpolation/sampling artifacts when matching across scales), so that users had to step back a little to recover tracking: "I found that when I [...] needed to 'zoom in' to physically put the push pins in the correct spots, the tracking marks sometimes did not show up [...] until I zoomed out again... this slowed things down a bit." Improving tracking for this condition in particular would be beneficial.

- Since interface C offers more functions, it is slightly more complex, which may have counteracted potential benefits. Although the data shows that in general, users did not find interface C more difficult to use (cf. Fig. 6), individual users commented that C might benefit from longer-term use: "I would utilize B [...]. However, with more time to

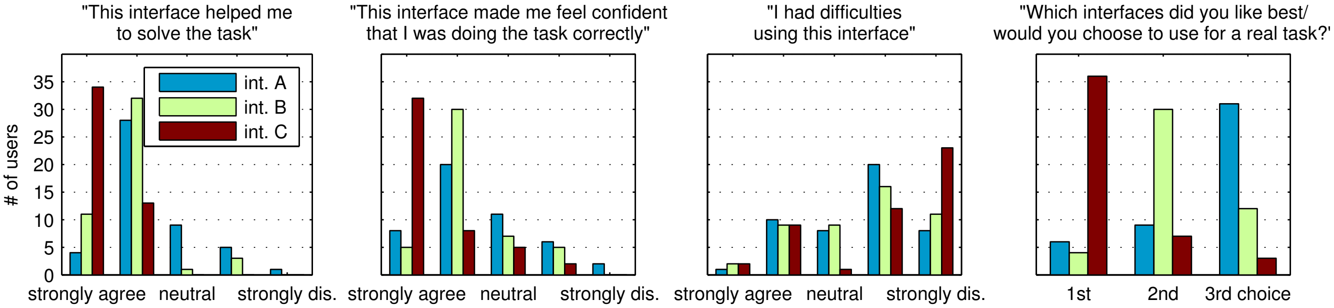


Figure 6. User responses from post-study questionnaire.

adapt, I probably would utilize C"; "I thought using the markers and the freeze was more difficult [...]. I think with another trial's worth of practice it would have less effect."

CONCLUSIONS AND FUTURE WORK

We described a system framework which uses computer vision and AR to facilitate better integration of the physical environment for unobtrusive mobile telecollaboration. We designed one particular prototype that implements this framework, featuring several novel interface elements. We then provided empirical evidence in the form of a user study that our framework can provide significant benefits for a remote collaborative task: Even though our current visual tracking is noticeably imperfect and our prototype is not the most immersive implementation of our framework, the task performance was significantly better compared to a video-only reference interface, and our prototype was much preferred by users over both compared interfaces.

We conclude that the proposed framework is promising and has the potential of significantly improving telecollaboration interfaces. We suggest that this requires neither futuristic, fully immersive hardware interfaces nor perfect tracking, but that current computer vision and AR technologies can be used effectively for collaboration without any special training.

The most obvious extension of our work is to remove the tracking and modeling system's restriction to homographic warps. A potential remedy is to integrate SLAM-like modeling [19, 27] or use active depth sensors [28].

Further, our prototype implementation does not exhaust the potential of the proposed framework: Guided by Fig. 1, several components may be substituted with more flexible, more powerful, or more immersive components. This includes more flexible viewpoint control for the remote user, support for more complex virtual annotations, and use of more immersive display systems. For example, one could use the feature-rich annotation tool and pico projector-based setup by Gurevich et al. [11] as interfaces in Fig. 1 (note that they mention the use of visual tracking for future work, thus arriving at a similar vision), or directly overlay hand gestures (as in [18, 29], but now for a moving camera). When using more complex annotation or gesture interfaces, existing work on the design of shared visual spaces and gesture support [10, 18, 20] will be very valuable. All of these aspects deserve further investigation and bear the potential to further increase the applicability of mobile interactive telecollaboration interfaces.

ACKNOWLEDGMENTS

We thank Andy Beall (Dept. of Psychology, UCSB) for helpful discussions and suggestions on the design of the study, Daniel Sheinson (Dept. of Statistics and Applied Probability, UCSB) for his help in conducting the statistical analysis, and our anonymous reviewers for several suggestions that helped to improve our manuscript. This work was partially supported by a fellowship from UCSB, NSF CAREER grant IIS-0747520 and ONR grant N00014-09-1-1113.

REFERENCES

1. Alem, L., Tecchia, F., and Huang, W. HandsOnVideo: Towards a gesture based mobile AR system for remote collaboration. In *Recent Trends of Mobile Collaborative Augmented Reality Systems*, L. Alem and W. Huang, Eds. Springer New York, 2011, 135–148.
2. Baudisch, P., and Rosenholtz, R. Halo: a technique for visualizing off-screen objects. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI)*, 2003, 481–488.
3. Bauer, M., Kortuem, G., and Segall, Z. ”Where are you pointing at?” a study of remote collaboration in a wearable videoconference system. In *Proc. Intl. Symp. on Wearable Computers (ISWC)*, 1999, 151 –158.
4. Billinghurst, M., Bowskill, J., Jessop, M., and Morphett, J. A wearable spatial conferencing space. In *Proc. Intl. Symp. on Wearable Computers (ISWC)*, 1998, 76 –83.
5. Billinghurst, M., Weghorst, S., and Furness, T. Shared space: An augmented reality approach for computer supported collaborative work. *Virtual Reality 3* (1998), 25–36.
6. Butz, A., Höllerer, T., Feiner, S., MacIntyre, B., and Beshers, C. Enveloping users and computers in a collaborative 3D augmented reality. In *Proc. IEEE/ACM Intl. Workshop on Augmented Reality*, 1999, 35–44.
7. Chastine, J., Nagel, K., Zhu, Y., and Hudachek-Buswell, M. Studies on the effectiveness of virtual pointers in collaborative augmented reality. In *Proc. IEEE Symp. on 3D User Interfaces (3DUI)*, 2008, 117 –124.
8. Fussell, S. R., Setlock, L. D., Yang, J., Ou, J., Mauer, E., and Kramer, A. D. I. Gestures over video streams to

support remote collaboration on physical tasks. *Hum.-Comput. Interact. 19* (Sept. 2004), 273–309.

9. Gauglitz, S., Foschini, L., Turk, M., and Höllerer, T. Efficiently selecting spatially distributed keypoints for visual tracking. In *Proc. IEEE Intl. Conf. on Image Processing (ICIP)*, 2011.
10. Gergle, D., Kraut, R., and Fussell, S. Action as language in a shared visual space. In *Proc. ACM Conf. on Computer Supported Cooperative Work*, 2004, 487–496.
11. Gurevich, P., Lanir, J., Cohen, B., and Stone, R. TeleAdvisor: a versatile augmented reality tool for remote assistance. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems*, 2012, 619–622.
12. Gustafson, S., Baudisch, P., Gutwin, C., and Irani, P. Wedge: clutter-free visualization of off-screen locations. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems*, 2008, 787–796.
13. Gutwin, C., and Penner, R. Improving interpretation of remote gestures with telepointer traces. In *Proc. ACM Conf. on Computer Supported Cooperative Work*, 2002, 49–57.
14. Güven, S., Feiner, S., and Oda, O. Mobile augmented reality interaction techniques for authoring situated media on-site. In *Proc. IEEE/ACM Intl. Symp. on Mixed and Augmented Reality (ISMAR)*, 2006, 235–236.
15. Hayne, S., Pendergast, M., and Greenberg, S. Implementing gesturing with cursors in group support systems. *J. Manag. Informat. Syst.* (1993), 43–61.
16. Hindmarsh, J., Fraser, M., Heath, C., Benford, S., and Greenhalgh, C. Fragmented interaction: establishing mutual orientation in virtual environments. In *Proc. ACM Conf. on Computer Supported Cooperative Work*, 1998, 217–226.
17. Kirk, D., and Stanton Fraser, D. Comparing remote gesture technologies for supporting collaborative physical tasks. In *Proc. SIGCHI Conf. on Human Factors in Computing Systems (CHI)*, 2006, 1191–1200.
18. Kirk, D. S. *Turn It This Way: Remote Gesturing in Video-Mediated Communication*. PhD thesis, University of Nottingham, 2006.
19. Klein, G., and Murray, D. Improving the agility of keyframe-based SLAM. In *Proc. European Conf. on Computer Vision (ECCV)*, 2008, 802–815.
20. Kraut, R., Miller, M., and Siegel, J. Collaboration in performance of physical tasks: Effects on outcomes and communication. In *Proc. ACM Conf. on Computer Supported Cooperative Work*, 1996, 57–66.
21. Kurata, T., Sakata, N., Kourogi, M., Kuzuoka, H., and Billinghurst, M. Remote collaboration using a shoulder-worn active camera/laser. In *Proc. Intl. Symp. on Wearable Computers (ISWC)*, 2004, 62–69.
22. Kurillo, G., Bajcsy, R., Nahrsted, K., and Kreylos, O. Immersive 3D environment for remote collaboration and training of physical activities. In *Proc. IEEE Virtual Reality*, 2008, 269 –270.
23. Kuzuoka, H., Oyama, S., Yamazaki, K., Suzuki, K., and Mitsuishi, M. Gestureman: a mobile robot that embodies a remote instructor's actions. In *Proc. ACM Conf. on Computer Supported Cooperative Work*, 2000, 155–162.
24. Ladikos, A., Benhimane, S., Appel, M., and Navab, N. Model-free markerless tracking for remote support in unknown environments. In *Proc. Intl. Conf. on Computer Vision Theory and Applications*, 2008.
25. Lee, T., and Höllerer, T. Viewpoint stabilization for live collaborative video augmentations. In *Proc. IEEE/ACM Intl. Symp. on Mixed and Augmented Reality*, 2006, 241–242.
26. Mortensen, J., Vinayagamoorthy, V., Slater, M., Steed, A., Lok, B., and Whitton, M. Collaboration in tele-immersive environments. In *Proc. Workshop on Virtual environments*, 2002, 93–101.
27. Newcombe, R., Lovegrove, S., and Davison, A. Dtam: Dense tracking and mapping in real-time. In *Proc. IEEE Intl. Conf. on Computer Vision*, 2011, 2320–2327.
28. Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., Kohli, P., Shotton, J., Hodges, S., and Fitzgibbon, A. KinectFusion: Real-time dense surface mapping and tracking. In *Proc. IEEE Intl. Symp. on Mixed and Augmented Reality*, 2011.
29. Ou, J., Fussell, S. R., Chen, X., Setlock, L. D., and Yang, J. Gestural communication over video stream: supporting multimodal interaction for remote collaborative physical tasks. In *Proc. Intl. Conf. on Multimodal Interfaces (ICMI)*, 2003, 242–249.
30. Regenbrecht, H., Lum, T., Kohler, P., Ott, C., Wagner, M., Wilke, W., and Mueller, E. Using augmented virtuality for remote collaboration. *Presence: Teleoper. Virtual Environ. 13* (July 2004), 338–354.
31. Reitmayr, G., and Schmalstieg, D. Mobile collaborative augmented reality. In *Proc. IEEE/ACM Intl. Symp. on Augmented Reality*, 2001, 114 –123.
32. Rosten, E., and Drummond, T. Machine learning for high-speed corner detection. In *Proc. IEEE European Conf. on Computer Vision*, 2006, 430–443.
33. Sadagic, A., Towles, H., Holden, L., Daniilidis, K., and Zeleznik, B. Tele-immersion portal: Towards an ultimate systhesis of Computer Graphics and Computer Vision Systems. In *Proc. Intl. Workshop on Presence*, 2001.
34. Tang, J. C., and Minneman, S. L. VideoDraw: a video interface for collaborative drawing. *ACM Trans. Inf. Syst. 9* (April 1991), 170–184.
35. Wagner, D., Mulloni, A., Langlotz, T., and Schmalstieg, D. Real-time panoramic mapping and tracking on mobile phones. In *Proc. IEEE Virtual Reality*, 2010.
36. Wagner, D., Schmalstieg, D., and Bischof, H. Multiple target detection and tracking with guaranteed framerates on mobile phones. In *Proc. IEEE Intl. Symp. on Mixed and Augmented Reality (ISMAR)*, 2009, 57–64.
37. Wellner, P., and Freeman, S. The DoubleDigitalDesk: Shared editing of paper documents. Tech. Rep. EPC-93-108, EuroPARC, 1993.