RealityReplay: Detecting and Replaying Temporal Changes In Situ Using Mixed Reality

HYUNSUNG CHO, Human-Computer Interaction Institute Carnegie Mellon University, USA
MATTHEW L. KOMAR, Human-Computer Interaction Institute Carnegie Mellon University, USA
DAVID LINDLBAUER, Human-Computer Interaction Institute Carnegie Mellon University, USA

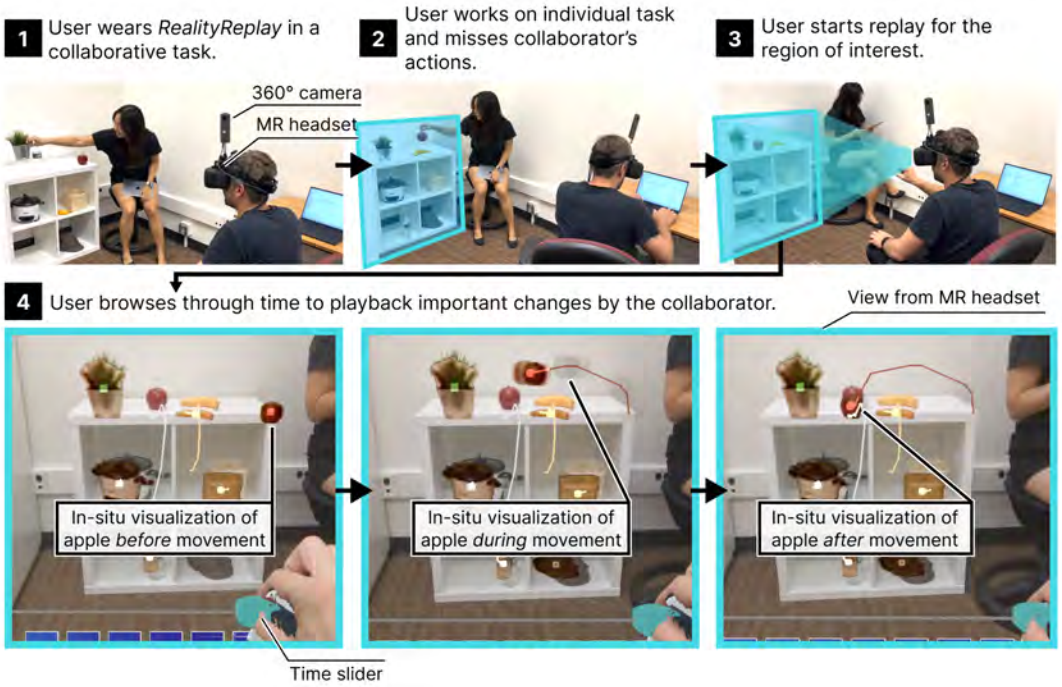

Fig. 1. We present RealityReplay, a system for in-situ playback of important past events that users had missed, using a head-mounted 360° camera and a head-mounted Mixed Reality (MR) display. This shows an example of RealityReplay in use. (1) Two users collaborate to re-arrange items on a shelf to create an inventory. One user catalogues the items on a computer while wearing the RealityReplay system. The collaborator on the left moves the items between different slots. (2) While the RealityReplay user inputs an item into the computer, the collaborator moves another object. This is initially missed by the user, who lost track of the items' positions. (3) To find out which items were moved and their original positions, the user looks at the blue *primary region* and activates RealityReplay. (4) The user then uses the time slider (bottom) to review what happened while they were away. The movements of objects are highlighted by a trajectory visualization, as well as a light shadow and an object mask that make it easier for users to identify which item the visualization belongs to.

Authors' addresses: Hyunsung Cho, Human-Computer Interaction Institute and Carnegie Mellon University, Pittsburgh, PA, USA, hyunsung@cs.cmu.edu; Matthew L. Komar, Human-Computer Interaction Institute and Carnegie Mellon University, Pittsburgh, PA, USA; David Lindlbauer, Human-Computer Interaction Institute and Carnegie Mellon University, Pittsburgh, PA, USA, davidlindlbauer@cmu.edu.

2474-9567/2023/9-ART90
https://doi.org/10.1145/3610888

ABSTRACT

Humans easily miss events in their surroundings due to limited short-term memory and field of view. This happens, for example, while watching an instructor's machine repair demonstration or conversing during a sports game. We present *RealityReplay*, a novel Mixed Reality (MR) approach that tracks and visualizes these significant events using in-situ MR visualizations without modifying the physical space. It requires only a head-mounted MR display and a 360-degree camera. We contribute a method for egocentric tracking of important motion events in users' surroundings based on a combination of semantic segmentation and saliency prediction, and generating in-situ MR visual summaries of temporal changes. These summary visualizations are overlaid onto the physical world to reveal which objects moved, in what order, and their trajectory, enabling users to observe previously hidden events. The visualizations are informed by a formative study comparing different styles on their effects on users' perception of temporal changes. Our evaluation shows that RealityReplay significantly enhances sensemaking of temporal motion events compared to memory-based recall. We demonstrate application scenarios in guidance, education, and observation, and discuss implications for extending human spatiotemporal capabilities through technological augmentation.

CCS Concepts: • **Human-centered computing → Mixed / augmented reality**; • **Computing methodologies → Perception**.

Additional Key Words and Phrases: Mixed Reality, Augmented Reality, Computational Interaction

ACM Reference Format:
Hyunsung Cho, Matthew L. Komar, and David Lindlbauer. 2023. RealityReplay: Detecting and Replaying Temporal Changes In Situ Using Mixed Reality. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 7, 3, Article 90 (September 2023), 25 pages. https://doi.org/10.1145/3610888

1 INTRODUCTION

Humans inevitably miss events in their environment that occur outside of their field of view (FoV). This happens, for example, while observing an instructor demonstrating how to operate a machine and turning away to take notes; or while watching a sports event such as football and turning to a friend for a quick chat. Even after observing the events, humans easily forget them due to our limited memory capacity. For example, humans are prone to being oblivious to complex sequences of instructions (e. g., cooking recipes, dance instructions, etc.) or subconscious actions (e. g., putting away glasses or a keychain on a desk). In order to catch up with such missed events, users have to rely on the verbal recollection of others. This, however, is only feasible if others can provide accurate information, and if they are around after all.

The ability of Mixed Reality (MR) technology to display dynamic contents opened the way to solutions to the challenge of missing or failing to recollect important information. Previous approaches extended the human field-of-view (FoV) through head-mounted sensors and cameras (e. g., [1, 9, 24, 48]), for example, such that users can observe a wider range of real-time events while they happen. With these approaches, users still miss events if they do not observe them while they happen. Other approaches record a full environment, such that users can play back all events in that room asynchronously [10, 34], or play back a per-object history for improved sensemaking [31] in Virtual Reality (VR). These approaches, however, require fully-instrumented special-purpose rooms and their virtual replicas, meaning that users cannot benefit from these systems when leaving the room, e. g., to run errands, attend outdoor events, or visit new places.

In this work, we introduce RealityReplay, an MR-based system that detects and records important changes that users missed in their environment. RealityReplay tracks users' surroundings while they perform tasks such as participating in meetings, observing sporting events, or watching an instructor, and analyzes the input video for changes. When the user wants to play back a missed event, e. g., because they did not see where someone else placed an object, they look at the region of interest, and RealityReplay provides them with in-situ MR summary visualizations of the past events that indicate the movement of objects directly on top of the environment (Figure 1). Users can control the playback of events via a simple slider-based interface in MR. This

enables users to directly make sense of what changes have occurred while they were not attending a specific area. RealityReplay employs a combination of see-through MR headset and head-mounted 360°camera that provides an omnidirectional video as input. Our approach takes an egocentric tracking approach for portable MR usage and does not require instrumenting the environment or prior knowledge about the environment. We envision that future MR headsets will provide this functionality without having to rely on external hardware, as they already employ camera-based inside-out sensing for applications such as hand tracking.

RealityReplay contributes a novel end-to-end pipeline that detects important changes in users' immediate environment through user-centric sensing, and then outputs a meaningful summary visualization of temporal changes. To achieve this, the system detects and tracks changes through a combination of semantic segmentation and saliency prediction. After users specify the area of interest they would like to have a summary from, RealityReplay first performs semantic segmentation to detect objects and their boundaries. To filter out static and less important objects, RealityReplay then performs saliency prediction on the same region, and combines the results to retrieve image masks of areas that changed significantly. RealityReplay uses the information of where and when changes happened to create a summary visualization. Users can choose between different types of summary visualizations with varying level of abstraction, from simple motion lines to a joint visualization of texture, shape, and position. The design of the visualization is informed by a small-scale formative study (Section 4.3.2), where we compare the effects of three visualizations (motion replay, motion history, and motion lines) and a video representation on perception of temporal changes. To enable playback, RealityReplay further stabilizes and aligns the image of the head-mounted 360° camera and the user's field of view.

We evaluated the performance and efficacy of RealityReplay in assisting users in sensemaking of temporal motion changes through a user study (n = 14). We compared our approach to memory-based recall and active monitoring. Participants achieved significantly better performance in identifying the order and trajectory of objects with RealityReplay, with reduced mental load and a higher level of success, in comparison to memory-based recall and active monitoring. RealityReplay captured and visualized 97.6% of all object movements, with 11.6% of missing frames on average. These results show that RealityReplay helps users in sensemaking of temporal motion events.

We demonstrate the versatility of the proposed approach by showcasing a set of applications, specifically using RealityReplay for collaboration and guidance, multi-tasking, security monitoring, and memory assistance, described in the next section. We believe that RealityReplay is a step towards exploring usages of MR technologies beyond accessing apps towards enhancing users' capabilities, in line with research on human augmentation [27, 41, 60]. The source code is available at https://augmented-perception.org/publications/2023-realityreplay.

2 APPLICATIONS

In the following, we showcase a set of application scenarios of RealityReplay, shown in Figure 2.

Collaboration and guidance. RealityReplay can be used in the context of collaboration, guidance, and instructions. As an example, we use our system to monitor progress for pick-and-place applications, or when sorting items on shelves (Figure 1). While the collaborator arranges items, the user of RealityReplay takes notes and keeps a list of items. In case the user loses track of what the collaborator did, they can activate RealityReplay and replay the placement actions. Alternatively, users can use RealityReplay to follow procedural instructions, such as the order of ingredients while cooking (Figure 2a). RealityReplay filters only important changes and composes a visual summary even when users get distracted and miss parts of the live instructions. Similar scenarios include retrieving instructions to assemble circuits or maintaining complex machines.

Multi-tasking. Users can also facilitate multi-tasking by offloading a monitoring job to RealityReplay. For example in the context of personal training (Figure 2b), an instructor can check students' progress, i. e., how many repetitions are completed while taking an important call or instructing another trainee.

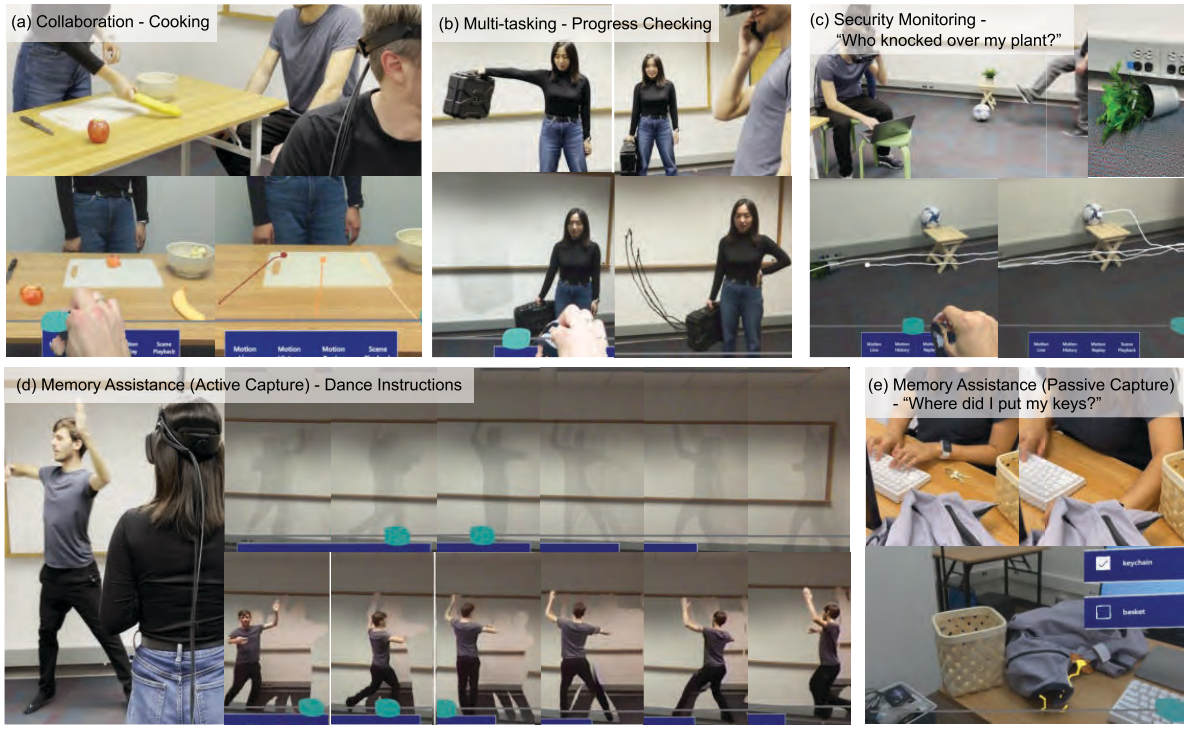

Fig. 2. Illustration of potential applications implemented using RealityReplay, including cooking, demonstration of fitness exercises, security monitoring, dance instructions, and personal memory assistance. For all those applications, RealityReplay enables users to replay events, which they missed due to limited FoV or short-term memory, multiple times for better understanding. The top row of each figure for each application represents the scene view, and the bottom row corresponds to the user's view from the MR headset while interacting with RealityReplay.

Security monitoring. RealityReplay can be used for general monitoring of scenes (Figure 2c). In case users turn away from a scene (e. g., table in coffee shop) and return to an altered scene, they can replay what happened while they were not attending the primary region, e. g., to figure out the cause of an incident.

Memory assistance. RealityReplay can also assist users in comprehending complex series of events that they could not memorize immediately, even though they directly observed the event (Figure 2d). Students can replay the instructor's movement to follow instructions of complex movements at their own pace or when they missed while performing other tasks (e. g., trying out a motion themselves or drinking water). Furthermore, RealityReplay can assist in recollecting important events that were performed subconsciously and thus could not be remembered. For example, people often forget where they placed their keys, glasses, or phones when they pay attention to other tasks (Figure 2e). RealityReplay can detect the movement of the keychain and visualize the trajectory to help the user find the misplaced key, assisting recall of subconscious actions.

Other applications. We believe that RealityReplay can be useful in a wide range of other applications. It can be useful when watching team sports such as basketball or football. Users can activate the system for a specific region of interest. In case they are distracted (e. g., talking to another person), RealityReplay enables them to replay movements of people (e. g., defense players in basketball) or objects (e. g., the motion of a pool ball) in-situ

as visual overlay. We believe that this leads to a much more embedded experience than traditional replays e. g., on large screens. Other example applications include replaying events for physical games such as Connect 4 or Jenga; tracking progress of sketches and brainstormings on whiteboard; or visualizing past movement of robots in a manufacturing site.

3 RELATED WORK

3.1 Extending the Human Field of View

The human FoV without eye movement extends roughly 200° horizontally and 130° vertically [56]. This means that users inevitably miss events that happen outside this area, particularly events that happen behind them. Various research explores technological solutions to extend the human FoV via 360° cameras, such as JackIn head and JackIn Space by Kasahara et al. [24] and Komiyama et al. [28]. They showcase applications for collaboration and telepresence, typically enabling others to take on the perspective of the user who wears the 360° camera. In contrast, FlyVIZ [1] and LiDARMAN [41] enable users to see *themselves* from a third-person perspective by combining VR headsets with head-mounted 360° cameras or Lidar, respectively. SpiderVision [9] enlarges users' visual field by exploiting a front and back camera mounted on a VR headset. As soon as the system detects motion, the view of the back camera is overlaid on top of users' FoV. All those works extend the human FoV for active monitoring. However, the broadened visual channel adds more sensory information for users to parse in real time and has the potential to deplete limited memory capacity at a faster rate. Our work leverages a head-mounted 360° camera for finding and visualizing events that users might have missed both *spatially and temporally*, so the user has full control over choosing the space and time for recall. In our evaluation, we show that the asynchronous approach of RealityReplay outperforms an active monitoring baseline. Besides direct augmentation of users' FoV, various works aim to increase awareness of out-of-view objects such as the work by Gruenefeld et al. on EyeSee360 [17, 18]. Our work is complementary to their work, as we are concerned with in-situ playback (i. e., visual overlay), as well as enabling users to view and make sense of historic data. The combination with such out-of-view visualizations, however, presents an interesting line of future research.

There exists much research that aims at enhancing the visual abilities of humans. This ranges from amplifying (slow) movements so they become perceivable (e. g., Knierim et al. [27]) to visual enhancements of the environment for low-vision users (e. g., Zhao et al. [60, 61]) and magnifying users' vision in Navicam by Rekimoto [48]. These works, as well as ours, aim to provide users with information that they would have otherwise missed, or to enlarge their visual abilities. Our work extends and complements these works by taking the temporal dimension into account. Veas et al. [55] direct users' attention subtly towards target areas by modifying visual saliency. In our work, we leverage saliency detection rather than manipulation of saliency to track and filter important events.

3.2 Temporal Interaction

We refer to temporal interaction as enabling users to perceive or interact with data that has changed over time, particularly in the field of MR. Our work builds on two connected lines of research: enabling users to view historic data, and manipulation of video data. AsyncReality [10] enables users to replay events that have happened in their surrounding area. Recording is automatically triggered by a causality-preservation algorithm. This work requires room-scale instrumentation for tracking, whereas RealityReplay relies on ego-centric tracking using a 360° camera, which can be more flexible. Additionally, AsyncReality directly plays back events in a space, e. g., a person that entered a room. RealityReplay provides summary visualizations that are directly embedded as MR overlays into the physical world. Our work further contributes empirical findings on how such reality playback system could assist enhance users' sensemaking ability through extended spatial and temporal awareness, and

that they are preferred compared to a simple playback in MR. It would be interesting to integrate AsyncReality's trigger-detection approach into our pipeline for automatic extraction of important events.

Remixed Reality [34] uses a related room-scale tracking setup with multiple depth cameras mounted in the ceiling. The approach enables recording and playback of the full environment space, rather than selective playback with abstract visualization like RealityReplay. Asynchronous interaction in MR has been explored in the context of collaboration and task instruction as well (e. g., [19, 21, 30]). In collaboration and task instructions, the MR visual cues are constructed and annotated by a producer or an instructor, whereas RealityReplay automatically generates summary visualizations of important temporal changes in-situ.

Besides recorded live events, Lilija et al. [31] allow users to directly manipulate VR recordings for better scene understanding. Their system records the movement of objects in VR and shows their trajectory over time. Given that their system has full control over the immersive world, they can record all movements of the whole environment, which is not possible in the real world. Tesseract [39] further enables users to query VR spatial design recordings for collaborative design. We demonstrate a system that extracts and tracks important events in the physical world, rather than VR. Additionally, we expand on their work by providing different summary visualizations that are distinct from their trajectory visualization. RealitySketch by Suzuki et al. [54] extracts motion information from videos and enables users to interact with the data through embedded graphics. Our event detection approach would be an interesting extension of their direct manipulation techniques.

In terms of manipulation of video data, Wildemuth et al. [57] show that in conventional videos, a speed-up of 1:64 is suitable for fast-forwarding videos while keeping the contents comprehensible. This insight has been exploited in approaches to quickly browse videos (e. g., [8, 15, 23]). Nguyen et al. [43], for example, propose a 3D Direct Manipulation Video Navigation (DMVN) system to resolve temporal ambiguities by mapping the temporal dimension to the depth coordinate in 3D. Our work is informed by these works and enables users to quickly make sense of historic data in MR.

3.3 Visualizing Motion

RealityReplay was inspired by previous work that visualizes motion. With MoSculp, Zhang et al. [59] created a system that fabricates physical manifestations of moving objects from a video. Kazi et al. [25] created ChronoFab, a system that conveys motion in static objects. Oshita [44] and Balasubramanyam [2] demonstrated how to visualize motion in a volumetric manner and how to represent motion in a spherical representation, respectively. Lastly, Cutting [7] discusses ways to represent motion in static images. They discuss effects such as symmetry, lean, blur and vector-like action lines with respect to criteria such as evocativeness, clarity, direction, and precision. We extend these works on motion representation to in-situ MR visualization. As an example, we utilize vector-like action lines to represent change in objects. Our work enables users to perceive previously missed events to enable precise recall of prior events, including motion trajectories.

3.4 Interacting with 360° Video

We leverage omnidirectional video of a head-mounted 360° camera to find important events in a scene. Researchers have explored ways to leverage 360° data for interactive purposes. Speicher et al. [53] developed 360Anywhere, a system for remote collaboration that enables remote participants to communicate with in-person users through projected annotations. HindSight [51] detects objects surrounding the user using 360° video and sonifies the position and class of objects through bone-conduction headphones. Huang et al. [20] created a system for automatic sonification of 360° video. Liu et al. [35] and Schoedl et al. [50] contribute different types of video textures for seamless video playback loops with gated clips to ensure that viewers do not miss important narrative elements in 360° videos. 360° filmmakers can create gated clips by specifying the gated timecode and the region of interest (ROI) so that the clip would proceed to the next part only if the viewer looks at the ROI. There also

Table 1. Classification of related work with respect to parameters of event detection and filtering, tracking method, connection between virtual and physical objects, representation and spatio-temporal expansion of users' abilities.

	Event detection and filtering		Tracking target and method			Physical-virtual connection	Representation		Dimension	
	Automatic	Manual	Real / Egocentric	Real / External	Virtual	Connected	Processed	Raw	Spatial expansion	Temporal expansion
RealityReplay	**X**		**X**			**X**	**X**	**X**	**X**	**X**
AsyncReality [11]	X			X				X	X	X
Remixed Reality [35]		X		X				X	X	X
Who put that there? [32]	X				X		X			X
Tesseract [40]	X				X			X		X
MoSculp [60]		X			X		X			X
360 off-view visualization [18]		X			X		X		X	
360 video systems [42, 49]		X	X					X	X	
Video summary [9, 16, 24]	X				X		X		X	
Async collaboration [20, 22, 31]		X	X			X		X		X

exists a range of works on shot orientation for 360° videos, such as the work by Pavel et al. [45], which aims at improving the viewing experience for users. In our work, we automatically extract information from the relevant areas for later presentation to users. Leveraging more advanced editing methods and methods for information extraction from 360° videos would be an interesting extension of our approach.

3.5 Summary and Categorization

We identified a set of main categories to contextualize RealityReplay within prior works, shown in Table 1. Each of the categories is applicable to different applications, and enables us to highlight their differences, rather than ascribing value to either category. *Event detection* specifies whether approaches automatically or manually detect important events. While with RealityReplay, users can choose between automatic detection, other works such as MoSculp require manual specification of events. *Tracking target and method* refers to whether the target of tracking is real or virtual (e. g., objects in 2D videos, virtual 3D models), and whether the real-world tracking is performed in an egocentric manner (e. g., head-mounted camera) or using external sensors (e. g., room-mounted cameras). While RealityReplay is a mobile system that does not inherently rely on room-scale tracking, AsyncReality or Remixed Reality require external sensors. Other static video summary tools, for example, were built for desktop environments. *Physical-virtual connection* relates to whether the virtual contents are directly connected to the physical objects, or separate visualizations. AsyncReality [10], for example, replaces the physical object while showing its trajectory, while RealityReplay shows trajectory visualizations in a connected manner. The work by Lilija et al. [31] shows trajectories of movement, but only of virtual elements in fully immersive environments. *Representation* refers to whether a summary visualization of events is displayed (e. g., a trajectory), whether events in the room are partially but directly visualized (e. g., movement of objects), or whether a full environment is recorded. Previous work has discovered advantages of processed representations of past events in contrast to the raw representation of data, e. g., videos, in terms of aspects such as immersion, ease of noticing what had happened, and enjoyment. Our work investigates the effects of different representations in understanding real-life events through virtual MR overlays (Section 4.3.2). Lastly, *Dimension* describes whether users are enabled to see contents outside their typical field of view (spatial expansion) or beyond what they currently see (temporal expansion).

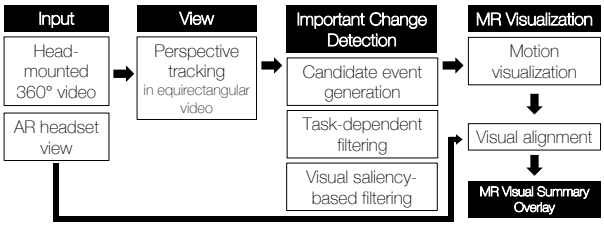

Fig. 3. Overview of the different modules of RealityReplay.

Fig. 4. One frame of the 360° input video (*top*), extracted primary region (*bottom left*) used for processing, and MR headset view (*bottom right*) used for alignment.

4 REALITYREPLAY

RealityReplay aims at helping users make sense of missed visual history. To achieve this, RealityReplay takes input from a head-mounted 360° camera, detects and tracks changes in the environment while users are looking away, and then enables users to replay the missed changes through in-situ MR summary visualizations. An overview of the system can be found in Figure 3.

4.1 Input

We take input from two main sources: a video stream of a head-mounted 360° camera and the camera view of an MR headset. An example of the input is in Figure 4. The hardware is shown in Figure 8. We believe that the capabilities of the system can be integrated with future consumer-friendly hardware by placing two 180-degree cameras on the sides of the headset, rather than the current standalone 360° camera on top.

We use the video of the head-mounted 360° camera for processing and a camera view of the MR headset for alignment. For clarity of exposure, we describe the processing as offline process, whereas in our software, processing starts immediately and can provide visualizations few seconds after users fixate on the region of interest. Our processing pipeline takes as input the frames of the 360° video in equirectangular projection, denoted as $I_t \in (I_1, \ldots, I_N)$, where I_t refers to the frame at timestep t of all N frames.

Perspective Tracking. Users first define the *primary region of interest*, i. e., the area in which RealityReplay tracks events, by turning towards it and pressing an MR button to activate the system. Conceptually, RealityReplay could easily track changes in the full visual field, and users decide which region to observe after recording. We chose a limited manual approach to decrease computational complexity.

Internally, RealityReplay marks the primary region, denoted as I', that users are currently facing. Once they turn away from the primary region, video frames of the primary region are recorded, processed, and stored. Upon returning to the primary region, processing is finalized, and users start playback. An example primary region is shown in Figure 4. We currently leverage the absolute position system of the MR headset for tracking the region. Correspondence between the 360° camera and the MR headset is created by tracking a single ArUco marker [11] in the room. We chose this method as the currently employed hardware is wired and fixed in a certain room.

Future versions of RealityReplay can provide a more flexible implementation by tracking environment features through SIFT [36] or other approaches for localization. At the time of primary region marking, we store the relative spherical distance between the marker and the center of the view.

During tracking, we compute the spherical coordinates of the primary region and extract the flattened perspective at these coordinates from the equirectangular 360° frame. We store the cropped primary regions for all frames, denoted as $(I'_1, \ldots, I'_N)$, that were captured while the user was facing away for further processing.

Stabilization. In addition to tracking the primary region, we perform simple stabilization to account for small-scale head movements. For each frame, RealityReplay first identifies a set of edges to track in the image (Shi & Tomasi [52]) and calculates the optical flow using the Lukas-Kanade method [37]. We then calculate the affine transformation between two consecutive frames using the optical flow at the location of the detected features. This transformation is applied to the primary region in all frames I'.

4.2 Tracking Important Motion Events

Recording and replaying every change that occurred while the user's attention has drifted away will clutter user's visual field with excessive information, being another source of distraction itself. Therefore, RealityReplay extracts only the important portions of the scene or event that happened.

To achieve this, our approach extracts visually significant changes in a scene using object-based segmentation (Detic [62]) and saliency prediction (TASED-Net [40]). Note that our pipeline is agnostic to the underlying approaches, i. e., its accuracy will only improve with future developments of the underlying components. In the evaluation, discussed later, we found that our system detects 97.6% of object movements.

The goal of our approach is to find important events in the input frames of the primary region I' and visualize those. Specifically, we aim to find objects that move, filter them by significance, and visualize their trajectory.

4.2.1 Generating a List of Candidate Events. For each input frame, we run the semantic segmentation model, resulting in a list of object proposals for each frame I'_t, denoted as $M_t = \{\mathbf{b}_t, \mathbf{f}_t, o_t\}$. Following the notation of Detic [62], $\mathbf{b}_t \in \mathbb{R}^4$ corresponds to the bounding boxes of the found objects, $\mathbf{f}_t \in \mathbb{R}^D$ is the D-dimensional region feature (i. e., the mask), and o_t corresponds to the label confidence.

4.2.2 Task-dependent Filtering. Semantic segmentation models can detect a large number of different objects (e. g., 20K for Detic [62]). Certain applications, however, benefit from more fine-grained filtering. Our pipeline features task-dependent filtering, specifically modifying the list of classes that the model outputs. For the evaluation, for example, we set an explicit allowlist to avoid distracting users. For other applications, an explicit blocklist can be beneficial (e. g., do not track people when working on a whiteboard).

4.2.3 Filtering Based on Visual Saliency. Presenting users with all possible labels and regions in an image might lead to clutter, limiting users' ability to gather insights into past events. We therefore apply a filtering mechanism based on visual saliency prediction for which we employ TASED-Net [40]. TASED-Net is a 3D fully-convolutional network architecture that produces a saliency heatmap S_t for a given input image sequence. The heatmap corresponds to the inferred probability that each region in an image would attract human gaze. We run TASED-Net for all recorded frames of primary regions and retrieve the heatmap for a given frame I'_t with the input frames $(I'_{t-K+1}, \ldots, I'_t)$, where K is the length of the image sequence ($K = 32$ for TASED-Net). We then calculate the overall saliency value for each object proposal normalized per frame in the found sequence as

$$s_b = \frac{1}{N} \sum_{t=1}^{N} \frac{S_t(\mathbf{b}_t)}{S_t}. \tag{1}$$

Lastly, we filter all objects that did not attract saliency above a minimal value ($\epsilon_s = .01$), resulting in a list of important object proposal M' (i. e., a filtered version of all detected objects in regions M). This is the input for the temporal summary visualizations.

4.3 In-situ Summary Visualization for Temporal Changes

RealityReplay aims to efficiently summarize important temporal changes through in-situ visualization. We break down this goal into subgoals of visualizing *what* moved (clarity), *in which direction* it moved (direction), and *how much* it moved (precision) based on Cutting's criteria [7]. As an initial step, we designed three types of visualizations tailored for each subgoal—motion replay, motion history, and motion line (shown in Figure 5). We conducted a small-scale formative study to gather insights into the individual abstract visualizations and inform the design of the final visualization, described below.

4.3.1 Three Types of Visualizations. **Motion replay** is a transparent overlay masked with the regions of changes. Specifically, for each frame, we create an image I_t'' that is transparent except in the regions f_t in M_t'. It preserves the full texture of the change, focusing on the clarity of *what* moved. **Motion history** shows shadowed silhouettes of the moving object with fading transparency. It keeps the shape of the object that changed but disregards its texture. For both Motion replay and Motion history, RealityReplay presents the past five frames to users, with fading transparency in case of Motion history. These fading shadows indicate the *direction* of the movement. **Motion lines** shows the trajectory of the object's movement as a line. It focuses on the positional change of *how much* the object moved, retaining neither its shape nor texture information. For each frame, we extract the positions of all important objects by calculating the centroids of the regions f_t in M_t'. The centroids of each object are connected to a line in the object's color. Time windowing is not applied to motion lines.

4.3.2 Small-scale Formative Study. We performed a small-scale formative study to investigate the effects of the three visualizations on how users perceive temporal changes through comparison, and to inform the final design of the visualization. We used a within-subject design with two independent variables, *visualization* (four levels) and *complexity* (4 objects and 8 objects). For *visualization*, we employed the three visualizations described above in Section 4.3.1 and Figure 5, and added a *no abstraction* condition for comparison, which shows a full-size video overlay of the primary region without any processing.

Twelve paid students (8 male, 4 female) from a local university (aged $M = 24.0, SD = 2.1$ years) participated. On average, the study took 71 minutes to complete, and participants were compensated with a $15 Amazon gift card for their participation. We used the hardware (Varjo XR-3, gaming computer) and software (Unity 2019, Python 3.7) described in the Implementation section. RealityReplay was configured to run in real-time, and participants could control the replay functionality with a simple time slider.

Figure 6 illustrates the study setup. Participants sat on an office chair and turned 180° towards a tablet. While they watch a distractor video, the experimenter moved the objects on the table. After the movement was completed, participants rotated back to the primary region and were asked to explore the temporal visualization of the condition using RealityReplay. During playback, they were asked to name objects that moved and describe the motion trajectory of the objects and the order of movements.

Between each condition, we asked participants to verbally describe why they liked or disliked using each visualization to complete the task. These responses were transcribed for analysis. After completing all conditions, to enable users to compare the visualizations holistically, they completed a questionnaire to describe why they liked or disliked each visualization: *"What did you like/dislike about this visualization?"*. We analyzed each participant's verbal explanations as well as the questionnaire responses through open coding to generate codes that characterize each visualization's pros and cons.

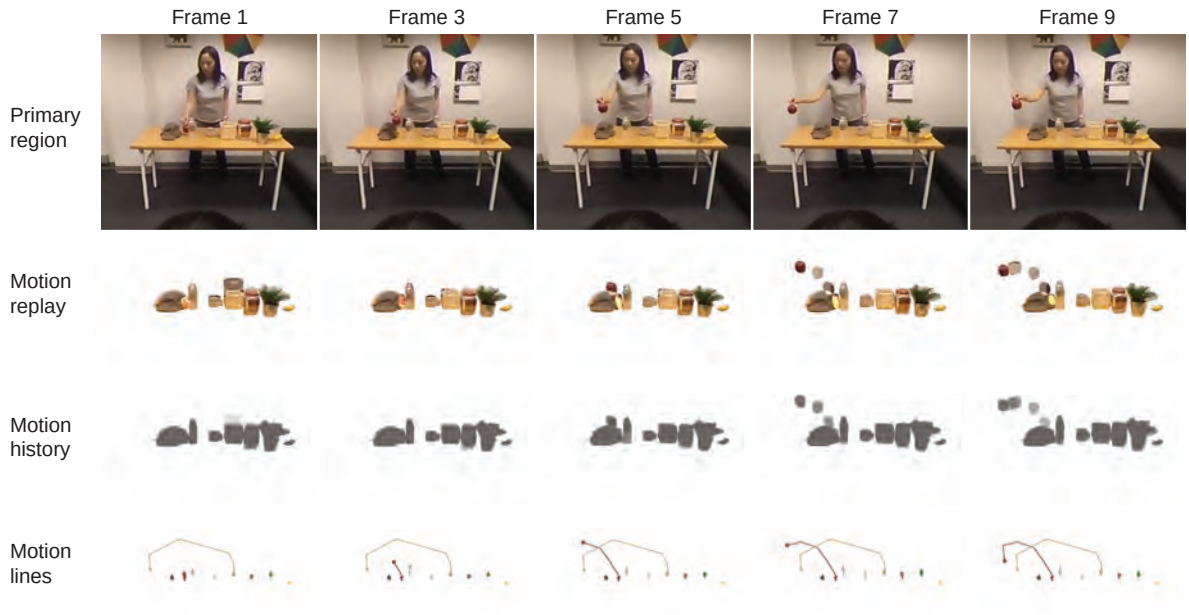

Fig. 5. Overview of the visualizations for a sequence of frames. Motion replay shows the regions of change in the original texture, shape, and position of the subject of change. Motion history keeps the shape of the object in change with trailing shapes from the previous five frames but disregards the texture of the subject of change. Motion line focuses on the positional change, connecting the centroids of subjects in motion across all frames.

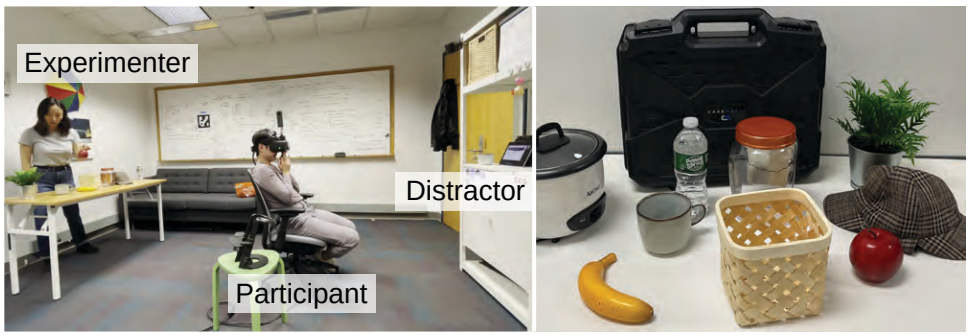

Fig. 6. Setup of the evaluation (*left*). A different set of objects was placed on the table in front of participants, and moved once they hard turned away. (*Right*) shows all objects used in the evaluation.

4.3.3 Results of Small-scale Formative Study. Users successfully reproduced events for all complexities. Less abstract representations resulted in higher precision, especially for more complex scenes. However, ten of the twelve participants mentioned that isolating important information through abstraction reduced distraction and obtrusiveness.

Motion lines captures the full history of the moving object and thus requires less efforts to memorize the changes (n=11) with less visual clutter (n=4). However, the color-based association between an object and its line was not always intuitive, especially when there were adjacent objects with similar colors (n=8). As they rely on color and position information, motion lines are also more prone to jitters and alignment (n=4).

Motion history provides enough information about the objects through shapes (n=8) while remaining unobtrusive (n=5). The "onion skinning" effect through time-windowed shadows also makes it easier to infer the direction of movement when paused at any arbitrary frame as described by P3 and echoed by other participants (n=5). However, in high-complexity scenes, participants found it hard to distinguish between different objects with similar shape (n=6) and occlusions of shadows (n=2). Temporal aspects in motion history and motion lines further reduced scrubbing efforts and time for sensemaking by adding directionality and past trajectories to still frames.

Motion replay provides full information about the moving instances, making it easy to identify objects and associate them with corresponding visualizations (n=12). It also "isolates the object image from the background" (P3) making it non-invasive (n=3). The rich amount of information, however, was at times perceived as "overwhelming" (P5), introducing clutter when there were many objects (n=4).

No abstraction offers "context information including why and how an object was moved" (P10) (n=4), which are not portrayed in other visualizations. However, the extra information also presents noise and distracts users from focusing on the task (n=4). Participants mentioned that "too much information" (P3) in the whole primary region was "distracting" (P10) and required "more cognitive effort to isolate objects" (P8). Furthermore, the large video occludes the environment (n=3), making it hard for users to take advantage of MR and compare the content to the current reality through augmented overlays (n=11). Blocking the entire area made it "impossible to compare two states [reality vs. virtual contents] at once" so users "had to move back and forth often without the trail of shapes or line to assist" (P2).

4.3.4 Combined Visualization. Based on the results of the formative study, we developed a combined version of visualizations, shown in Figure 1 and 7. We merged the characteristics of motion line, motion history, and motion replay into one visualization. We display the motion line for better temporal understanding, paired with motion history for short-distance understanding and motion replay for object-visualization mapping. Figure 7 illustrates the per-object and all-objects versions of the combined visualization. Users can choose which objects they would like to see as replay, or all combined.

The visualization is based on four key aspects that were present in the formative study. First, the visualization minimizes its containing information to reduce distraction and visual clutter. This is especially crucial for motion visualizations in MR because the visualization is situated in the real world. Second, the visualization supports clear association between the object and the visualization by showing the motion line and the virtual replica of the object at the end. Third, the visualization portrays the direction of movement (i. e., direction of time) through the motion history (gray shadow). These two echo the importance of clarity and direction in Cutting's criteria. Fourth, by including the motion line, the visualization maintains the full trajectory of motion, thus reducing memorization efforts.

4.4 Spatial Alignment of Visual Summary Overlay in MR

Presenting the visualizations in the coordinates of the input frames would lead to misalignment, as the view of the 360° camera and the view of users through the MR headset are not aligned due to the height difference (Figure 8).

Instead of performing a global alignment procedure, we perform a per-object region alignment. When users return to the primary region of interest, we capture an image from the perspective of the user through the front-facing camera of the pass-through MR headset, shown in Figure 4. We then run semantic segmentation on both the current frame of the 360° camera and the view of the MR headset. For each important object proposal in M' in both cameras, we retrieve the delta in translation and scale between the two cameras. The position and scale of all visualizations are then adjusted so that the parameters in the final frame corresponds to the position in the MR headset. All prior positions are set relative to this end position. If an object is not found because it

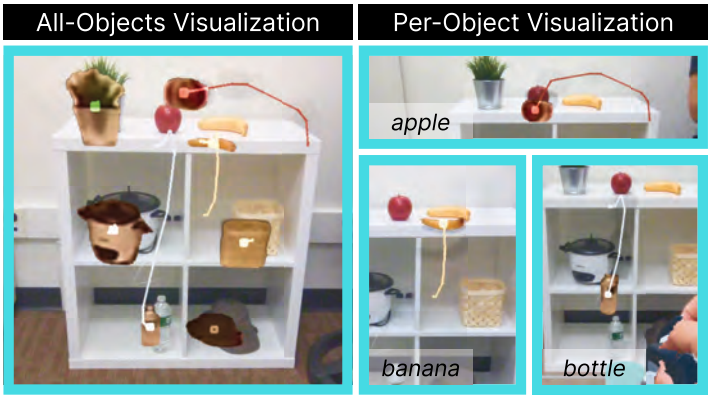

Fig. 7. Combined visualization of motion lines, motion history, and motion replay based on insights obtained from the formative study (Section 4.3). Users can select which object(s) to show the playback (per-object visualization), or choose to see all objects (all-objects visualization).

Fig. 8. Hardware setup for RealityReplay.

moved out of view, our system currently skips the alignment step for this object. Alternative approaches for out-of-view objects would be to use the average transformation from the other objects or align the object relative to the position at the beginning of the recording.

The visualizations are displayed as a textured plane in users' FoV that corresponds to the area of the primary region. Based on our tests, we fix the virtual plane at a distance of 2 meters, as this results in a comfortable viewing experience. Given that most of the plane is transparent, though, the overlay is only visible in areas where important changes happen, and is less distracting than a plane blocking the full FoV, as confirmed by our findings in the formative study (Section 4.3).

Ideally, RealityReplay would present the objects at the correct spatial position of the objects that moved, i. e., as 3D objects directly in the scene. Given that our current system only works with 2D information, however, this is not feasible. We hope to extend the system with this functionality in the future, either by incorporating omnidirectional depth cameras, or depth estimation techniques such as the work by Kopf et al. [29] that infer the depth of regions from monocular images.

5 IMPLEMENTATION

5.1 Hardware

RealityReplay requires display hardware with a reasonably large FoV to show information on the primary region, and a camera system that can track the region while users are facing away. This is implemented using commercially available but high-end hardware (Figure 8). We used a Varjo XR-3 video pass-through MR headset for the visualization. We chose this headset as it has a larger FoV (~120° horizontally) than other optical see-through headsets, and better ability to overlay visual information. For capturing the 360° video, we mount a Ricoh Theta V camera onto the headset. The camera sits approximately 11 centimeters above the headset (with ~25cm lens-to-lens difference) so that we can capture images when users are facing away. A more integrated approach would be to have 180° lenses on the sides (and potentially backside) of the headset to capture omnidirectional video. The 360° streams the video via USB to the host computer at its maximum streaming rate of 10 frames per second. The software runs on a gaming computer (Alienware Aurora R12, Windows 10, Intel Core i9 11900KF, NVIDIA GeForce RTX 3090 24GB, 32 GB RAM).

5.2 Software

RealityReplay is developed in Unity 2019 using SteamVR and the Varjo SDK for displaying MR content. A separate program written in Python 3.7 is used to retrieve and process the images from the 360° camera. The two are connected through a local socket. The Python program runs at approximately 3 fps. Processing happens while users are not attending the primary region and is finalized once users return to the primary region. We use Detic [62] for semantic segmentation, which identifies up to 20,000 objects at approximately 200 ms per frame. Saliency prediction is performed using TASED-Net [40], taking approximately 100 ms per frame. We chose those two approaches as they provide a good balance between accuracy and inference speed. This processing could be further sped up to be fully real-time by further parallelizing the semantic segmentation and the saliency model, and by using faster and more specialized models. We are confident that real-time processing at 30 fps would be possible, although not necessary, since the program runs in an offline manner after users return to the primary region.

We perform the equirectangluar-to-perspective correction using the Equirec2Perspec package[1]. Image stabilization including feature detection and optical flow calculation, as well as marker detection for primary region tracking, is implemented using OpenCV [4]. We use the native C++ Varjo SDK to extract the view of the headset and send the images to Python through a local connection.

The visualizations are implemented in Python and sent as frames with transparency (see Figure 5) to Unity. In the view of users in the MR headset, the visualizations with transparency appear as shown in Figure 2. Frames are cached so users can browse the temporal changes in real time. Users can interact with the visualizations using interface elements provided by the Microsoft Mixed Reality Toolkit[2], with hand tracking provided by the Varjo SDK.

6 EVALUATION

We evaluate the efficacy of our approach in a controlled lab setting through a user study. The study aims to measure the system and user performance of RealityReplay in making sense of temporal changes in motion, especially events that users might have missed otherwise.

6.1 Study Design

Participants performed two tasks that simulate different scenarios for active monitoring and recall. First, similar to the visualization preliminary study, participants were asked to describe verbally (along with hand gestures) (1) which objects moved, (2) in what trajectory and (3) in what order as a primary task. We fixed the number of objects to eight objects, with six objects moving in each condition.

In addition, participants completed a signal detection task as secondary task. This dual-task methodology is commonly used in research on peripheral displays (e. g., [5, 47]), human state estimation (e. g., [16]) and for evaluating novel enabling technologies (e. g., [32, 33]). In our task, a random alphabet letter was shown to participants in the MR headset, and they were asked to respond to a target stimulus (letter "K") as fast as possible (cf. [33, 47]). Participants performed the secondary task at different times depending on the study condition. We added this secondary task to simulate more realistic scenarios in which users would be performing multiple tasks simultaneously (cf. Section 2). Participants continued the secondary task while RealityReplay generated visualizations.

We used a within-subject design with four conditions: two baseline conditions (*Memorize* and *Active Monitoring*) and two versions of RealityReplay, specifically *RealityReplay (Abstraction)* and *RealityReplay (No Abstraction)*. Figure 9 illustrates all conditions. For *Memorize*, participants do not observe the motion changes and rely on

[1]Equirec2Perspec package https://github.com/fuenwang/Equirec2Perspec
[2]Microsoft Mixed Reality Toolkit https://github.com/microsoft/MixedRealityToolkit

Fig. 9. Overview of four conditions used in Study 2: Memorize, Active Monitoring, RealityReplay (Abstraction), and RealityReplay (No Abstraction). In Memorize, the participant tried to memorize the initial layout of items, turned to their back and performed the secondary task, and turned back again to guess what changes were made based on the difference between the initial and current state. In Active Monitoring, the participant observed the experimenter moving objects while performing the secondary task in the same region. In RealityReplay (Abstraction), the participant turned away from the experimenter as in Memorize, but they could use RealityReplay with abstract visualization using the time slider and checkbox in the MR headset view. RealityReplay (No Abstraction) had the same setup, but the participant used RealityReplay with no abstract visualization but a picture-in-picture view of the primary region.

their memory of the previous state. In the *Active Monitoring* condition, participants observe the changes live. In the *RealityReplay (Abstraction)* condition, participants do not observe the motion changes and use the abstract visualizations of RealityReplay as assistance to playback the changes. The *RealityReplay (No Abstraction)* condition is similar except that participants use the non-abstract version of RealityReplay, which enables users to play the video recording of the primary region with no abstraction. All participants performed all conditions.

We measured both system and user performances when using RealityReplay. For system performance, we report the success rates of object detection, moving object detection, and motion visualization. For user performance, we measured how well users are able to describe the intended temporal motion changes (i. e., primary task error rate), and how much the review distracts the users (i. e., secondary task error rate).

6.2 Participants & Apparatus

We recruited 14 paid participants (7 male, 7 female) from a local University (13 students, one research assistant), aged $M = 20.4$ ($SD = 2.2$) years, all with normal or corrected-to-normal vision based on self-reports. Participants had an average experience using MR devices of $M = 2.6$ ($SD = 0.76$) on a scale from 1 (none) to 5 (expert). No participants reported elevated susceptibility for motion sickness when queried using the Motion Sickness Susceptibility Questionnaire Short (MSSQ-Short) form [14]. Participants were compensated with a $15 Amazon gift card for their participation. On average, the study took 43 minutes to complete.

The experiment was conducted in a quiet lab space with the same apparatus, hardware, and software as the preliminary study (Figure 6). On top of the real-time replay functionality using the time slider, participants could select which object(s) to see the replay for in the RealityReplay (Abstraction) condition using the built-in controls of RealityReplay. In ths experiment, participants interactively made an explicit choice over which object(s) to include in the allowlist for visualizations instead of fully relying on saliency prediction results.

6.3 Procedure

Participants completed the consent form, demographic questionnaire, and MSSQ-Short. The experimenter explained the primary and secondary task of the study ("You will be asked to describe changes to the objects on the table that happened while you are looking away or observing depending on different conditions. At the same time, you will be performing a signal detection task where you need to press a left or right key on the keyboard based on the alphabet letter you see in the MR headset.")

Participants completed both two tasks for each condition. The order of conditions was counterbalanced using a Balanced Latin Square for 12 participants, and two additional participants followed two orders from the Balanced Latin Square. After each condition, participants completed the NASA TLX questionnaire and a questionnaire on visualization for RealityReplay (Abstraction) and RealityReplay (No Abstraction) conditions. The visualization questionnaire consisted of questions regarding usefulness of the visualizations to complete the task, rated on a Likert-type scale from 1 (low) to 5 (high), a tailored System Usability Scale (SUS), and why participants liked or disliked each visualization. All questions and their results are illustrated in Figure 12 and Figure 13 of the Appendix.

6.4 Data Collection

For system performance analysis, we processed the system logs to manually label each frame to find out which object is moving, whether each object was detected through the important change detection module, and whether each object motion trajectory was successfully captured in the visualization.

For user performance analysis, we measured the hit rate by comparing the answer and the ground truth video recorded through a separate smartphone camera. For the question "Which objects moved?", we calculated the number of correctly answered items divided by the total number of moved items. For the question "How did object X move?", we computed the number of correct answers where participants' description of the start position and direction of movement matched to the ground truth (i.e., to the current position from far left, mid-left, middle, mid-right, and far right). This value divided by the number of total object movements gives the accuracy. For the question, "What was the order of the moving objects?", participants listed the moved objects in the order of occurrence. We counted the number of mentioned objects that moved in the correct relative order. we checked for the correctness in relative order. For example, if the correct answer was 'pot, banana, bottle, hat, jar, crock pot' while the participant answered 'pot, bottle, banana, basket, jar, crock pot', the accuracy was 3 (pot, jar, crock pot) out of 6.

6.5 Results: System Performance

In all trials, perspective tracking worked correctly by capturing all 8 objects within the primary region in every frame. For important change detection, 97.6% of all object movements were represented in the combined visualizations. Specifically, 97.6% represents the recall score, i. e., correctly detected movements (true positives) out of all object movements (true positives plus false negatives). At the frame level, 88.4% of per-frame moving object instances were correctly detected on average. This indicates that the abstract visualization partially missed around 11.6% within an object's trajectory on average in most cases. Across all frames including non-moving frames, 93.8% of object instances were correctly detected by RealityReplay on average, 19129 out of 20392 total (2549 frames * 8 objects).

6.6 Results: User Performance

User performance measures users' end-to-end task success rate using RealityReplay. In summary, the results (Figure 10) indicate that participants could achieve better performance in sensemaking of temporal motion events

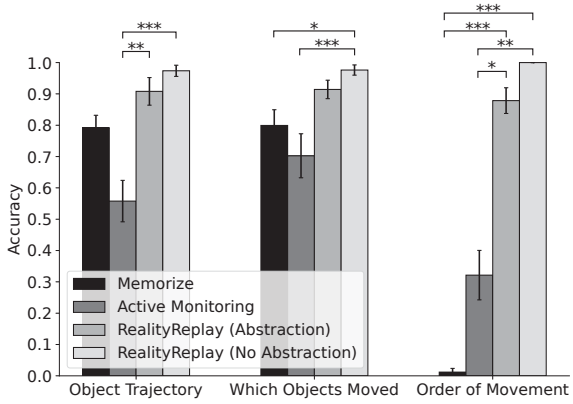

Fig. 10. User performance in describing how objects moved. Error bars indicate standard error.

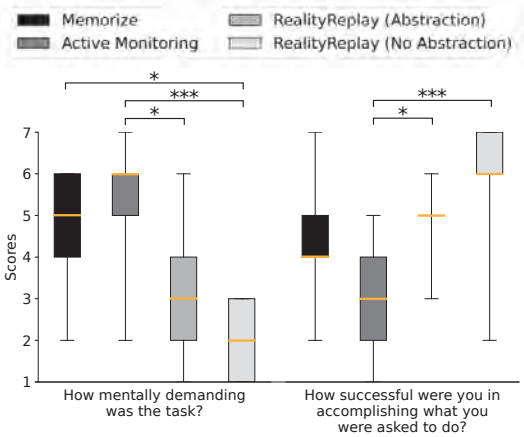

Fig. 11. User rating on the mental load and successfulness induced by the task as part of NASA Task Load Index (TLX) questionnaire. Full TLX results are in Figure 12 of the Appendix. The boxplot shows median as the orange line, interquartile range (IQR) as the box, and minimum and maximum values as the whiskers.

with RealityReplay in comparison to memory-based recall (Memorize and Active Monitoring), with no significant difference between Abstraction and No Abstraction.

We use JASP [22] for statistical analysis. We performed Friedman non-parametric tests to check for main effects in task performance in answering object-related questions, and Conover tests with Bonferroni adjustment for post-hoc testing when main effects were present. Results with main effects are illustrated in Figure 10.

For the Object Trajectory question "How did object X move?", results indicate a main effect, $\chi^2(3) = 23.067,\ p < .001,\ Kendall's\ W = 0.549$. Post-hoc tests show that compared to Active Monitoring, both RealityReplay (Abstraction) ($t = 3.398,\ p = 0.009$) and RealityReplay (No Abstraction) ($t = 4.505,\ p < .001$) yield significantly higher accuracy with 90.8% and 97.4% on average, respectively, whereas Memorize and Active Monitoring resulted in 79.2% and 55.8% correctly answered questions. Results also indicate a main effect for the question "Which objects moved?", $\chi^2(3) = 14.243,\ p = 0.003,\ Kendall's\ W = 0.339$. Post-hoc tests show that RealityReplay (No Abstraction) with the mean accuracy 97.6% leads to a significantly higher accuracy than Memorize ($t = 2.818,\ p = 0.045$) and Active Monitoring ($t = 3.416,\ p = 0.009$) with 79.9% and 70.2% mean accuracies, respectively. For the task of answering "What was the order of the moving objects?", results indicate a main effect ($\chi^2(3) = 39.071,\ p < .001,\ Kendall's\ W = 0.930$). RealityReplay (Abstraction) and RealityReplay (No Abstraction) yield higher accuracy (87.9% and 100.0%) than Memorize with 1.2% accuracy ($t = 4.293, p < .001$ and $t = 5.443, p < .001$, respectively) and Active Monitoring with 32.1% accuracy ($t = 2.990, p = 0.029$ and $t = 4.140, p = 0.001$, respectively).

There were no main effect for secondary task performance across conditions. The accuracy of each condition was generally very high: Memorize 98.8%, Active Monitoring 96.7%, RealityReplay (Abstraction) 98.0%, and RealityReplay (No Abstraction) 96.9%. This indicates that the secondary task, albeit attention demanding, was not difficult, reflecting a realistic scenario in which users perform a known task while monitoring a scene for changes.

6.7 Questionnaire Results

We analyzed participants' questionnaire responses to the NASA Task Load Index (TLX) and the usefulness of visualization questions. For the TLX, we performed Friedman non-parametric tests and post-hoc Conover tests with Bonferroni adjustment. For the visualization questionnaire, we performed a series of paired samples Wilcoxon signed-rank tests. In the following, we discuss main findings. All questionnaire data are shown in Figure 12 and Figure 13 of the Appendix.

Results indicate a main effect for the mental load ($\chi^2(3) = 25.898,\ p < .001,\ Kendall's\ W = 0.617$) shown in Figure 11. Post-hoc tests show that RealityReplay (No Abstraction) yields significantly lower mental load than Memorize ($t = 3.269,\ p = 0.011$) and Active Monitoring ($t = 4.713,\ p < .001$). RealityReplay (Abstraction) also induces significantly lower mental load than Active Monitoring ($t = 3.269,\ p = 0.011$). Results also indicate a main effect for the success ($\chi^2(3) = 28.053,\ p < .001,\ Kendall's\ W = 0.668$). Both RealityReplay (Abstraction) ($t = 3.229,\ p = 0.015$) and RealityReplay (No Abstraction) ($t = 5.181,\ p < .001$) resulted in significantly higher level of success than Active Monitoring.

Participants gave positive ratings ($M > 3.0$) in all questions of the visualization usability scale for both RealityReplay (Abstraction) and RealityReplay (No Abstraction). Participants also gave high ratings for the usefulness of RealityReplay in sensemaking of motion events in terms of identification of which objects moved, and the direction, degree, and order of moving objects. In comparison, participants generally found RealityReplay (No Abstraction) easier and more comfortable to use, and found the information in RealityReplay (No Abstraction) to be more useful for exact sensemaking, all $p < .05$, as indicated by the Wilcoxon tests illustrated in Figure 13. From the qualitative results, we believe that this is due to the difficulty in selecting the object checkboxes and in using the pinch-based time slider, in particular due to inaccuracies in the hand tracking of the Varjo headset. Participants noted physical load and difficulty of using the system, which they reported to have incurred from the unfamiliarity of using hand gesture-based interface in MR. We therefore believe that with improved interactions, the difference in perceived effort would be eliminated, and plan to change the general interaction in the future.

7 DISCUSSION

7.1 Evaluation Results

Our evaluation showed that RealityReplay enables users to mentally reconstruct events that they missed. In the following, we discuss insights about system performance, user performance, and subjective ratings, and contextualize them with respect to generalizability of RealityReplay and opportunities for future work.

7.1.1 System Performance. Our system was able to detect and visualize nearly all important changes, represented in a recall score of 97.6%. We did not calculate precision, as our system does not reject non-movement (i. e., false positives) but visualizes the masks for static objects. Detection performance relies heavily on the underlying semantic segmentation model, Detic in our case. Incorporating newer models (e. g., Segment Anything Model (SAM) [26]) or multimodal models (e. g., ImageBind [13]) that are trained on egocentric datasets [38] would further increase detection performance. Our proposed pipeline is flexible and allows for this type of substitution. Additionally, future versions of RealityReplay could incorporate methods that reconstruct 3D models from the 2D object masks [58] to further enhance the quality of the visualizations. Extending RealityReplay with newer models and 3D reconstruction capabilities would enable RealityReplay to detect and visualize fine-grained movements and small changes, which might be beneficial in applications such as training.

7.1.2 User Performance. RealityReplay enabled participants to maintain better awareness of what happened in their surroundings than Memorize and Active Monitoring conditions. Notably, participants showed the lowest performance with the Active Monitoring condition. We believe that this is because participants overestimated their ability to monitor and memorize events while engaging in a secondary task, and thus did not put as much

deliberate effort into memorizing the initial state. For the Memorize condition, they tried to memorize the original positions before the signal detection task, resulting in higher performance. These results indicate that RealityReplay is beneficial for users even in situations where they can actively observe events.

7.1.3 Subjective Ratings. Participants found RealityReplay to be less mentally taxing and exhibited higher perceived performance and confidence, highlighting the benefits of our memory assistance system. This is also reflected in the visualization usability questionnaire. Comparing the RealityReplay (No Abstraction) and RealityReplay (Abstraction) conditions, participants perceived RealityReplay (No Abstraction) as easier for exact sensemaking. Since this condition revealed the full information of a scene like a traditional 2D video, participants could digest the information with familiarity with the interface. The qualitative results of the formative study, however, indicate that the RealityReplay (Abstraction) condition provided users with a targeted summary of an event, which reduced distraction. This echoes the findings of Lilija et al. [31] that using objects' trajectories for sensemaking helped users become quick to inspect changes of objects in question in their sensemaking tasks to understand what had happened, as well as providing a sense of "being there" and fun. We believe that enabling users to select the appropriate visualization for their task provides them with the most benefits. Future work should evaluate RealityReplay in less controlled settings to deeply understand the benefits and limitations of the individual visualizations, and what their optimal application scenarios are.

7.2 Privacy and Ethics

RealityReplay in its current form requires continuous monitoring of the environment, which has obvious negative implications for privacy. Others might be inadvertently recorded without their consent, or users might see information that was not meant for them, not unlike in "real life" when users see events that they should not see. These challenges are exacerbated by our approach as it relies on omnidirectional video. Certain challenges can be overcome with hardware and software solutions: masking regions users never attended to, and only recording the primary region, for example; or relying on further scene analysis to detect events that should not be recorded, which in itself could introduce more bias in the system. Currently, users have to actively and manually trigger the recording of specific regions, which alleviates parts of the problem of always-on recordings. Like most MR systems, ours too relies on a continuous stream of multiple cameras for tracking and scene understanding. These challenges, and others with respect to bias in the machine learning methods used, for example, need to be taken into account when we investigate the feasibility of deploying technologies such as ours in the future. Manual activation and an active negotiation process of who, what, when, and how to record, in combination with an in-depth analysis of the employed components, will be necessary steps for future research. We believe, however, that the ability to replay missed events for education, training, and general decision-making purposes, is generally beneficial for users, and will enable more integrated MR approaches in the future.

7.3 Limitations and Future Works

We provide an initial implementation of our approach, which runs in real-time and provides benefits in terms of awareness and mental load, as shown in our evaluation. There are, however, limitations in terms of visualization quality and task-dependent filtering which we hope to address in future versions of RealityReplay. We believe that extending RealityReplay with multi-modal capabilities and extending its applicability to dynamic scenes present interesting directions for future research, as outlined below.

7.3.1 Immersive Visualizations. The visualizations are not perfectly aligned with the corresponding physical objects, as apparent in the figures showing our current prototype. The offset comes from the fact that all visualizations are presented as size-matched 2D images without depth information. Specifically, while the see-through experience is delivered through the stereo cameras of the Varjo headset, the visualizations are based on

the images of the left camera. Additionally, the visualizations are displayed on a virtual plane at a fixed depth. We believe that by incorporating depth information, the virtual overlays can be placed more accurately in future versions of our system.

7.3.2 Task-dependent Filtering and Context-awareness. Our current approach filters changes and moving objects based on their visual saliency, and provides manual task-dependent filtering capabilities. This only partly corresponds to task importance or personal preferences. In a scenario where multiple instructors demonstrate tasks, for example, users might want to choose which actions are important and should be tracked. Similarly, in a sports event, users might only be interested in replaying the movement of particular players. Extending RealityReplay with methods that enable direct manipulation and filtering techniques (e. g., Nguyen et al. [43]) would allow for more fine-grained control. We believe that this extension is also useful for future always-on Mixed Reality devices. In such scenarios, digital content such as notifications can easily become a source of distraction. RealityReplay can support users in recovering from these digital distractions by playing back which important changes happened during the period of distraction. Future work in this direction can draw insights from existing literature on automatic highlight generation in videos, especially with personal history [3, 49].

7.3.3 Beyond Visual Augmentation. Our current work is focused on visual replay. Extending RealityReplay with audio would create a more immersive and complete representation of past events. In situations where multiple sound sources are present, e. g., in a stadium while having a conversation with a friend, any future system would need to be able to extract audio from the primary region of interest, for example using multiple directed microphones.

7.3.4 Dynamic Scenes: Blending Replay and Current Events. Our system currently assumes a static scene once users return to the primary region and start playback. While this applies to many scenarios, it is not always the case. In a stadium, for example, a game might continue while users want to catch up. Similarly, in a dynamic group setting, others might continue their work. During playback, users therefore miss other events and need to prioritize whether to attend to the current or past environment. With the current prototype, once the user starts playback, the augmented scene will freeze at that moment. A future implementation could continue tracking of events in a separate thread so that the user can catch up with what they missed while interacting with the playback. Furthermore, as these scenarios require quick sensemaking of what had happened, we believe temporal summary visualizations such as motion lines or motion history can help "reduce scrubbing efforts and time for sensemaking by adding directionality and past trajectory to sill frames" as found in our formative study (Section 4.3.2). It would be valuable to create a merged approach that enables, e. g., a sped-up playback of the past to enable users to catch up more seamlessly. Visualization, tracking, and real-time blending are all challenging aspects that push the limits of software and hardware. We plan to extend our work with a more integrated approach in the future. RealityReplay shows that the general concept is valuable, and its implementation feasible.

7.3.5 Evaluation. We currently only evaluate RealityReplay in a lab setting and control for the task complexity and environment. Running more longitudinal and exploratory evaluations will certainly surface new interesting challenges, such as finding optimal time spans for replay. We hope to overcome the constraints of the employed hardware in the future, for example by using less embedded visualizations using optical pass-through devices such as the Microsoft Hololens.

Much prior work in MR focused on moving digital interfaces from 2D screens to 3D representations that are embedded in the physical world, essentially bringing current capabilities to novel devices. This is especially true for context-aware interfaces that leverage advanced sensing, optimization, and applied machine learning (e. g., [6, 12, 32, 46]), for example, for controlling when, where and how to display virtual user interface elements. In our work, we leverage similar (semi-)automatic methods, but use them to enable users to see real-world events

that they otherwise might have missed. This goes in line with prior research that aims at enhancing human capabilities (e. g., [27, 42, 48]). We hope to further combine approaches for advanced user interface adaptation in MR with methods to augment human capabilities in the future.

8 CONCLUSION

In this work, we present RealityReplay, a system that enables users to replay events in their environment that they have missed. We leverage an MR headset paired with an omnidirectional camera, and contribute a novel end-to-end pipeline based on semantic segmentation and saliency prediction to detect and track important changes. RealityReplay provides users with a set of summary visualizations of temporal events, which they can use for general sensemaking, improved situational awareness, and learning. We showcase a set of applications ranging from education, sports, and physical games. Our evaluation showed that in-situ MR visual summaries of temporal changes can assist users in understanding unattended past events in users' surroundings. RealityReplay reveals the potential of using always-on MR devices for in-situ support to expand human's spatio-temporal capabilities.

REFERENCES

[1] Jérôme Ardouin, Anatole Lécuyer, Maud Marchal, Clément Riant, and Eric Marchand. 2012. FlyVIZ: a novel display device to provide humans with 360 vision by coupling catadioptric camera with hmd. In *Proceedings of the 18th ACM symposium on Virtual reality software and technology*. 41–44.

[2] Adithya Balasubramanyam, Ashok Kumar Patil, Bharatesh Chakravarthi, Jae Yeong Ryu, and Young Ho Chai. 2020. Motion-Sphere: Visual Representation of the Subtle Motion of Human Joints. *Applied Sciences* 10, 18 (2020), 6462.

[3] Uttaran Bhattacharya, Gang Wu, Stefano Petrangeli, Viswanathan Swaminathan, and Dinesh Manocha. 2021. HighlightMe: Detecting Highlights from Human-Centric Videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 8157–8167.

[4] G. Bradski. 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000).

[5] Matthew Brehmer, Joanna McGrenere, Charlotte Tang, and Claudia Jacova. 2012. Investigating interruptions in the context of computerised cognitive testing for older adults. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 2649–2658.

[6] Yifei Cheng, Yukang Yan, Xin Yi, Yuanchun Shi, and David Lindlbauer. 2021. SemanticAdapt: Optimization-based Adaptation of Mixed Reality Layouts Leveraging Virtual-Physical Semantic Connections. In *The 34th Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '21)*. Association for Computing Machinery, New York, NY, USA, 282–297. https://doi.org/10.1145/3472749.3474750

[7] James E Cutting. 2002. Representing motion in a static image: constraints and parallels in art, science, and popular culture. *Perception* 31, 10 (2002), 1165–1193.

[8] Pierre Dragicevic, Gonzalo Ramos, Jacobo Bibliowitcz, Derek Nowrouzezahrai, Ravin Balakrishnan, and Karan Singh. 2008. Video Browsing by Direct Manipulation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy) *(CHI '08)*. Association for Computing Machinery, New York, NY, USA, 237–246. https://doi.org/10.1145/1357054.1357096

[9] Kevin Fan, Jochen Huber, Suranga Nanayakkara, and Masahiko Inami. 2014. SpiderVision: extending the human field of view for augmented awareness. In *Proceedings of the 5th augmented human international conference*. 1–8.

[10] Andreas Rene Fender and Christian Holz. 2022. Causality-preserving asynchronous reality. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*.

[11] Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco José Madrid-Cuevas, and Manuel Jesús Marín-Jiménez. 2014. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition* 47, 6 (2014), 2280–2292.

[12] Christoph Gebhardt, Brian Hecox, Bas van Opheusden, Daniel Wigdor, James Hillis, Otmar Hilliges, and Hrvoje Benko. 2019. Learning Cooperative Personalized Policies from Gaze Data. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 197–208. https://doi.org/10.1145/3332165.3347933

[13] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. 2023. ImageBind: One Embedding Space To Bind Them All. arXiv:2305.05665 [cs.CV]

[14] John F. Golding. 2006. Predicting individual differences in motion sickness susceptibility by questionnaire. *Personality and Individual Differences* 41, 2 (2006), 237 – 248. https://doi.org/10.1016/j.paid.2006.01.012

[15] Dan B. Goldman, Chris Gonterman, Brian Curless, David Salesin, and Steven M. Seitz. 2008. Video Object Annotation, Navigation, and Composition. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology* (Monterey, CA, USA) *(UIST*

'08). Association for Computing Machinery, New York, NY, USA, 3–12. https://doi.org/10.1145/1449715.1449719

[16] David Grimes, Desney S Tan, Scott E Hudson, Pradeep Shenoy, and Rajesh PN Rao. 2008. Feasibility and pragmatics of classifying working memory load with an electroencephalograph. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 835–844.

[17] Uwe Gruenefeld, Dag Ennenga, Abdallah El Ali, Wilko Heuten, and Susanne Boll. 2017. EyeSee360: designing a visualization technique for out-of-view objects in head-mounted augmented reality. In *Proceedings of the 5th Symposium on Spatial User Interaction* (Brighton, United Kingdom) *(SUI '17)*. Association for Computing Machinery, New York, NY, USA, 109–118. https://doi.org/10.1145/3131277.3132175

[18] Uwe Gruenefeld, Ilja Koethe, Daniel Lange, Sebastian Weiß, and Wilko Heuten. 2019. Comparing Techniques for Visualizing Moving Out-of-View Objects in Head-mounted Virtual Reality. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 742–746. https://doi.org/10.1109/VR.2019.8797725

[19] Gaoping Huang, Xun Qian, Tianyi Wang, Fagun Patel, Maitreya Sreeram, Yuanzhi Cao, Karthik Ramani, and Alexander J Quinn. 2021. AdapTutAR: An Adaptive Tutoring System for Machine Tasks in Augmented Reality. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21, Article 417)*. Association for Computing Machinery, New York, NY, USA, 1–15. https://doi.org/10.1145/3411764.3445283

[20] Haikun Huang, Michael Solah, Dingzeyu Li, and Lap-Fai Yu. 2019. Audible Panorama: Automatic Spatial Audio Generation for Panorama Imagery. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19, Paper 621)*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3290605.3300851

[21] Andrew Irlitti, Ross T Smith, Stewart Von Itzstein, Mark Billinghurst, and Bruce H Thomas. 2016. Challenges for Asynchronous Collaboration in Augmented Reality. In *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*. 31–35. https://doi.org/10.1109/ISMAR-Adjunct.2016.0032

[22] JASP Team. 2021. JASP (Version 0.16.0)[Computer software]. https://jasp-stats.org/

[23] Thorsten Karrer, Malte Weiss, Eric Lee, and Jan Borchers. 2008. DRAGON: A Direct Manipulation Interface for Frame-Accurate in-Scene Video Navigation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy) *(CHI '08)*. Association for Computing Machinery, New York, NY, USA, 247–250. https://doi.org/10.1145/1357054.1357097

[24] Shunichi Kasahara and Jun Rekimoto. 2014. JackIn: integrating first-person view with out-of-body vision generation for human-human augmentation. In *Proceedings of the 5th augmented human international conference*. 1–8.

[25] Rubaiat Habib Kazi, Tovi Grossman, Cory Mogk, Ryan Schmidt, and George Fitzmaurice. 2016. ChronoFab: fabricating motion. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 908–918.

[26] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. 2023. Segment Anything. arXiv:2304.02643 [cs.CV]

[27] Pascal Knierim, Thomas Kosch, Gabrielle LaBorwit, and Albrecht Schmidt. 2020. Altering the Speed of Reality? Exploring Visual Slow-Motion to Amplify Human Perception using Augmented Reality. In *Proceedings of the Augmented Humans International Conference* (Kaiserslautern, Germany) *(AHs '20, Article 2)*. Association for Computing Machinery, New York, NY, USA, 1–5. https://doi.org/10.1145/3384657.3384659

[28] Ryohei Komiyama, Takashi Miyaki, and Jun Rekimoto. 2017. JackIn space: designing a seamless transition between first and third person view for effective telepresence collaborations. In *Proceedings of the 8th Augmented Human International Conference* (Silicon Valley, California, USA) *(AH '17, Article 14)*. Association for Computing Machinery, New York, NY, USA, 1–9. https://doi.org/10.1145/3041164.3041183

[29] Johannes Kopf, Xuejian Rong, and Jia-Bin Huang. 2021. Robust consistent video depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1611–1621.

[30] Gun A Lee, Seungjun Ahn, William Hoff, and Mark Billinghurst. 2020. Enhancing First-Person View Task Instruction Videos with Augmented Reality Cues. In *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 498–508. https://doi.org/10.1109/ISMAR50242.2020.00078

[31] Klemen Lilija, Henning Pohl, and Kasper Hornbæk. 2020. Who Put That There? Temporal Navigation of Spatial Recordings by Direct Manipulation. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3313831.3376604

[32] David Lindlbauer, Anna Maria Feit, and Otmar Hilliges. 2019. Context-Aware Online Adaptation of Mixed Reality Interfaces. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 147–160. https://doi.org/10.1145/3332165.3347945

[33] David Lindlbauer, Klemen Lilija, Robert Walter, and Jörg Müller. 2016. Influence of Display Transparency on Background Awareness and Task Performance. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 1705–1716. https://doi.org/10.1145/2858036.2858453

[34] David Lindlbauer and Andrew D Wilson. 2018. Remixed reality: manipulating space and time in augmented reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–13.

[35] Sean J. Liu, Maneesh Agrawala, Stephen DiVerdi, and Aaron Hertzmann. 2019. View-Dependent Video Textures for 360° Video. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 249–262. https://doi.org/10.1145/3332165.3347887

[36] David G Lowe. 1999. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, Vol. 2. 1150–1157 vol.2. https://doi.org/10.1109/ICCV.1999.790410

[37] Bruce D Lucas and Takeo Kanade. 1981. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th international joint conference on Artificial intelligence - Volume 2* (Vancouver, BC, Canada) *(IJCAI'81)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 674–679. https://dl.acm.org/doi/10.5555/1623264.1623280

[38] Zhaoyang Lv, Edward Miller, Jeff Meissner, Luis Pesqueira, Chris Sweeney, Jing Dong, Lingni Ma, Pratik Patel, Pierre Moulon, Kiran Somasundaram, Omkar Parkhi, Yuyang Zou, Nikhil Raina, Steve Saarinen, Yusuf M Mansour, Po-Kang Huang, Zijian Wang, Anton Troynikov, Raul Mur Artal, Daniel DeTone, Daniel Barnes, Elizabeth Argall, Andrey Lobanovskiy, David Jaeyun Kim, Philippe Bouttefroy, Julian Straub, Jakob Julian Engel, Prince Gupta, Mingfei Yan, Renzo De Nardi, and Richard Newcombe. 2022. Aria Pilot Dataset. https://about.facebook.com/realitylabs/projectaria/datasets.

[39] Karthik Mahadevan, Qian Zhou, George Fitzmaurice, Tovi Grossman, and Fraser Anderson. 2023. Tesseract: Querying Spatial Design Recordings by Manipulating Worlds in Miniature. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–16.

[40] Kyle Min and Jason J Corso. 2019. Tased-net: Temporally-aggregating spatial encoder-decoder network for video saliency detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2394–2403.

[41] Takashi Miyaki and Jun Rekimoto. 2016. Lidarman: Reprogramming reality with egocentric laser depth scanning. In *ACM SIGGRAPH 2016 Emerging Technologies*. 1–2.

[42] Florian Floyd Mueller, Pedro Lopes, Paul Strohmeier, Wendy Ju, Caitlyn Seim, Martin Weigel, Suranga Nanayakkara, Marianna Obrist, Zhuying Li, Joseph Delfa, Jun Nishida, Elizabeth M Gerber, Dag Svanaes, Jonathan Grudin, Stefan Greuter, Kai Kunze, Thomas Erickson, Steven Greenspan, Masahiko Inami, Joe Marshall, Harald Reiterer, Katrin Wolf, Jochen Meyer, Thecla Schiphorst, Dakuo Wang, and Pattie Maes. 2020. Next Steps for Human-Computer Integration. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–15. https://doi.org/10.1145/3313831.3376242

[43] Cuong Nguyen, Yuzhen Niu, and Feng Liu. 2013. Direct Manipulation Video Navigation in 3D. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) *(CHI '13)*. Association for Computing Machinery, New York, NY, USA, 1169–1172. https://doi.org/10.1145/2470654.2466150

[44] Masaki Oshita. 2019. Motion Volume: Visualization of Human Motion Manifolds. In *The 17th International Conference on Virtual-Reality Continuum and its Applications in Industry*. 1–7.

[45] Amy Pavel, Björn Hartmann, and Maneesh Agrawala. 2017. Shot Orientation Controls for Interactive Cinematography with 360 Video. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* (Québec City, QC, Canada) *(UIST '17)*. Association for Computing Machinery, New York, NY, USA, 289–297. https://doi.org/10.1145/3126594.3126636

[46] Ken Pfeuffer, Yasmeen Abdrabou, Augusto Esteves, Radiah Rivu, Yomna Abdelrahman, Stefanie Meitner, Amr Saadi, and Florian Alt. 2021. ARtention: A design space for gaze-adaptive user interfaces in augmented reality. *Computers & graphics* 95 (April 2021), 1–12. https://doi.org/10.1016/j.cag.2021.01.001

[47] Kathrin Probst, David Lindlbauer, Michael Haller, Bernhard Schwartz, and Andreas Schrempf. 2014. A Chair as Ubiquitous Input Device: Exploring Semaphoric Chair Gestures for Focused and Peripheral Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 4097–4106. https://doi.org/10.1145/2556288.2557051

[48] Jun Rekimoto. 1997. Navicam: A magnifying glass approach to augmented reality. *Presence: Teleoper. Virtual Environ.* 6, 4 (Aug. 1997), 399–412. https://doi.org/10.1162/pres.1997.6.4.399

[49] Mrigank Rochan, Mahesh Kumar Krishna Reddy, Linwei Ye, and Yang Wang. 2020. Adaptive video highlight detection by learning from user history. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*. Springer, 261–278.

[50] Arno Schödl, Richard Szeliski, David H. Salesin, and Irfan Essa. 2000. Video Textures. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '00)*. ACM Press/Addison-Wesley Publishing Co., USA, 489–498. https://doi.org/10.1145/344779.345012

[51] Eldon Schoop, James Smith, and Bjoern Hartmann. 2018. Hindsight: enhancing spatial awareness by sonifying detected objects in real-time 360-degree video. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–12.

[52] Jianbo Shi and Tomasi. 1994. Good features to track. In *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 593–600. https://doi.org/10.1109/CVPR.1994.323794

[53] Maximilian Speicher, Jingchen Cao, Ao Yu, Haihua Zhang, and Michael Nebeling. 2018. 360anywhere: Mobile ad-hoc collaboration in any environment using 360 video and augmented reality. *Proceedings of the ACM on Human-Computer Interaction* 2, EICS (2018), 1–20.

[54] Ryo Suzuki, Rubaiat Habib Kazi, Li-Yi Wei, Stephen DiVerdi, Wilmot Li, and Daniel Leithinger. 2020. RealitySketch: Embedding Responsive Graphics and Visualizations in AR through Dynamic Sketching. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. Association for Computing Machinery, New York, NY, USA, 166–181. https://doi.org/10.1145/3379337.3415892
[55] Eduardo E Veas, Erick Mendez, Steven K Feiner, and Dieter Schmalstieg. 2011. Directing attention and influencing memory with visual saliency modulation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI '11)*. Association for Computing Machinery, New York, NY, USA, 1471–1480. https://doi.org/10.1145/1978942.1979158
[56] Elliot B Werner. 1991. *Manual of visual fields*. Churchill Livingstone.
[57] Barbara M. Wildemuth, Gary Marchionini, Meng Yang, Gary Geisler, Todd Wilkens, Anthony Hughes, and Richard Gruss. 2003. How Fast is Too Fast? Evaluating Fast Forward Surrogates for Digital Video. In *Proceedings of the 3rd ACM/IEEE-CS Joint Conference on Digital Libraries* (Houston, Texas) *(JCDL '03)*. IEEE Computer Society, USA, 221–230.
[58] Chao-Yuan Wu, Justin Johnson, Jitendra Malik, Christoph Feichtenhofer, and Georgia Gkioxari. 2023. Multiview Compressive Coding for 3D Reconstruction. *arXiv:2301.08247* (2023).
[59] Xiuming Zhang, Tali Dekel, Tianfan Xue, Andrew Owens, Qiurui He, Jiajun Wu, Stefanie Mueller, and William T Freeman. 2018. Mosculp: Interactive visualization of shape and time. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. 275–285.
[60] Yuhang Zhao, Sarit Szpiro, and Shiri Azenkot. 2015. ForeSee: A Customizable Head-Mounted Vision Enhancement System for People with Low Vision. In *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility* (Lisbon, Portugal) *(ASSETS '15)*. Association for Computing Machinery, New York, NY, USA, 239–249. https://doi.org/10.1145/2700648.2809865
[61] Yuhang Zhao, Sarit Szpiro, Lei Shi, and Shiri Azenkot. 2019. Designing and Evaluating a Customizable Head-mounted Vision Enhancement System for People with Low Vision. *ACM transactions on accessible computing* 12, 4 (Dec. 2019), 1–46. https://doi.org/10.1145/3361866
[62] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. 2021. Detecting Twenty-thousand Classes using Image-level Supervision. In *arXiv preprint arXiv:2201.02605*.

A APPENDIX

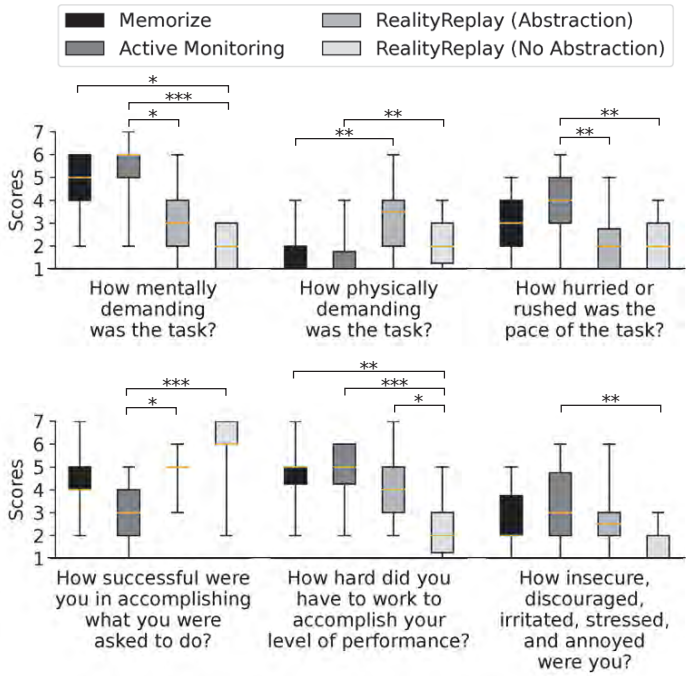

Fig. 12. Rating from the post-experiment NASA Task Load Index (TLX) questionnaires from 1 (low) to 7 (high). The boxplot shows median as the orange line, interquartile range (IQR) as the box, and minimum and maximum values as the whiskers.

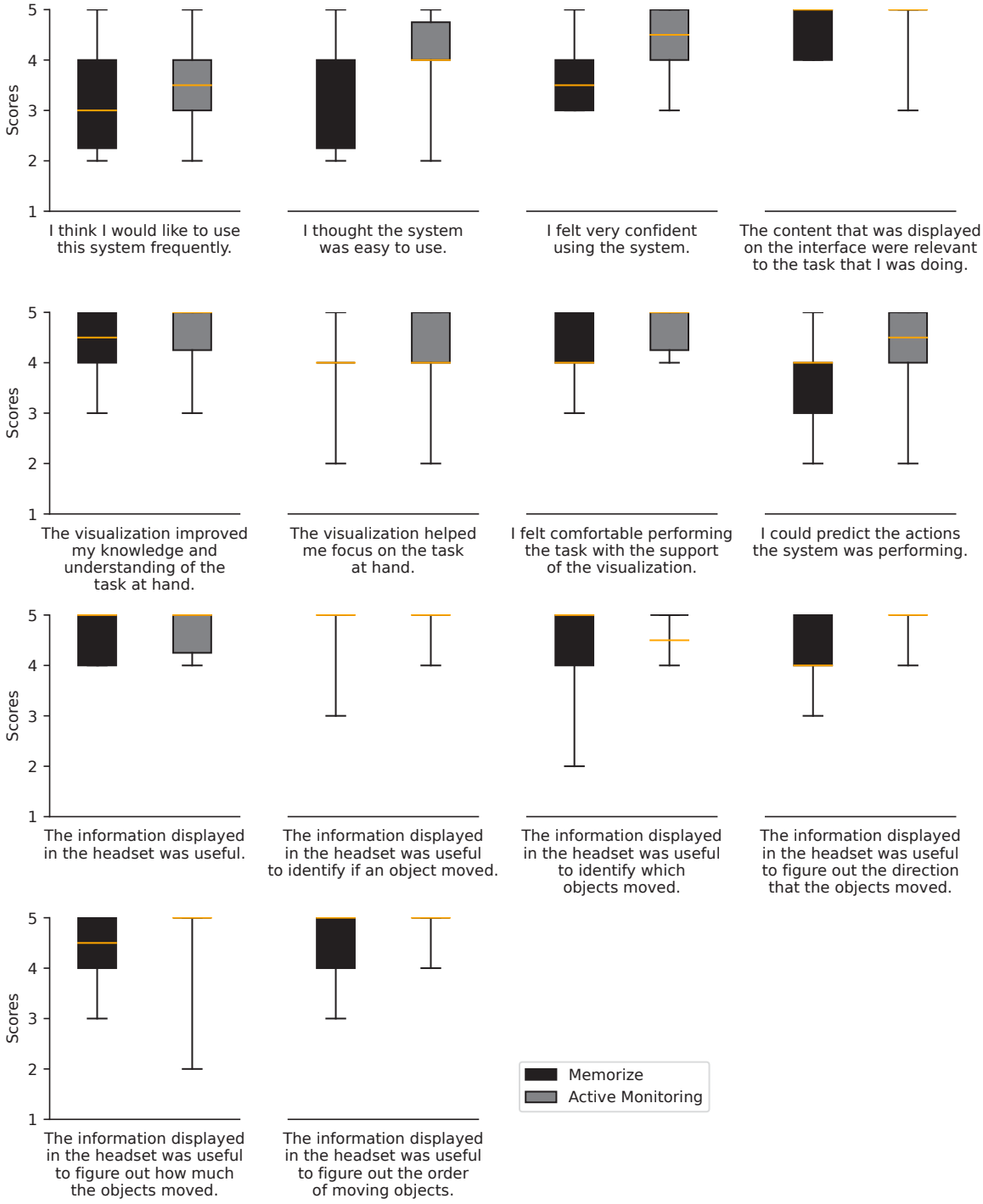

Fig. 13. Rating from the post-experiment visualization usability questionnaires from 1 (low) to 5 (high). The boxplot shows median as the orange line, interquartile range (IQR) as the box, and minimum and maximum values as the whiskers.