International Conference on Artificial Reality and Telexistence
Eurographics Symposium on Virtual Environments (2015)
M. Imura, P. Figueroa, and B. Mohler (Editors)

How to Time Travel in Highly Immersive Virtual Reality

Rodrigo Pizarro[1], Keren-Or Berkers[4], Mel Slater[1,2,3], and Doron Friedman[4]

[1]Event Lab, Faculty of Psychology, University of Barcelona, Spain
[2]Institució Catalana de Recerca i Estudis Avançats (ICREA), Barcelona, Spain
[3] Department of Computer Science, University College London, UK
[4]The Interdisciplinary Center, Herzliya

Abstract

What would it be like to be able to time travel to the past, meet your previous self, and override your previous actions in order to achieve a better outcome? While we cannot (yet?) achieve physical time travel, digital technologies now allow us to experience virtual time travel. We have developed a method for implementing time travel in highly immersive virtual reality (VR) and here we describe the underlying technology in the context of a scenario that involves a shooting event in a virtual gallery. Our method includes two layers of abstraction: i) a narrative layer that represents scenarios as a set of events and state transitions, and uses preconditions to enforce consistency, and ii) a VR layer that includes low level controllers for low level synchronization and animation. The narrative layer is designed to ensure that following time travel the events would unfold exactly as they did in the previous time around, except for the specific changes resulting from the actions of the time traveling participant. The VR layer controls the fine details, including recording and replaying motion capture data and audio, which allows the participants to experience their own previous selves as animated avatars. The system was used for a psychological experiment, and in this paper we focus on the technical method and on the lessons learned from implementing VR time travel.

Categories and Subject Descriptors (according to ACM CCS): I.3 [Computer Graphics]: Three-dimensional graphics and realism—Virtual reality

---

1. Introduction

Time travel has long been a topic of interest for fiction, and there are numerous stories and films in which protagonists travel to the past in order to change a future outcome. Outside science fiction, traveling through time at a human-sized scale has not yet proven possible. Immersive virtual reality (IVR), on the other hand, has allowed researchers to study situations that would be impossible or unacceptable in the physical world. VR, and especially highly-immersive VR, goes beyond literature or movies by actually providing people with visceral, interactive experiences. IVR utilizes three types of illusions: presence, defined as the illusion of being in the place depicted by the virtual environment [VS05, Sla09], body ownership –the illusion that the virtual body seen is one's body [Ehr07, SSSV*10], and agency (for example, as in [BGS13]), where the participant has the sensation of being the cause of the movements of his or her virtual body. These illusions, especially when combined, have the potential to alter people's mindset, not only while being immersed in the IVR, but also later on in the physical world. In this paper, we describe how to implement another type of illusion in IVR, the time travel illusion. This illusion as well as the experimental details are described in [FPOB*14].

2. Related Work

Like other IVR illusions, time travel may have applications beyond entertainment. In basic science, there has been growing interest in “mental time travel”, or the ability of humans to mentally reconstruct personal events from the past (episodic memory) and to mentally construct possible events in the future [BDM08, SC07]. Confronting your past (or future) may have psychological or even psychotherpeutic consequences and applications; for example, meeting your older self in virtual reality was found to increase your saving behavior [HGS*11]. Reliving past experiences in VR , sometimes traumatic, is often part of virtual reality exposure ther-

DOI: 10.2312/egve.20151318

apy (VRET) [PR08], and 'meeting your previous self' could be a potential extension to such exposures.

In [FPOB*14] we described an experiment where participants in one condition experienced the illusion of time travel. Participants had played an important part in events with a tragic outcome – deaths of strangers – by having to choose between saving 5 people or 1. We studied whether the ability to go back through time, and intervene, to possibly avoid all deaths, had an impact on how participants would view such moral dilemmas, and also whether this experience would lead to a re-evaluation of past unfortunate events in their own lives. In the "Time Travel" condition 16 participants "relived" the same events three times, seeing incarnations of their past selves carrying out the actions that they had previously carried out (Figure 1), and were able to use their present time embodiment to override their past decisions. In this paper we describe the underlying technology necessary for implementing this type of VR time travel.

Figure 1: *A snapshot of the virtual gallery experiment. The participant is embodied in a virtual avatar shown in the bottom right. The avatar standing next to it performs a repetition of the actions taken by the participant in the previous time around.*

While time travel has not yet been addressed by the virtual reality community, similar challenges in modeling time and causality have been addressed by the collaborative virtual environment (CVE) community; in many applications, such as shooter games, network latency is prohibitive, so a simple approach using a centralized server and dumb clients is unacceptable. Various methods have been suggested to cope with this challenge, and these include some analysis of causality [RS97, SRR98, SO09]. Another relevant thread of research is automated reasoning about digital narrative. While such reasoning was not used for time travel, it has been able to address similar challenges involving narrative consistency, mostly by using artificial intelligence techniques such as planning [YR03, ML06, RY06].

3. The Scenario

The specific scenario we created takes place in an art gallery on two levels (ground and upper), and is based on earlier research on action in response to a moral dilemma in IVR [PS11]. The participant learns to operate a virtual elevator that takes (virtual human) visitors to the upper level or down from upper level to the ground level at their request, and also learns to operate an alarm that freezes the elevator in place and triggers an alarm sound. After six visitors have entered the gallery there are five people browsing the paintings upstairs and one person at the ground level, and then a seventh person enters the gallery and asks to be taken to the upper level. Upon arrival, and while still on the elevator platform, he pulls out a gun out of his pocket and starts shooting at the five people on the upper level, as shown in Figure 2.

Figure 2: *A snapshot of the shooting event.*

The participant has also previously learned that pressing an alarm button will immediately freeze the elevator in place (but this is no use at this moment since the gunman is already shooting). After a few seconds of this mayhem the scene dissolves and the participant is back at the start of the whole sequence of events. However, unlike typical IVR scenarios that you can play over and over, the scenario can also be repeated as 'time travel'. Following time travel the participant controls a new avatar, standing slightly behind the original avatar, which has now become a clone controlled by a software agent. The participant can observe the events unfold from this vantage point, and they would replay exactly as they did in the previous time around. However, the participant can also decide to intervene and 'change history', by pressing the elevator buttons. If that happens our narrative engine tries to playback events as close as possible to the first time around, but also taking into account the actions of the participant in the second time around. This is done using a mechanism for tracking the causality behind actions. A companion video describes the possible unfolding of events[†].

We define two levels of abstraction: the narrative layer and the VR layer. The narrative layer deals with the unfolding of the narrative at the level of states and events, and the VR layer has a much richer description of the virtual world, which is required to enable the immersive experience of virtual time travel. The two layers communicate using high level controllers as explained below.

[†] `https://goo.gl/VGpRmJ`

4. The Narrative Engine

The conceptual building blocks in our model are entities, history, states, and events. There are three types of entities, which are different in terms of the way they are simulated:

- `Dynamic objects` are assumed to follow deterministic behavior.
- `Agents` are simulated human beings (or, in principle, objects that are assumed to have agency and trigger events).
- `The participant` is free to interact with the system, like in any VR system. If the participant goes back in time, he or she may meet previous clones of herself. These previous clones are now agents; they are controlled by software and their behavior is based on a recording of the participant's actions in the previous time around, as described below.

All entities have states, and each type of entity has its own state variables: these are application dependent, although generic objects that can be shared across applications are possible. The entities in the elevator scenario are the visitors (including the gunman), the participant, the elevator, and the alarm. The entities have state variables such as their location. The possible events are the following, according to the entities that perform them:

- Visitor: Enter gallery, Enter elevator, Wait in elevator, Exit elevator, Watch paintings
- Gunman (same as visitor, and also): Shoot, Pull gun, Hide gun
- Participant and his clones: Press Up, Press Down, Press Alarm
- Elevator: Start going up, Start going down, Stop
- Alarm: Toggle (on/off)

A history h is a set $\{s_0, E\}$, where s_0 includes the initial state of all entities and E is a sequence of events. In the narrative layer all events are assumed to be instantaneous, so each event e is associated with a point in time t_e, even though in the VR layer many of the events have a duration. For example, consider the event `Enter elevator` – in the narrative layer we keep the start time, but in the VR layer the corresponding animation sequence lasts a few seconds. To handle such events, in the narrative layer they are split into a start event and a stop event. In the case of entering the elevator the start event is `Enter elevator` and the stop event is `Wait in elevator`. There are no concurrent events and the state at each moment during the history can be computed by deterministically applying the events in the order they take place, starting from s_0.

Each event e is associated with a set of preconditions $C_{e,pre}$ and a set of post-conditions $C_{e,post}$. Each condition C is a tuple $< n, r, v >$, where n is an entity, r is a state variable of that entity and v is one of the possible values that this variable can accept. If $< n, r, v >$ is a precondition of event e then e can only take place if at time t_e, before applying e, the value of the variable r of entity n is equal to v. If $< n, r, v >$ is a post-condition of e then after applying e the value of variable r of entity n is set to v.

The history h includes three types of events, which correspond to the three types of entities: i) events that happen to inanimate objects, ii) events triggered by autonomous agents, and iii) events triggered by the participant. The participant is always free to take actions that trigger events at any time (pressing the elevator and alarm buttons). Events taken by objects are not required to be kept in the history, since they are deterministically derived by the other two classes of events. Similarly, the entity states are maintained in the history, although they can be deterministically derived from the events. We keep object events in the history because it is easier for the human programmer to inspect the history with these events appearing at the right times. We keep the updated state information after each event because this allows quickly resetting the environment state after time travel to any arbitrary point in time.

Some events triggered by agents do not play a role in the reasoning about the narrative but are required by the VR engine. For example, the VR engine needs to render the gunman pulling the gun out before shooting and hiding it afterwards. One option is to consider this a low level detail, which does not need to be represented in the narrative engine. The problem is that the VR engine, by design, makes no attempt to predict future events, and the animation for pulling out the gun needs to happen before shooting. We have thus opted to include pulling out the gun and hiding it in the narrative layer, even though it is not required for reasoning about the narrative.

The objects are based on simple state machines. They never initiate events, and always respond deterministically based on their current state and external events. For example, the alarm works as a simple on/off toggle, and the elevator can travel between the two floors and is blocked when the alarm is on.

At the beginning of the first execution, the history h is empty, since nothing has happened yet. The system includes a scripted narrative with timed events, e.g., specifying that Visitor 1 enters the room 20 seconds after the session started. In addition, the history records resulting object events, such as the elevator doors opening, and the elevator going up or down, as a result of the participant's button presses, as well as the actions taken by the participant. Five of the visitors are scripted to go to the top floor and watch the paintings, another visitor is scripted to watch the paintings on the ground floor, and the gunman is scripted to take the elevator to the top floor and shoot everyone there. An example of the corresponding history (sequence of timed events) appears in Table 1. For brevity we only show a subset of the history and skip to the part when all visitors are already in the gallery and the gunman steps in.

Now let us consider a simple scenario in the second time around: the participant goes back to time t. All the entities

Time	Entity	Event
125.4	Participant	press down
125.4	Elevator	start moving
132.4	Elevator	stop moving
132.4	Shooter	enter elevator
132.4	Elevator	open doors
134.7	Elevator	close doors
137.5	Participant	press up
137.5	Elevator	start moving
143.6	Shooter	pull gun
144.5	Elevator	stop moving
145.7	Shooter	shoot V5
147.7	Shooter	shoot V1
149.7	Shooter	shoot V0
150.6	Participant	press A
151.6	Shooter	shoot V4
153.7	Shooter	shoot V3
155.7	Shooter	hide gun

Table 1: *Execution example of a subset of the first time around. The first column specifies the time from the beginning of the session, in seconds. The second column specifies the entity responsible for the event and the third column specifies the event and its parameters.*

are reset to their state at time t in the first time around, and the participant from the first time around now becomes an agent. The playback of scripted narrative is disabled and the narrative engine drives the execution using based on the history recorded in the first time around.

The algorithm that deals with the replay of the narrative after time travel is as follows: at each frame the update loop handles all events that need to take place at that point in time. These events come either from the history to be replayed or from the VR (events triggered by the participant), and these need to be merged into one consistent narrative. The algorithm distinguishes among the three types of events, corresponding to the three classes of entities:

- If the event was triggered by the `participant` then it takes place in any case and is recorded in the history structure for the next time around.
- If the event is a result of a simulation of an `object` then it behaves according to the state machine of that object.
- If the event is triggered by an `agent` (including previous clones of the participant) then it is executed only if its preconditions are met. If the preconditions are violated, the event may be replaced, as explained below, or it is discarded.

Table 2 provides the details of the new history resulting from the participant's interventions after time travel. For convenience we display this history next to the original one from the first time around. The first time around is replayed and unfolds exactly as it did in the first time around, until the participant (denoted as Participant_1) intervenes: at time 133.8 the participant, who now recognizes the gunman and anticipates trouble, presses the alarm. The gunman is in the elevator, but now the elevator is disabled because of the alarm; this is reflected in the elevator's state variables.

At time 137.5, just like the first time around, the participant's previous clone presses the up button. However, since the elevator is now disabled, it does not go up – the object events at times 137.5 and 144.5 do not take place in this round, and this is marked by blank lines in the history, for readability. At 143.6 the gunman pulls out the gun. This is the same as in the previous time around, but this time the gunman is stuck in the ground floor, whereas in the first time around he was approaching the upper floor.

At 145.7 the engine tries to replay the recorded event – shooting visitor v5. However, the preconditions of that event are violated: unlike the first time around, the gunman and v5 are now not in the same floor. If a recorded event's preconditions are violated, the engine tries to replace that event. In our current version replacement takes place for two events: i) shooting, and ii) visiting a certain floor in the gallery. The engine is able to replace the shooting event by replacing v5 by v2, since v2 is in the same floor as the gunman (the elevator is open and allows shooting, even though the gunman is inside). The other shooting events are ignored, since there are no more targets in the ground level (in this scenario we prevented the gunman from shooting the participant or her clones). Similar replacement of event parameters is supported for the visitors' event of browsing the paintings in the gallery. The narrative requires that five of the visitors would spend time in the top floor. However, if the participant prevents one or more of these visitors from reaching the top floor by not sending the elevator, they will proceed to explore the ground floor, and might be shot as well, if the shooting takes place in the ground floor.

The other events are replayed as they took place in the first time around: the participant's previous clone presses the alarm, at 150.6. In this time around the alarm is turned off, whereas the previous time around this turned it on, but this has no other effect on the unfolding of events.

There are two possible solutions to avoid the shooting. A simple solution (Table 3) is to stop the elevator half way with the shooter inside. Alternatively, a participant could prevent visitors from taking the elevator, which would cause them to stay in the ground level. The shooter would take the elevator to find that there are no targets in the top level (Table 4).

4.0.1. Proactive Agents

The approach described so far treats the agents after time travel as "dumb" agents, who just carry on doing what they did in the previous time around, as long as preconditions are met; they do not respond to the new unfolding narrative. In this scenario we have provided a mechanism for overcom-

1st-time around			2nd-time around		
112.4	Shooter	enter floor	112.4	Shooter	enter floor
125.4	Participant	press down	125.4	Participant	press down
125.4	Elevator	start moving	125.4	Elevator	start moving
132.4	Elevator	stop moving	132.4	Elevator	stop moving
132.4	Shooter	enter elevator	132.4	Shooter	enter elevator
132.4	Elevator	open doors	132.4	Elevator	open doors
			133.8	Participant_1	press A
134.7	Elevator	close doors	134.7	Elevator	close doors
137.5	Participant	press up	137.5	Participant	press up
137.5	Elevator	start moving			
143.6	Shooter	pull gun	143.6	Shooter	pull gun
144.5	Elevator	stop moving			
145.7	Shooter	shoot v5	145.7	Shooter	shoot v2
147.7	Shooter	shoot v1			
149.7	Shooter	shoot v0			
150.6	Participant	press A	150.6	Participant	press A
151.6	Shooter	shoot v4			
153.7	Shooter	shoot v3			
155.7	Shooter	hide gun	155.7	shooter	hide gun

Table 2: *Events in the first time around (left) and the modified history in the second time around (right).*

1st-time around			2nd-time around		
112.4	Shooter	enter floor	112.4	Shooter	enter floor
125.4	Participant	press down	125.4	Participant	press down
125.4	Elevator	start moving	125.4	Elevator	start moving
132.4	Elevator	stop moving	132.4	Elevator	stop moving
132.4	Shooter	enter elevator	132.4	Shooter	enter elevator
132.4	Elevator	open doors	132.4	Elevator	open doors
134.7	Elevator	close doors	134.7	Elevator	close doors
137.5	Participant	press up	137.5	Participant	press up
137.5	Elevator	start moving	137.5	Elevator	start moving
			140.0	Participant_1	press A
			140.8	Elevator	stop moving
143.6	Shooter	pull gun	143.6	Shooter	pull gun
144.5	Elevator	stop moving			
145.7	Shooter	shoot v5			
147.7	Shooter	shoot v1			
149.7	Shooter	shoot v0			
150.6	Participant	press A	150.6	Participant	press A
151.6	Shooter	shoot v4			
153.7	Shooter	shoot v3			
155.7	Shooter	hide gun	155.7	shooter	hide gun

Table 3: *A 'solution': preventing the shooting by getting the gunman caught in between floors. Left: events in the first time around, right: the modified history in the second time around.*

ing this limitation – the automatic replacement of event parameters. This mechanism allows the agents with plausible behaviour following time travel, but it is limited, since it allows ignoring events or replacing them, but not adding new events. Sometimes we expect humans to do something, and failing to act can be considered not plausible. Moreover, this is also true of the participant's clone. For example, if in the second time around the participant triggers the alarm before any shooting actually happened (Table 4), then we would expect some reaction from the participant's clone, which repre-

1st-time around			2nd-time around		
			95.0	Participant_1	press up
			95.0	Elevator	start moving
102.0	Elevator	stop moving			
100.3	V1,V2,V3,V4,V5	enter elevator			
102.0	V1,V2,V3,V4,V5	wait in elevator			
102.5	Participant	press up	137.5	Participant	press up
102.5	Elevator	start moving			
109.5	Elevator	stop moving			
109.5	V1,V2,V3,V4,V5	exit elevator			
111.0	V1,V2,V3,V4,V5	visit upper floor	111.0	V1,V2,V3,V4,V5	visit ground floor
112.4	Shooter	enter floor	112.4	Shooter	enter floor
125.4	Participant	press down	125.4	Participant	press down
125.4	Elevator	start moving	125.4	Elevator	start moving
132.4	Elevator	stop moving	132.4	Elevator	stop moving
132.4	Shooter	enter elevator	132.4	Shooter	enter elevator
132.4	Elevator	open doors	132.4	Elevator	open doors
134.7	Elevator	close doors	134.7	Elevator	close doors
137.5	Participant	press up	137.5	Participant	press up
137.5	Elevator	start moving	137.5	Elevator	start moving
143.6	Shooter	pull gun	143.6	Shooter	pull gun
144.5	Elevator	stop moving			
145.7	Shooter	shoot v5			
147.7	Shooter	shoot v1			
149.7	Shooter	shoot v0			
150.6	Participant	press A	150.6	Participant	press A
151.6	Shooter	shoot v4			
153.7	Shooter	shoot v3			
155.7	Shooter	hide gun	155.7	shooter	hide gun

Table 4: *Alternative 'solution': preventing the shooting by keeping all visitors in the ground level.*

sents the participant before he knew that the gunman is about to shoot.

Given that the participant‘s real behaviour is unpredictable, achieving a realistic behaviour of the previous clones involves a major challenge, which is, in general, beyond the current state of the art in artificial intelligence. One option is to use "behavioural cloning" (e.g., [FT11]): the participant is recorded, a behavioural model is extracted, and this behavioural model is then used in the second time around. Such approaches are beyond the scope of this paper.

Regardless of the algorithm that controls the clones, the system records certain actions in the introduction or first time around, which can then be replayed "on demand" by the algorithm. For example, in our experiment participants were instructed to react to alarms triggered without apparent reason by turning the alarm off and saying "False alarm". During the second time around, if a participant pressed the alarm without any apparent reason from the clone's perspective, the clone could be programmed to use the recorded motion and voice to say "False alarm" and turn the alarm off..

5. The IVR Engine

In order to provide an immersive experience of time travel, much more detail is required; the IVR engine needs to replay dynamic information with a temporal resolution and accuracy that is significantly higher than that required by the narrative engine.

The first task of the IVR engine is to induce in participants the most common illusions in IVR, which are presence, body ownership, and agency. If this task cannot be achieved, it is highly unlikely that participants would experience a time travel illusion, since if they do not recognize their movements as their own, they would not recognize their past movements as such either. In order to induce the illusions, we used the HUMAN software library [SNN*13] to correlate participants' real body movements with virtual ones, and Unity3D to implement a believable scenario.

The interface between the two layers of abstraction takes place at the state machine level for all entities. In addition, a low level controller (LLC) per entity is coupled with this state machine, and deals with the specific low level rendering details of the entity (e.g., animation blending or linear displacement of objects). Additionally, some events (e.g., button presses) may be triggered by participants at a low level of abstraction, using the physics in the rendering engine, and that needs to be translated into a high level event. Furthermore, the IVR scenario includes interaction among dynamic entities (e.g., the alarm blocks the elevator).

In order to control all the interactions between entities we developed a centralized events routing system to ensure that events are directed to the proper state machines (Figure 3). Each entity subscribes to the router with the events that it needs to respond to, while LLCs only subscribe to the relevant events in their corresponding state machine. The router also sends all events coming from state machines to the narrative engine. The latter checks the preconditions, as described earlier, and if the event is accepted the engine sends them back to the router, which then broadcasts the events to its subscribed state machines. It is the responsibility of the programmer to avoid possible loops in event routing.

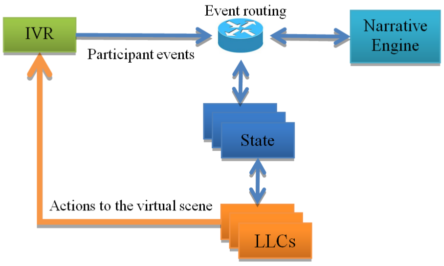

Figure 3: *Diagram of the event routing process.*

5.1. Recording and Playback

We used the HUMAN library [SNN*13] for body tracking and VRPN [TIHS*01] for head tracking. During each time around the system records and stores all the tracking data for each frame. During the second time around, the participants see a different avatar performing the same actions that they had just performed a few minutes ago; this is achieved by reloading and replaying the stored motion capture files. Given that the frame rate remained constant at 60 frames per second throughout the entire experience, the playback was always accurate.

It was also critical to avoid virtual spatial collision between the participant's avatar and the avatars that represented their past selves. If that happened, the participant would get the confusing illusion that he or she is inside two avatars. To solve that problem, we stored the participant's position at the beginning of the first time around. When the second time around started, the new avatar was created in the background and a translation offset was applied to it.

5.2. Playback Synchronization

The execution is driven by events coming from two sources in parallel: the history replay and the current VR scene. Discrepancies in reproduction may occur due to imprecisions in timing and floating point precision; e.g., a visitor enters the gallery with the left foot in the first time around but with the right foot in the second time around. The latter example can be explained by discrepancies in playback timing; such gaps of up to approximately 16 milliseconds (a frame time at 60 frames per second) are common in VR and game engines. In this example, if in the second time around the state machine of the visitor received the `Enter Gallery` event with a delay of a few milliseconds relative to the first time around, and the calculated path varied a few virtual centimeters due to floating precision errors, the walking animation cycle could be in a different state by the time the avatar reached the gallery. It is highly unlikely, however, that those discrepancies are noticeable, and in our experiment none of the subjects referred to such issues during their debriefing.

The main challenge is to prevent possible accumulation of error over time. To address this, we used a conservative method, similar to what is known as lockstep synchronization [SO09] in the network computing community. In this approach clients do not immediately react to new events. Instead, they send the new events to remote clients and wait for a response before they locally apply the event. This method is simple and is considered a good practice for latencies below 75ms. There is a direct analogy between this multi-user framework and our time travel IVR engine, where the local clients are mapped to state machines are and a remote client is mapped to the narrative engine. Since both the IVR engine and the narrative engine were executed on the same computer and all events were processed in a per-frame basis, the latency never exceeded 1/60 seconds.

This strategy, however, is not valid for all entities. In particular, because the elevator moves at a constant predefined speed and stops when it receives the `Stop Moving` event, accumulated drifts of fractions of a second can lead to noticeable position discrepancies over time. The elevator travels at a constant speed of 0.64 m/s, which translates to approximately one virtual centimeter every frame. Since it is scripted to be used by six avatars, the accumulated error over the twelve trips (up and down) can add up to twelve centimeters of displacement, which can be visually noticeable. In order to avoid this drift, we implemented a position recalibration. The events coming from the narrative layer include the position of the elevator, expressed as seconds from the ground floor. When the IVR engine receives an event from the narrative engine, it adjusts the position of the elevator to match this position.

6. Discussion

The method described here can be used to create experiences that induce a rich illusion of time travel, which provides a new type of human experience, opening up a vast number of applications. We have provided a detailed explanation of how our approach was used to achieve a non-trivial example, but we see this as only a first step, and much more work is required. In modelling time and causality we have taken some 'metaphysical' decisions. Working out the implications of these decisions, as well as exploring alternatives, could be of high interest. Clearly, additional examples and case studies are needed. A formal treatment is required, which would allow us to prove that the unfolding of events is in some sense correct: consistent and plausible. Scaling is a notorious problem in artificial intelligence, and is also a major issue for digital time travel.

We note that this type of time travel illusion does not necessarily need to be limited to virtual reality. It may be possible to extend it to augmented or mixed reality (in fact, the participant's clones are based on realistic motion capture rather than synthetic animation). In the future, we can imagine a personal life-logger that could allow people not only to relive previous events from a different perspective, but also as a personal time travel device. Rather than just regretting about past mistakes, people may opt to practice reliving the events and intervening, while exploring the possible alternative outcomes. In general, we suggest that experiences such as the illusion of time travel may become part of daily life, and perhaps becoming a useful tool at our disposal, to revisit our decisions and reflect on the consequences of our behavior.

7. Acknowledgements

This research was partially supported by the European Union FP7 project BEAMING (`http://beaming-eu.org`) and European Research Council Advanced Grant TRAVERSE (227985).

References

[BDM08] Botzung A., Denkova E., Manning L.: Experiencing past and future personal events: Functional neuroimaging evidence on the neural bases of mental time travel. *Brain and cognition 66*, 2 (2008), 202–212. 1

[BGS13] Banakou D., Groten R., Slater M.: Illusory ownership of a virtual child body causes overestimation of object sizes and implicit attitude changes. *Proceedings of the National Academy of Sciences 110*, 31 (2013), 12846–12851. 1

[Ehr07] Ehrsson H. H.: The experimental induction of out-of-body experiences. *Science 317*, 5841 (2007), 1048–1048. 1

[FPOB*14] Friedman D., Pizarro R., Or-Berkers K., Neyret S., Pan X., Slater M.: A method for generating an illusion of backwards time travel using immersive virtual realityâĂŤan exploratory study. *Frontiers in psychology 5* (2014). 1, 2

[FT11] Friedman D., Tuchman P.: Virtual clones: data-driven social navigation. In *Intelligent Virtual Agents* (2011), Springer, pp. 28–34. 6

[HGS*11] Hershfield H. E., Goldstein D. G., Sharpe W. F., Fox J., Yeykelis L., Carstensen L. L., Bailenson J. N.: Increasing saving behavior through age-progressed renderings of the future self. *Journal of Marketing Research 48*, SPL (2011), S23–S37. 1

[ML06] Mott B. W., Lester J. C.: U-director: a decision-theoretic narrative planning architecture for storytelling environments. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems* (2006), ACM, pp. 977–984. 2

[PR08] Parsons T. D., Rizzo A. A.: Affective outcomes of virtual reality exposure therapy for anxiety and specific phobias: A meta-analysis. *Journal of behavior therapy and experimental psychiatry 39*, 3 (2008), 250–261. 2

[PS11] Pan X., Slater M.: Confronting a moral dilemma in virtual reality: a pilot study. In *Proceedings of the 25th BCS Conference on Human-Computer Interaction* (2011), British Computer Society, pp. 46–51. 2

[RS97] Roberts D. J., Sharkey P. M.: Maximising concurrency and scalability in a consistent, causal, distributed virtual reality system, whilst minimising the effect of network delays. In *Enabling Technologies: Infrastructure for Collaborative Enterprises, 1997. Proceedings., Sixth IEEE Workshops on* (1997), IEEE, pp. 161–166. 2

[RY06] Riedl M. O., Young R. M.: Story planning as exploratory creativity: Techniques for expanding the narrative search space. *New Generation Computing 24*, 3 (2006), 303–323. 2

[SC07] Suddendorf T., Corballis M. C.: The evolution of foresight: What is mental time travel, and is it unique to humans? *Behavioral and Brain Sciences 30*, 03 (2007), 299–313. 1

[Sla09] Slater M.: Place illusion and plausibility can lead to realistic behaviour in immersive virtual environments. *Philosophical Transactions of the Royal Society B: Biological Sciences 364*, 1535 (2009), 3549–3557. 1

[SNN*13] Spanlang B., Navarro X., Normand J.-M., Kishore S., Pizarro R., Slater M.: Real time whole body motion mapping for avatars and robots. In *Proceedings of the 19th ACM Symposium on Virtual Reality Software and Technology* (New York, NY, USA, 2013), VRST '13, ACM, pp. 175–178. 6, 7

[SO09] Steed A., Oliveira M. F.: *Networked Graphics: Building Networked Games and Virtual Environments*. Elsevier, 2009. 2, 7

[SRR98] Sharkey P. M., Ryan M. D., Roberts D. J.: A local perception filter for distributed virtual environments. In *Virtual Reality Annual International Symposium, 1998. Proceedings., IEEE 1998* (1998), IEEE, pp. 242–249. 2

[SSSV*10] Slater M., Spanlang B., Sanchez-Vives M. V., Blanke O., et al.: First person experience of body transfer in virtual reality. *PloS one 5*, 5 (2010), e10564. 1

[TIHS*01] Taylor II R. M., Hudson T. C., Seeger A., Weber H., Juliano J., Helser A. T.: Vrpn: a device-independent, network-transparent vr peripheral system. In *Proceedings of the ACM symposium on Virtual reality software and technology* (2001), ACM, pp. 55–61. 7

[VS05] Vives M. S., Slater M.: From presence to consciousness through virtual reality. *Nature Reviews Neuroscience 6*, 4 (2005), 332–339. 1

[YR03] Young R. M., Riedl M.: Towards an architecture for intelligent control of narrative in interactive virtual worlds. In *Proceedings of the 8th international conference on Intelligent user interfaces* (2003), ACM, pp. 310–312. 2