Temporal Links: Recording and Replaying Virtual Environments

Chris Greenhalgh, Jim Purbrick, Steve Benford, Mike Craven, Adam Drozd, Ian Taylor
School of Computer Science and Information Technology
The University of Nottingham, Jubilee Campus, Nottingham, NG8 1BB, UK
+44 115 9514221 {cmg, jcp, sdb, mpc, asd, imt}@cs.nott.ac.uk

ABSTRACT

Virtual reality (VR) currently lacks the kinds of sophisticated production technologies that are commonly available for established media such as video and audio. This paper introduces the idea of temporal links, which provide a flexible mechanism for replaying past or recent recordings of virtual environments within other real-time virtual environments. Their flexibility arises from a combination of temporal, spatial and presentational properties. Temporal properties determine the relationship between time in a live environment and time in a recording, including the apparent speed and direction of replay. Spatial properties determine the spatial relationship between the environment and the recording. Presentational properties determine the appearance of the recording within the environment. These properties may be fixed, dynamically varied by an application, or directly controlled in real-time by users. Consequently, temporal links have a wide variety of potential uses, including supporting post-production tools for virtual environments, post-exercise debriefing in training simulators, and asynchronous communication such as VR email, as well as providing new forms of content for virtual worlds that refer to past activity. We define temporal links and their properties and describe their implementation in the MASSIVE-3 Collaborative Virtual Environment (CVE) system, focusing on the underlying record and replay mechanisms. We also demonstrate applications for adding new content to an existing virtual world, and a VR post-production editor.

Keywords

Collaborative virtual environments, temporal properties, recording and editing techniques, post-production technologies.

1. INTRODUCTION

Virtual reality (VR) is rapidly emerging as an entertainment medium in areas such as games, theme park rides, and on-line communities. There are also opportunities for convergence with other media such as film, video and television. At one level, virtual reality provides a production tool to support these other media. For example, motion-capture techniques use virtual reality technology to record actors' movements in real-time as part of the process of creating computer animations for films, videos and games [1]. At another level, content originally developed for films and TV shows may find its way into virtual reality and vice versa. It is now commonplace to spin-off real-time games and rides from films, and more recently films from games. Finally, virtual reality may be directly combined with other media. For example, inhabited television involves staging television shows within on-line virtual worlds, so that the public can directly participate by taking control of virtual characters [2].

However, whereas established media such as video and audio are supported by a wide variety of production and post-production tools for editing, manipulating and assembling content, virtual reality generally is not. There are only a few examples of capturing and editing action within virtual environments so that it may be included in other virtual environments or be transferred into other media (see below).

This paper defines and demonstrates a flexible mechanism for recording activity in virtual environments, manipulating such recordings, and then accessing them as new content within other live virtual environments. This includes the ability to access recordings within recordings. These facilities are made available as a generic system mechanism whose properties may be configured to support a range of potential applications. We anticipate that such a mechanism will be an essential component of future VR production and post-production tools. Our discussion focuses on collaborative virtual environments (CVEs) – distributed virtual environments that support communication among potentially many users over a computer network. The heart of our mechanism is a technique for capturing all activities within a CVE, including the environment itself, multiple users' movements, speech, and interactions with virtual objects, in such a way that this action can be faithfully recreated at a later time. This record and replay technique is made available through the more general mechanism of a 'temporal link'. This allows application developers and end-users to program or directly manipulate the temporal, spatial and presentational relationships between the live environment and the recording. Action within the live environment can take place around and within the display of the recorded activity, and the composition of the two can itself generate further recordings.

Several existing CVEs support some form of record and replay facility. In the CAVERNsoft system [3], recording of an avatar's movements and audio is possible as part of general support for persistent virtual environments. This facility has been used to create the Vmail system [4], a form of VR email, and to create guided tours within tele-immersion applications. As part of the COVEN project, the DIVE system was extended with event logging facilities that could completely record an entire virtual environment and the activity within it [5]. Although initially implemented to support the statistical analysis of patterns of user activity in relation to network traffic, this recording facility was subsequently extended with a replay facility to allow a previous session to be recreated (although not within another live virtual environment). The dVS system supports a similar facility for recording and then replaying a virtual environment. Multi-player 3D games also record and replay techniques to show highlights of previous game-play, examples of which include FIFA soccer from Electronic Arts

ACM Multimedia 2000 Los Angeles CA USA

and the automobile game Driver from GT Interactive Software, the latter allowing players to edit together their own movies from recordings of their own actions. In a related area, recent work on 'virtualized reality' has developed techniques for capturing live action within a physical environment by synthesizing recordings from multiple video cameras to produce a 3D graphical simulation [6]. Furthermore, a script language VRML History has been proposed to facilitate time navigation in WWW browser-based 3D-worlds [7].

Examples of related work which refers to temporal properties in multimedia applications, but not specifically to VR, include the Media Editor Toolkit (MET++) application framework which has methods for manipulation of temporal relations in multimedia presentations [8], the Command Stream Model (and associated TclStream implementation) which introduces a media stream of code fragments that is synchronised to conventional media streams or device controllers, allowing presentation with VCR-type playback [9], and the spatio-temporal model of Vazirgiannis et al. which provides a framework for composition and indexing of multimedia objects [10]. The Where We Were (W3) system is being developed to allow recently recorded video, and 2D sketches, to be incorporated into real-time activities such as brainstorming meetings [11], and the Virtual Meeting-room Service (VMS) presents visual representions of recorded histories from distributed collaborations [12]. The CHIMP framework considers collaborative authoring and editing of multimedia documents, which includes temporal specification of objects [13].

2. DEFINING TEMPORAL LINKS

This section presents a general (i.e., system independent) description of the mechanism of temporal links. An implementation and demonstrations in the MASSIVE-3 CVE system [14] are presented in sections 3 and 4 respectively.

Our departure point is an established idea, that of being able to link together multiple spatial regions to create a virtual environment. In what is perhaps the most flexible scheme proposed to date, an environment is composed of multiple 'locales' that may be linked together to form a larger structure [15]. Each locale defines a local co-ordinate system and can be linked to a number of other locales or even to itself. Each such link specifies a spatial transformation that determines how the other locale is spatially related to the current locale, including the offset of its origin and a scale factor. In this way, locales may be placed adjacent to one another, may be overlaid, may be nested or may even be linked back to themselves to form mirrors and infinite corridors.

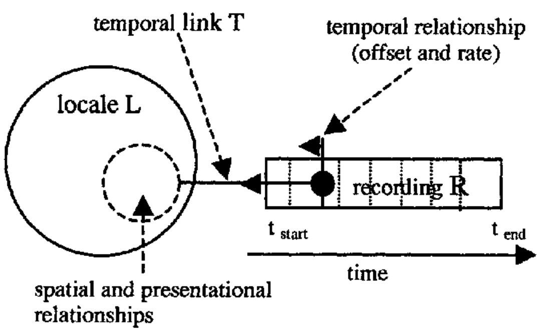

Figure 1. A temporal link

Our mechanism of temporal links extends this approach by allowing a locale to be linked to a recording of past activity from another locale (or indeed, from itself). The recording can be replayed and viewed from within the live locale via the temporal link. In addition to specifying the spatial relationship between a locale and a recording through its spatial properties, a temporal link can also specify temporal and presentational properties. The temporal properties determine the relationship between time in the live locale and time in the recording, including a current position in recorded time and a speed and direction (forwards or backwards) of replay. The presentational properties specify how the appearance of the recording is transformed when it is viewed from within the locale. For example, the recorded activity may be seen in 'black and white' or may be dimmed in order to distinguish it from the live activity.

Figure 1 summarizes the key properties of a temporal link. It shows a temporal link, T that allows a recording of a locale, R to be replayed within a live locale, L. One end of the link indicates its temporal relationship to R in terms of a current position in recorded time and a speed and direction of replay. The other end indicates its current spatial and presentational relationship to L. It should be noted that with a temporal link, a recording is always accessed through a live locale, although the live locale may have no additional content or purpose, and so may appear to be invisible for all intents and purposes.

The live activity within a locale that is linked to a recording may itself be recorded. The resulting new recording will contain both the live action and the temporal link which fully describes how the original recording is related to (and situated within) the new recording. Replaying the new recording will use this temporal link to access and replay the old recording within it in an appropriately synchronized manner. Figure 2 presents a more general scenario in which a locale is linked to multiple recordings, each associated with different spatial, temporal and presentational properties. One of these recordings is itself being generated by current activities within the locale.

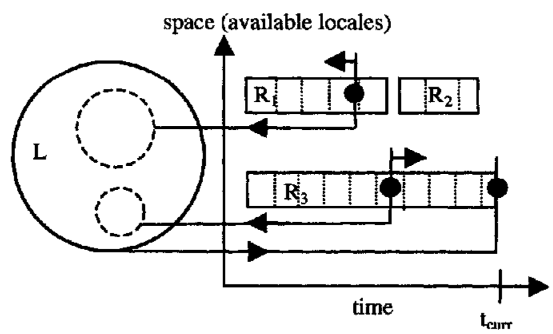

Figure 2. Linking to multiple recordings

The remainder of this section explores the definition of temporal links in greater depth, including the nature of temporal, spatial and presentational properties.

2.1 Temporal properties

As noted above, one aspect of a temporal link specifies the timing of the replay of a recording in terms of a current position in time within the recording, and the direction and speed of replay. This requires a mechanism to translate between 'live time' and 'recorded time'. Live time in the locale progresses according to the local system clock. Recorded time is determined by the system clock of the computer that ran the recording process that recorded and time-stamped events. The temporal link provides a function to map between these two timescales. Specifically, this function allows a value of recorded time to be calculated for each value of live time. The parameters of this function provide the means by which application developers or end users can control the temporal properties of the replay.

Assuming a linear function for the time being (the basis of our implementation as described below), a value of live time is multiplied by a rate and then an offset is added in order to arrive at the corresponding value of recorded time. The offset establishes an initial mapping from a chosen point in live time to a corresponding point in recorded time. The rate determines both the apparent speed and direction of the replay of the recording as live time advances. A rate of 0 will result in recorded time being frozen. A rate of 1 will result in recorded time advancing at the same rate as live time. A negative rate will cause recorded time to run backwards as live time advances.

The rate and offset may be fixed when the link is created, establishing a constant relationship between live and recorded time. Alternatively, they may be manipulated in real-time, for example to enable users in the locale to browse through the recording.

This linear function is just on example of the wide range of functions that might be chosen to map live time onto recorded time. Non-linear functions might be used to create interesting effects. A simple non-continuous function could be used to create a looped replay. A sinusoidal function would result in an continually repeated replay that oscillated forwards and backwards through recorded time. An inverse exponential function would result in a replay that edged ever closer to a moment in recorded time (perhaps a critical event in a story), the longer you watched it. Alternatively, a random function could be used to replay random moments from the past.

2.2 Spatial properties

The spatial properties of a temporal link control the relationship between the spatial frame of reference of the live locale and that of the recording. This relationship is specified by a homogeneous 3D matrix that allows the spatial frame of the recording to be translated, rotated, scaled and sheared relative to that of the live locale. The two spatial frames can be directly matched and overlaid so that the live action takes place around the recorded action and conversely, the recording appears to be a normal part of the live action. Alternatively, the recording can be situated as a separate locale (e.g., as an adjacent locale, a window or mirror onto the past, or as a 'world in miniature').

It may also be useful to be able to locate the recording within an enclosing object (for example making it appear to be playing back inside a virtual television set). This requires additional support for clipping the recorded material as it is replayed so that it does not spill outside of the containing object.

As with temporal properties, the spatial properties may be fixed when the link is created, or may vary dynamically. For example, the recording can appear to move around, rotate, expand and shrink, within the locale.

2.3 Presentational properties

Presentational properties can be used to systematically modify the appearance of the recording so that it can or cannot be distinguished from live action within the live locale according to the requirements of a particular application. This includes modifying graphical attributes such as brightness and color (e.g., dimming the replay or playing it back in 'black and white'), but may also include modifying audio information, for example playing it back quietly or with appropriate effects.

3. IMPLEMENTATION IN MASSIVE-3

This section describes how temporal links are implemented in the MASSIVE-3 CVE system [14]. We begin by considering mechanisms for recording virtual activity, followed by replaying, and finally control (i.e. the temporal links themselves). However, first we need to describe elements of MASSIVE-3's basic operation.

A MASSIVE-3 virtual world is comprised of one or more "locales", each of which is a distinct region of virtual space (as already described). Each locale is implemented as a distributed database (called an "Environment"). This database completely describes the locale and all of its contents, including scenery, buildings, users' avatars and any virtual objects in the locale; this is primarily a shared scene graph. Each process which joins a particular locale contacts a master process for that locale and obtains (over the network) an initial "snapshot" or checkpoint of the locale's current state. Any process that has joined a locale can change it, e.g. by adding, changing or deleting elements of the scene graph. As shown in the left-hand side of Figure 3, each change requested by application code is not performed directly; instead an event object is created which describes the requested change. This event object is then forwarded to the other processes that have joined the locale (not shown), as well as being used to update the local copy of the database.

3.1 Recording

The left-hand side of Figure 3 shows how MASSIVE-3 allows a virtual environment and the activity that occurs within it to be recorded for subsequent recreation. First, we create a checkpoint at the start of a recording, as an initial reference point. Then the rest of the recording consists of the sequence of events that are applied to the locale, each one being time-stamped and written to the file as it is performed. This allows us to record the complete state of a locale and everything that happens to it, including the avatars' movements and interactions with objects. In addition, other media must also be recorded. MASSIVE-3 currently records all audio streams within the locale, writing the audio packets to a second time-stamped log file.

3.2 Replay

The right hand side of Figure 3 shows how a recording can then be used to recreate past locales and activity. We consider this in a number of incremental stages.

3.2.1 Basic Replay

The replay process creates a new replay-only locale over which it has complete control. The initial state of the locale is taken from the first checkpoint in the recording, which is often empty (before any data was placed in the locale). The recorded events can then be read in turn from the recording, and, at the correct moment, injected into the new locale, recreating the activities which occurred in the original locale. The replay application also replays recorded audio streams, synchronized to the replay locale.

Note that the original locale may still exist, and the same recording may be used to create several concurrent replays (each

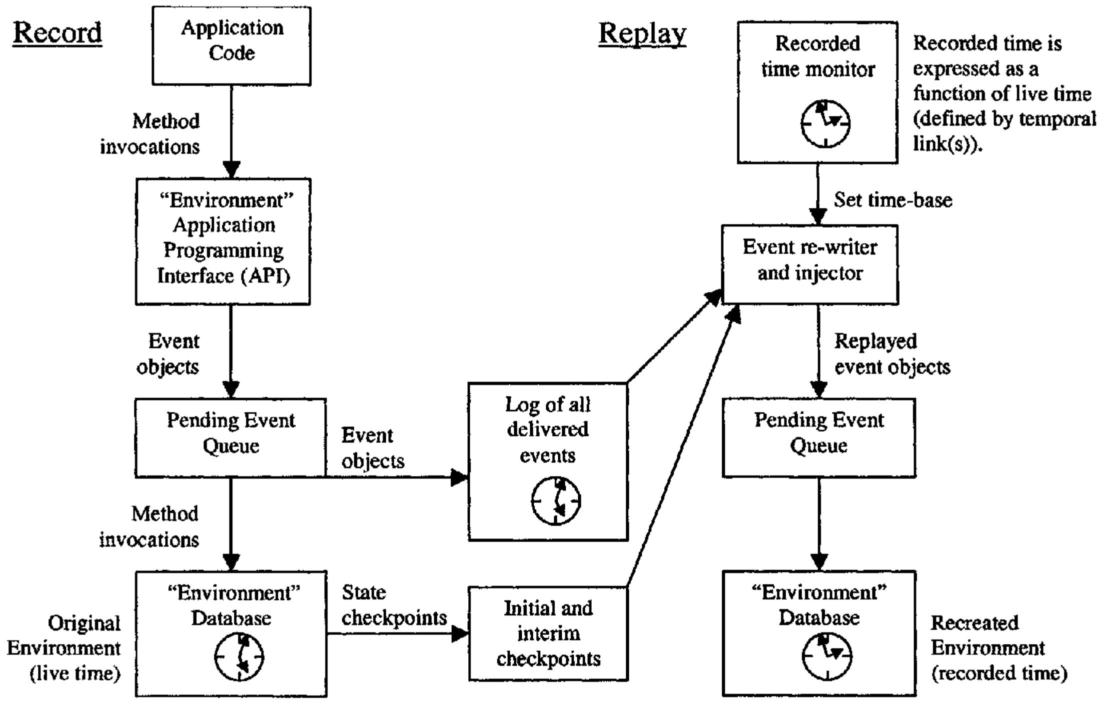

Figure 3. Record and replay in MASSIVE-3

at a different moment in recorded time). Consequently the internal identifiers for the recreated locale and its constituent data items must be changed to ensure that each replay locale is clearly distinguishable. The database annotations that describe audio streams must also be updated to correspond to the appropriate replay streams.

3.2.2 *Replay Time*

Not all changes in a locale are represented by explicit events. In particular, deterministic processes, such as movement with a known velocity or frame-by-frame interpolation between known positions, are not represented as recorded events. These are implied by the passage of time, and are independently evaluated frame-by-frame for each distributed copy of the locale. Consequently, a replayed locale must explicitly control its own definition of time so that these deterministic processes will reflect recorded time rather than current time.

Each locale has its own, potentially independent, definition of time. For current locales this is normally taken directly from the host computer's real-time clock. However, for replayed locales, time is the moment in history currently being replayed (i.e., recorded time). The replay application uses a specialized event ("SetTimebase") to set the replay locale's definition of time to correspond to that of the recording. Time in a locale is currently defined as a linear function of the current host time, as described (for temporal links) in section 2.1. Changing the recorded time rate (and the rate at which events are re-introduced) allows the replay application to stop the recording, or play it faster or slower than real-time.

Note that deterministic behaviors in the replay locale can be re-evaluated for any moment in recorded time, irrespective of whether this corresponded to a frame time or other key moment in the recording. This gives almost infinite temporal resolution for these elements of the recording (subject the precision of the behaviors and their evaluation). Consequently, states can be recreated that were never actually observed in the live action. It is also possible to replay the recorded action in a perfected form. For example, network delays may cause glitches in the performance of a virtual environment that could be removed when the action is subsequently replayed.

3.2.3 *Reversing Time*

MASSIVE-3 also allows recordings to be played backwards, reversing recorded time. Deterministic behaviors are trivially reversed by defining the replay locale's time so that it decreases as live time progresses; the frame-by-frame evaluation of these behaviors will naturally regress.

However, forward-time events cannot be used when moving backwards in a recording. For example, the reverse time version of adding a node in the scene graph is to *delete* that node, and the reverse time version of updating a node's value is to set it back to its old value. MASSIVE-3 allows recordings to be played backwards by constructing "anti-events" during normal (forward) playing of a recording: an anti-event completely reverses the effects of the corresponding forward-time event, as in the above examples. The replay application uses these anti-events when time is moving backwards.

3.3 Control

The last two sections have described the technical mechanisms by which activity in a locale can be recorded and subsequently replayed. This section explains how this is controlled, i.e. how temporal links operate in MASSIVE-3. However, first we must outline the implementation of normal (non-temporal) inter-locale links.

An inter-locale link (called a "boundary") is one of the standard data-types supported by the MASSIVE-3 locale database. As well as the spatial and presentational properties of the link it also gives the name of the environment at the far end of the link. A process that wants to join the linked (far-end) locale looks up the locale's given name using a specialized naming service that is part of MASSIVE-3's run-time system. For a normal locale, the naming system will return the system identifier of the far-end locale and the network address of the locale's current master process, which can then be contacted in order to join the locale (see also the start of section 3). The naming service can also start a locale master process on demand (when a named locale is requested) if one is not already running.

3.3.1 Temporal Links

The key difference between a normal inter-locale link and a temporal link is the form of name given for the far-end locale. For a normal link this is just the name of the far-end locale, but for a temporal link it has the form:

original-locale-name@temporal-property-id

The "temporal-property-id" is the system identifier of a data item that defines the temporal properties of the link, i.e. the rate and offset to be used to relate live (near-end) and recorded (far-end) time. The '@' symbol may be read as "at the time specified by". This temporal property is normally but not necessarily in the near-end locale.

When the naming service receives a new request for a name of this form it starts a new replay application with the given name. The new replay application locates the event and audio log files created from the original locale (e.g. using a standard file-naming convention). It also finds and joins the locale that contains the link's temporal property data (using the *temporal-property-id*). The replay application then continuously monitors the link's temporal properties and the time in the near-end locale. It combines these to dynamically determine the overall time rate and offset for the replay, which is used as in described in section 3.2.

In summary, to create a temporal link a process creates a data item to contain the link's temporal properties, and then creates the link itself using the correct name (including the identifier of the link's temporal property data item). The replay application will be started automatically as soon as any process requires it. Whenever the temporal property is changed the replay application observes this and adjusts the replay accordingly (e.g. stopping, running faster, slower or backwards).

3.3.2 Replaying Links

A link (temporal or otherwise) is recorded in the same way as anything else in a locale. If the locale is recreated then the link will also be recreated. However, the link must be modified when it is replayed in order to preserve the correct temporal relationships between locales and recordings. Specifically, the destination name of a replayed link must have the replay's temporal properties appended to it. This ensures that the far-end of any replayed link is also historically appropriate.

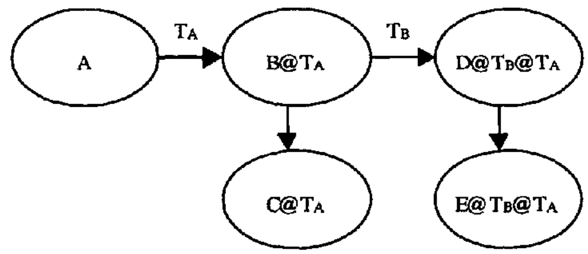

Figure 4. Example multi-locale environment with replay

Consider the example environment shown in Figure 4. This has one current locale, A, which includes a temporal link to a recording of locale B, which it accesses as "B@T_A", i.e. B at the time specified by temporal property T_A. When recorded, B had a normal inter-locale link to locale C. When B is replayed this must be rewritten to point to "C@T_A", so that (viewed from A) a corresponding recording of C is observed via the recording of B.

Similarly, B had its own temporal link to D that was originally viewed from B as "D@T_B". Now, viewed from A, this becomes "D@T_B@T_A", i.e. D at the time specified by temporal property T_A (which determines the time in B@T_A) combined with temporal property T_B (which is part of the recording).

4. DEMONSTRATING TEMPORAL LINKS

This section presents two demonstrations of temporal links in MASSIVE-3. Both are inspired by on-going experiments in inhabited television, the staging of television shows within CVEs [2]. Members of the public may either watch a broadcast of the show from the virtual environment as if it were television, or may metaphorically 'step through the screen' in order to enter the virtual environment and take control of a virtual character.

The basis of the demonstrators is an experiment called The Ages of Avatar, the latest in a series of public experiments with inhabited television [16]. Its goal was to explore whether entertaining broadcast material can be created by drawing on the characters, settings, and stories of a virtual community. The virtual community in this case was established around an existing television channel, Sky Television's [.tv] (pronounced "Dot TV"). The Ages of Avatar consisted of a series of virtual worlds, loosely based around the theme of The Ages of Man, which was initially implemented using Microsoft's VWorlds system.

To produce our demonstrators we have recreated the worlds in MASSIVE-3, with the aim of aim of showing how temporal links can enhance applications such as inhabited TV. The first demonstrator involves creating new content for an existing world that exploits the ability to refer to the past using temporal links. The second uses temporal links to create an editor tool for the post-production of broadcast material from recordings made in the virtual world.

4.1 Creating new world content

We have extended the content of a world called Behaviour Shift within The Ages of Avatar with new features that exploit the mechanism of temporal links. The theme of Behavior Shift is adolescence; it is a world of distorted perspectives, confused identities and changing behaviors. In keeping with this theme, our extensions allow participants to see their own actions in various, distorted and shifting, ways.

Our first extension has been to create 'shadows' that follow participants around the virtual environment, replaying their recent actions with a delay. These are not shadows in the conventional sense. Rather, they are objects created by replaying a recording in a suitably distorted way. A temporal link is established with an offset of ten seconds into the past and a rate of 1. The spatial properties of the link scale the Y axis to 0 and shear it to the X axis in order to create an approximation of a shadow on the floor. The presentational properties of the link are set to darken the replayed action, making it appear more shadow-like. The resulting shadows show where each participant was located within the environment ten seconds ago, including any interactions with objects at that time. An example of a shadow can be seen in the camera view window of Figure 7 (see later). In this case, the shadow is of a user's avatar interacting with a virtual ball.

Our second extension is a distorting mirror. From a user's point of view, the mirror appears to reflect the region of space directly in front of it, but with various temporal distortions that change their nature over time. Specifically, the replayed action is delayed in time so that the mirror shows action from the recent past. For two thirds of the time the action is shown at half speed and for the remaining third at double speed. Audio is also replayed at the appropriate speed. The net result is that the user sees their own gestures replayed at a mixture of speeds. Figure 5 shows a participant standing in front of the distorting mirror.

Our third extension is a 'Holovid', a device that enables participants to browse through past activity. The Holovid consists of a miniature view of the whole environment at (virtual) table-top size, combined with a series of controls for browsing the recording. Pressing 'play' runs the recording forward at normal speed (ratio of 1) from its current position. Repeatedly pressing play skips forward to the next recorded event. Pressing 'slow forward' plays forwards through the recording at half speed. Pressing 'rewind' plays backwards through the recording at normal speed. Pressing 'stop' halts the recording at its current position. Figure 6 shows a view of the Holovid, with the whole of Behavior Shift world being displayed in miniature. The figure on the left is the avatar of another user who is currently controlling the Holovid.

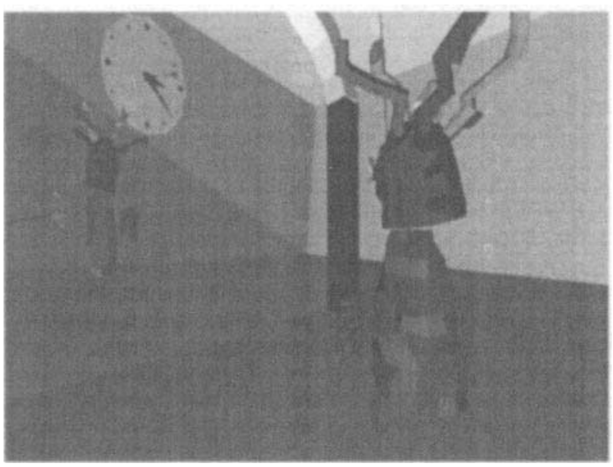

Figure 5. An avatar standing in front of the mirror

Between them, our three examples demonstrate a variety of configurations of temporal links. Temporal and spatial properties can be fixed (as in the shadow), can vary in a pre-programmed manner (the mirror) or can be under direct end-user control (the Holovid). Spatial and presentational properties are also used to situate the recorded material within the live locale in different ways.

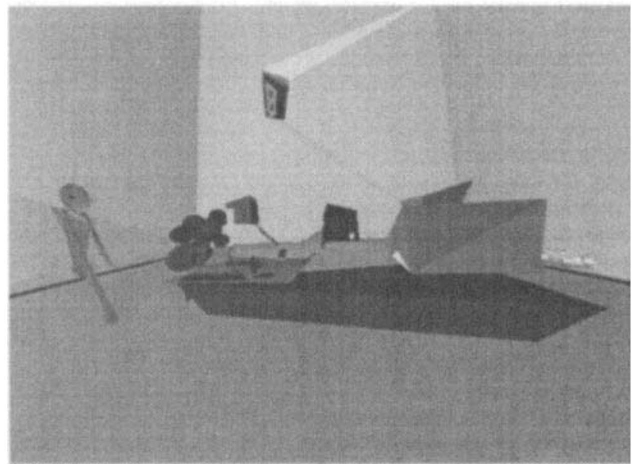

Figure 6. The 'Holovid', located inside a room, with user

4.2 A VR post-production editor

Our second demonstration, the VR editor, is a tool for the post-production of broadcast material from recordings made within virtual environments. The editor initially places the user in a live locale. This can be an empty locale that merely acts as a placeholder for recordings, or can contain some scenery, avatars and other objects that are to be overlaid on the recording when the final broadcast is produced.

The user can then create a temporal link, in effect loading an identified recording into this live locale. Once loaded they can browse through this recording at will, marking key moments in time in a list. They can then jump directly to these marked moments and run the recording from that point, forwards or backwards at whatever speed they specify. As the recording runs within the locale they can control the viewpoint of a virtual camera so as to view the replay from any desired perspective or sequence of viewpoints – effectively editing the camera work. Finally, the editor can be instructed to output the results of these manipulations in a number of ways. It can output a series of JPEG frames, with the user specifying how many frames are to be generated and the number of fast-forward, rewind and fast-rewind. A slider can be used to set the speed of these controls. Fast-forward and fast-rewind operate at twice this speed. The current recorded time is shown. At any time the user can use the store button to mark the current time, adding it to a visible list of marked times. Two further buttons can be used to jump to a particular time in this list or to remove an entry from it.

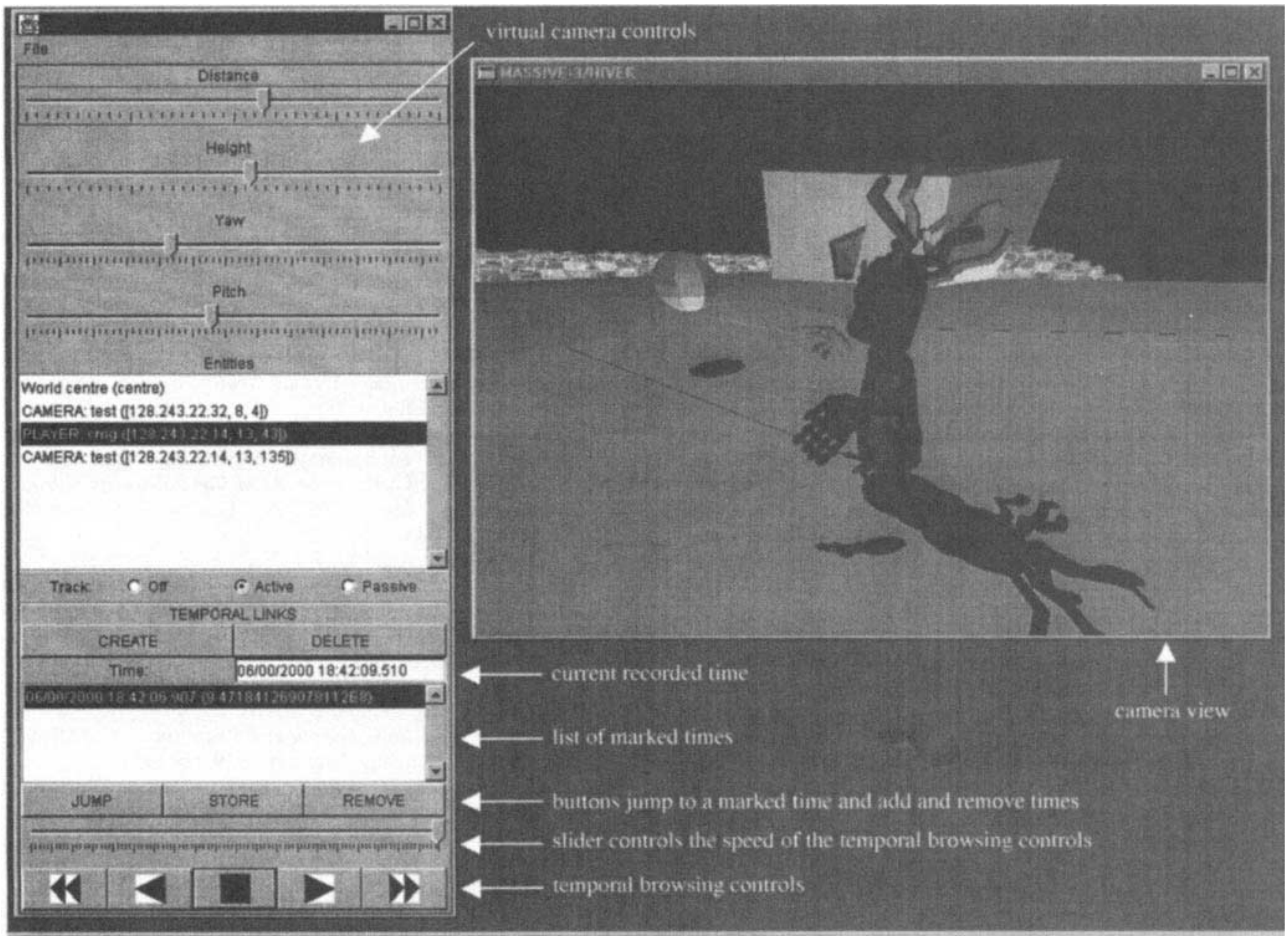

Figure 7. Graphical interface to the post-production

4.3 Other applications

There are many other potential uses of temporal links within CVEs. We conclude this section by briefly discussing two of them.

A key use might be in supporting the post-analysis and discussion of events within virtual environments. On-line debriefings can be carried out within virtual environments that are used for training and simulation, so that participants and experts can discuss and analyze their own performances in depth (e.g., viewing slow motion replays). They might also replay the performances of others to compare them with their own. A variation on this theme is supporting sociologists in analyzing social interaction within CVEs, using techniques such as ethnography and conversation analysis in order to evaluate their effectiveness. These techniques are already used for evaluating CVEs, but currently rely on studying video material captured from a virtual environment, often involving comparing multiple simultaneous recordings showing different participants' perspectives.

Temporal links might also be used to support asynchronous communication in CVEs. Recordings can be sent to other participants as a form of VR email that can then be replayed within a virtual environment (see [4]). The ability to build up layered recordings can support conversational threads in VR bulletin boards. They can also allow interviewers and interviewees to carry out asynchronous interviews. These kinds of techniques might be particularly useful for supporting low-bandwidth users of CVEs, for example where the CVEs use audio for communication, but where some participants are using dial-up modems that can only handle very limited real-time audio data.

5. SUMMARY

We have introduced the mechanism of temporal links as a way of replaying recordings of virtual environments within other virtual environments. The key feature of this mechanism is its generality – through a combination of temporal, spatial and presentational properties temporal links can establish a wide variety of relationships between the recorded material and the live action. Temporal properties control the timing of the recording relative to the live action, including speed and direction of replay. Spatial properties control the positioning of the recorded material within the live action, including scale, orientation and relative location. Presentational properties control the appearance of the recorded material relative to the live action so that it may be distinguished from it if appropriate. These various properties may be fixed, may be varied by the application or may be under the dynamic control of end-users.

We have described the implementation of temporal links within the MASSIVE-3 system, including a number of techniques that were required to make them effective, efficient and suitably flexible. The recording mechanism must capture all events within a locale, including avatars' movements and interactions with objects. The replay mechanism can infer the state of deterministic process at any given moment in recorded time, to give an almost infinite temporal resolution. The ability to reverse time and run the recording backwards is supported by the introduction of anti-events that effectively undo the effects of normal-events. Replaying a recording from a locale that contained links (spatial or temporal), requires that these links are also replayed appropriately to ensure historically accurate representations of the recorded locale's nested recordings and neighboring locales. Finally, we presented a number of demonstrations of temporal links based around an on-going public experiment in inhabited television called The Ages of Avatar. These included creating new content for virtual environments and also a VR editor for the post-production of broadcast material from live recordings in a virtual world.

Our future work will involve developing a broader range of applications of temporal links, especially extending our current post-production editor to become a fully functional VR editing suite that supports the creation of computer animations from action that has captured from real-time virtual environments. Indeed, we see this as being a key long-term benefit of temporal links – enabling non-real-time media such as computer animation to directly exploit live performances and improvised social interaction that take place within real-time collaborative virtual environments. In a sense, CVEs can act as a kind of social motion capture system, recording the dynamics of human social activity for subsequent use in film and video. However, we also propose that mechanisms such as temporal links are essential to the future development of virtual reality as a medium in its own right, one that like other media, needs to be supported by a rich set of production and post-production tools and techniques.

6. ACKNOWLEDGEMENTS

We acknowledge the support of the British EPSRC for the Multimedia Networking for Inhabited Television project under the Multimedia Networking Applications programme, and the European Community for the eRENA project under the ESPRIT IV Intelligent Information Interfaces (I3) programme. We would also like to thank our collaborators at Illuminations Television, British Telecom and BSkyB for their work on The Ages of Avatar that provided the content for our demonstration.

7. REFERENCES

[1] Gliecher, M., Retargetting motion to new characters, *Proc. ACM Computer Graphics* (SIGGRAPH'98), Orlando, USA, July 1998, pp. 33-43.

[2] Greenhalgh, C., Benford, S., Taylor, I., Bowers, J., Walker, G. and Wyver, J., Creating a live broadcast from a virtual environment, *Proc. ACM Computer Graphics* (SIGGRAPH'99), Los Angeles, USA, August 1999, pp. 375-384.

[3] Leigh, J., Johnson, A., DeFanti, T., Brown, M., et al., A review of tele-immersive applications in the CAVE research network, *Proc. IEEE Virtual Reality* (VR'99), Houston, USA, March 1999, pp. 180-187.

[4] Imai, T., Johnson, A., and DeFanti, T., The virtual mail system, *Proc. IEEE Virtual Reality* (VR'99), Houston, USA, March 1999, p. 78.

[5] Frécon, E., Greenhalgh, C. and Stenius, M., The DIVE-BONE – an application-level network architecture for Internet-based CVEs, *Proc. ACM Virtual Reality Software and Technonology* (VRST'99), London, UK, December 1999, pp. 58-85.

[6] Kanade, T., Rander, P., Vedula, S. and Saito, H., Virtualized reality: digitizing a 3D time varying event as is and in real time, in Ohta, Y. and Tamura, H. (eds.), *Mixed Reality, Merging Real and Virtual Worlds*, Springer-Verlag, 1999, pp. 41-57.

[7] Luttermann, H. and Grauer, M., VRML History: Storing And Browsing Temporal 3D-Worlds, *Proc. ACM Virtual Reality Modeling Language Symposium* (VRML'99), Paderborn, Germany, February 1999, pp. 153-181.

[8] Ackermann, P., Direct manipulation of temporal structures in a multimedia application framework, *Proc. ACM Multimedia* (MM'94) , November 1994, pp. 51-58.

[9] Herlocker, J.L. and Konstan, J.A., Commands as Media: Design and Implementation of a Command Stream, *Proc. ACM Multimedia* (MM'95), November 1995, San Francisco, USA, p. 155-165.

[10] Vazirgiannis M., Theodoridis Y. and Sellis T., Spatio-temporal composition and indexing for large multimedia applications, *Multimedia Systems*, 6(4), 1998, pp. 284-298.

[11] Minneman, S.L., Harrison, S.R., Where were we: making and using near-synchronous pre-narrative video, *Proc. ACM Multimedia* (MM'93), August 1993, Anaheim, USA, pp. 207-214.

[12] Ginsberg, A., Ahuja, S., Automating envisionment of virtual meeting room histories, *Proc. ACM Multimedia* (MM'95), San Francisco, USA, November 1995, pp. 65-75.

[13] Candan, K.S., Prabhakaran, B., Subrahmanian, V.S., CHIMP: A framework for supporting distributed multimedia document authoring and presentation, *Proc. ACM Multimedia* (MM'96), Boston, USA, November 1996, pp. 329-340.

[14] University of Nottingham, Communications Research Group, MASSIVE-3 / HIVEK, http://www.crg.cs.nott.ac.uk/research/systems/MASSIVE-3

[15] Barrus, J.W., Waters, R.C. and Anderson, D.B., Locales: Supporting Large Multiuser Virtual Environments, *IEEE Computer Graphics and Applications*, 16(6), November 1996, pp. 50-57.

[16] Craven, M., Benford, S., Greenhalgh, C., Wyver, J., Brazier, C.-J., Oldroyd, A. and Regan, T., Ages of Avatar: Community Building for Inhabited Television, to be published *Proc. ACM Collaborative Virtual Environments* (CVE'2000), San Francisco, USA, September 2000.