

Using record and reuse technologies to create a mixed reality TV show

Adam Drozd, Steve Benford, Duncan Rowland, Robin Allen, Martin Flintham

The Mixed Reality Laboratory

The University of Nottingham

Nottingham, NG8 1BB, UK

{asd, sdb, dar, rxa, mdf} @cs.nott.ac.uk

Introduction

In this position paper we explore the use of record and reuse facilities to create broadcast television material from mixed reality experiences that involve online and ‘on the streets’ participants. This is motivated by our previous work on inhabited television and also by ongoing discussions about making a TV pilot show based on the game *Can You See Me Now*. We describe this application area, introduce some early prototype production tools, and then draw out general requirements for underlying record and reuse facilities.

Inhabited television

Inhabited television sees the combination of collaborative virtual environments (CVEs) and broadcast TV to create a new medium for entertainment and social communication (Benford, 2000). The defining feature of this medium is that an on-line audience can socially participate in a show that is staged within a shared virtual world. The producer defines a framework, but it is the audience interaction and participation that brings it to life. A broadcast stream is then mixed from the action within the virtual world and transmitted to a conventional viewing audience, either as a live event or as a post-produced and rendered broadcast stream.

Inhabited TV has been a topic of research since 1996 through projects such as *The Mirror* (Illuminations Television/BT/BBC) and *Heaven and Hell Live!* (Illuminations Television/BT/Channel 4). At the time of writing, we are seeing the first commercial examples of inhabited TV beginning to appear on mainstream television, for example the current show *Fightbox* from Richochet/BBC.

Our own research into inhabited TV has unfolded through a series of collaborative projects that explored different aspects of the user experience and production technologies. In 1998 we staged a live inhabited TV show called *Out of This World* (University of Nottingham/Illuminations TV/BT) that demonstrated how dedicated virtual camera and participant management technologies could be used to create a relatively fast-paced and coherent gameshow within a CVE (Greenhalgh, 1999) In 2000, in a follow-on project called *Avatar Farm*, we attempted to create a more complex show which took the form of an online drama in which members of the public and professional actors collaborated as part of a non-linear drama of four

chapters that roamed across four virtual worlds (Craven,2001). A key feature of Avatar Farm was that we made use of a 3D record and replay mechanism for collaborative virtual environments that was implemented in the MASSIVE-3 system (Greenhalgh, 2000; Greenhalgh 2002) to enhance the experience in several ways:

- Our actors improvised and recorded several scenes in the virtual world on the day before the show. These pre-recorded scenes were then replayed during the main show as, enabling the live participants to move among a series of ghostly flashbacks as shown in figure 1 left.
- We saved the whole of Avatar Farm as a series of 3D recordings and subsequently created a series of interfaces to support observers in exploring the story, moving through the action and following different characters such as the table interface shown in figure 1 middle.
- We demonstrated the possibility of exporting selected parts of these 3D recordings into conventional animation tools so that offline rendered animations of the action could be made as shown in figure 1 right.



Figure 1: Uses of record and replay in Avatar Farm – flashbacks (left), reviewing the story (middle) and postproduced animation (right)

One of the potential problems with inhabited TV is the visual quality of the material produced from CVEs. Certainly our early discussions with broadcasters consistently raised the concern that the quality of the real-time computer graphics was insufficiently rich to provide a compelling viewing experience, for example that avatars were insufficiently expressive for viewers to be able to empathise with them. We see three approaches to dealing with this issue:

1. Simply wait for the quality of real-time graphics to improve sufficiently, and perhaps also for viewers to become increasingly accepting of real-time avatars as online games spread and increase in popularity.
2. Mix the computer graphics with conventional video footage of real actors and players. This is the approach taken by current inhabited TV shows which are typically set within a studio environment, providing the possibility of mixing footage of the human players and a studio audience with computer graphics.
3. Postproduce the computer graphics to create offline rendered animations that can match the quality of current cartoons and animated films.

The tools and techniques that we introduce in this paper are intended to support both approaches 2 and 3.

Can You See Me Now and Inhabited TV

We illustrate this paper with an example of a possible inhabited television show – making a television show from the mixed reality game Can You See Me Now.

CYSMN is a chase game. Up to fifteen *online players* at a time, logged in over the Internet, are chased through a virtual model of a city by three *runners*, professional performers, who are running through the actual city streets equipped with handheld computers, wireless network connections (using 802.11b) and GPS receivers. The online players can move through the model with a fixed maximum speed, can access a map view of the city, can see the positions of the other players and the runners, and can exchange text messages with them. The runners move through the streets, can see the positions of the online players and other runners on a handheld map, can see the players' text messages and can communicate with one another using walkie-talkies.

A key feature of the game is that the runners' walkie-talkie communication is streamed to the players over the Internet, providing them with ongoing description of the runners' actions, tactics and experience of the city streets, including reports of traffic conditions, descriptions of local topology and the sound of the physical exertion involved in catching a player. If a runner gets to within five virtual meters of an online player, the player is 'seen' and is out of the game (their score is the time elapsed since joining the game). Runners also carry digital cameras so that they can take a picture of the physical location where each player was caught and these pictures appeared on an archive web site after the event.

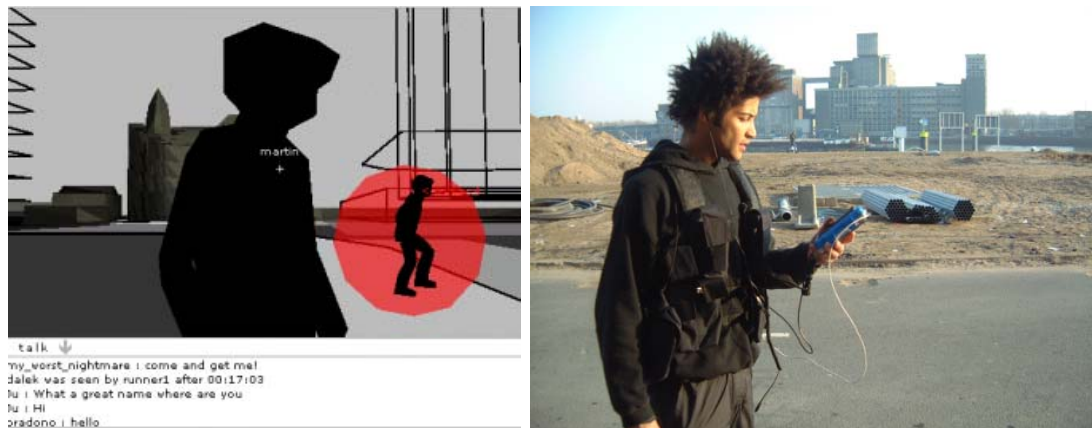


Figure 2: online player's view (left) and a runner (right)

We are interested in adopting CYSMN as the basis for an inhabited TV show – as are other bodies such as the Arts Council who have provided funding for creating a pilot in collaboration with a broadcaster – believing that it offers a rich way of engaging real-world participants with a virtual environment. Specifically, some physical participants are running through the city streets rather than being sat at computers, potentially providing more interesting visual material for television and opening up new storylines in terms of their engagement with the city and its inhabitants. Of course, the online players are still sitting at conventional computers. However, CYSMN also raises some specific challenges in terms of how the experience can be captured, for example, in terms of how video cameras can capture the actions of runners and players.

Supporting the production process

We now consider the production process for creating an inhabited TV show based upon CYSMN. Our assumption here is that in order for online players to be real viewers- members of the public – the game needs to be initially played live and a TV broadcast then created from the material that is recorded, rather than pre-scripting all of the action and then shooting the show scene by scene. In other words, the process is essentially one of creating a TV show of a live event (this is not to say that the TV show itself has to be live – but the event does).

Key activities in the production process are as follows.

Designing and staging a (revised) live experience

We need to stage the live experience in the first place. This involves the usual(?) activities of creating one of our mixed reality experiences: overall experience design, choice of physical location, 3D modelling, server and interface design, configuring the network and positioning technologies, testing and roll out. Some additional activities may be required to support the television process, for example, choosing camera positions and possibly marking the physical streets to support automated camera tracking techniques (where visual tracking is used) or to help runners understand where to run to generate the best shots. The live experience is then staged, possibly several times over several days, with actors playing the runners and members of the public as the online players.

Recording the live experience

There are two key aspects to recording the live experience:

- First we need to record the online experience. This means saving the system state as a series of log files that can subsequently be replayed so that the online aspects of the experience can be re-run as if live. The state recording is probably best made from the point of view of a single system component that offers an agree view of events – i.e., from the game server. State recordings will include timestamped logs of online players and runners movements and all text messages from online players as they were seen by this game server. The real time audio stream generated from the runners' walkie-talkies will also need to be recorded in such a way that it can be synchronised with the system state.
- We will need to record physical participants using conventional audio/video cameras. A variety of cameras may be used including: fixed cameras (e.g., on stands), handheld cameras, platform mounted cameras (possibly even on remote controlled helicopters) and wearable cameras (e.g., to record first person perspectives of the runners). It will be important to synchronise video data with the state recordings of the online play. This includes both temporal and spatial synchronisation, the latter requiring us to know how the cameras were positioned in relation to the 3D model of the city. For fixed cameras this will mean measuring their positions in relation to the model. For mobile cameras, it will mean capturing data about their movements, perhaps using a combination of GPS, inertial sensors and visual tracking. We may also need to record other camera information such as focus and zoom settings to support subsequent animation work.

Reviewing, selecting and overdubbing material

It will be necessary to select a subset of the recorded material to be used to create the final broadcast. It may also be necessary to improve some of it, for example, rerecording some parts of both the physical and online action.

- We require a tool to quickly select key scenes of interest from the overall recordings. In CYSMN these are most likely to be chase scenes involving one or more runners and a particular player. It would be useful – but potentially difficult – to be able to automatically detect such scenes. It is certainly possible to automatically detect the end-point of those chases where a runner eventually catches an online player. However, detecting the beginning of the scene, identifying which other runners were involved and spotting unsuccessful chases is more difficult.
- The production team may benefit from an overview of the structure of the experience, being able to visualise what happened and mark up where and when key scenes occurred. This might be supported through an appropriate visualisation tool.
- The production team will wish to review the selected scenes, quickly jumping to their start points and repeatedly replaying them. They will wish to review both the recorded online and video material at the same time so as to compare the activities of online players and runners during the scenes. This requires a technique to cross index between the online and video material. They will need to explore different camera views by positioning virtual cameras in the system state recordings and selecting from among the available video recordings.
- It may be useful to correct and overdub the recorded material. An example of correction might be altering the path of an online player. An example of overdubbing would be recording scenes involving runners on the streets so that they can more carefully enact the chase, making it more existing for viewers. This might be aided by enabling them to run live against a replay of selected scenes from the online recordings (as if the online players were present) and then to record their new movements (both online and on-video). We would ideally want non-destructive overdubbing where the new recordings are saved and indexed into the overall structure of the recordings, but without destroying the original recordings. This requires the support of an appropriate naming, versioning and browsing system. We might also want to add and remove online content from the recordings, for example, adding in new online players or removing extraneous existing ones.

Animating and rendering

The final part of the production process is final offline animation and rendering.

- We might export selected parts of the recordings (e.g., the movements of two players who are involved in a particular chase scene) into a conventional animation package so as to create a higher quality rendered animation of the action (e.g., with more detailed visual representations and movements and carefully controlled virtual cameras).
- We need to composite the graphical and video material. This might be done using relatively simple techniques such as cuts, cross-fades and inset views or

might exploit augmented reality rendering techniques in which avatars are seen to move through video material (or video is embedded in a virtual model), possibly with lighting being adjusted accordingly. Audio postproduction also needs to be considered.

Early prototype production tools

We have carried out some initial practical explorations of key aspects of the production process. This has involved building prototype tools to review recordings, automatically analyse and visualise their contents (in terms of key scenes and events) and export selected parts of the recordings into animation packages (3D Studio Max).

An initial test application

To initially trial different techniques and ideas we required a system to very simply view what was taking place within a recording made from a mixed reality game. The recordings in question are from the mixed reality game Can You See Me Now (CYSMN) when it took place in Rotterdam in early 2003.

The mechanism to view the data is a stripped down version of the orchestration interface used during the event in Rotterdam itself. This interface can be seen in figure 3 where a number of runners (performers on the street whose position was given by GPS) and players (online members of the public whose position is given by their location within a virtual environment) can be seen overlaid on a map of the game area. Text messages that were sent between online players are also displayed at the bottom of the interface.

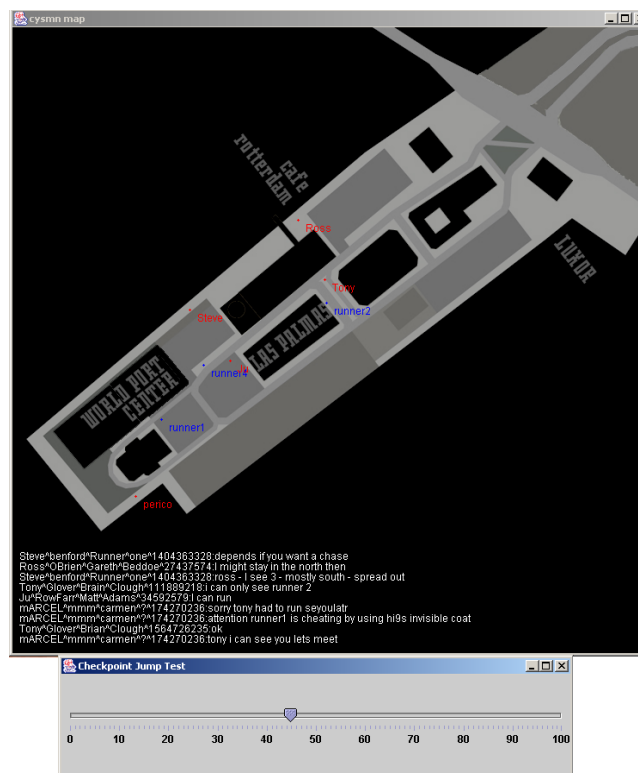


Figure 3: Reviewing a recording of CYSMN in Rotterdam

To allow the manipulation of the replay of the recordings, a simple slider bar is provided. The position of the slider represents the current point in time in the

recording being replayed and the user can move the slider to dynamically change this. The initial basic recording that was made during the live event consisted of all the events that took place within an Equip dataspace being logged to a file along with a time stamp giving the time the event occurred. These basic logs were then subsequently augmented with checkpoints (key frames) to allow any point within the recording to be selected by the user and the replay system is then able to jump to the nearest checkpoint and begin playback from that point. This is required as Equip is a state-full system and simply starting playback of the record from any given position (without using checkpoints) would not result in a dataspace that was consistent with the initially recorded dataspace.

Automated scene Extraction

From this point an application originally written for the MASSIVE-3 platform was ported to Equip, to allow the identification of scenes within the recording. This is achieved by periodically analysing the positions of the participants in order to extract information about momentary clustering of participants, and then comparing these clusters over time in order to extract information about ongoing scenes, including their begin and end times. This extracted scene data can be seen in figure 4 as white circles with the radius of the circle reflecting how many participants are regarded to be within that particular scene at the current moment in time. Ultimately we wish to extend this technique to take into account other factors beyond raw position such as direction of movement (e.g., is one participant chasing another?) or type of participant (e.g., is this an encounter involving a runner and a player?). The scene extraction software can be used in this way to provide hints as to where interesting activity may be taking place as a recording is replayed and reviewed, for example by an editor in TV postproduction.

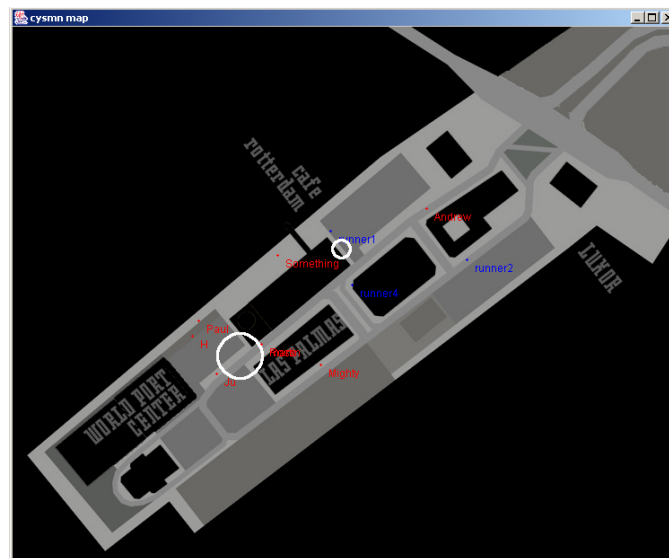


Figure 4: Automatically generated scene markers overlaid on the replay

Visualisation of Scene Data

Our scene extraction tool can be used in an offline mode where the recorded data is played back as quickly as possible through the same scene extraction algorithms. The extracted scene data is then saved to disk in XML format giving a chronological list

of where and when scenes took place. Offline analysis also detects some key game events (currently we only detect the event that signals a player was caught by a runner) and saves these game events, along with the scene event data for use by other software. Our next prototype tool is a piece of software to allow the visualisation of this scene overview data. To achieve this, the scenes are drawn in the style of a Gantt chart, where upon time is shown across the x-axis and each individual scene is given a separate bar descending down the y-axis (see figure 5). Also upon this chart we are able to show paths different participants took through different scenes and to show where key game events took place, such as caught events shown as red diamond shaped markers. The position of the markers is established by looking at the data within the caught event to establish which participants were involved and then to place to marker within the scene where those two participants were members.

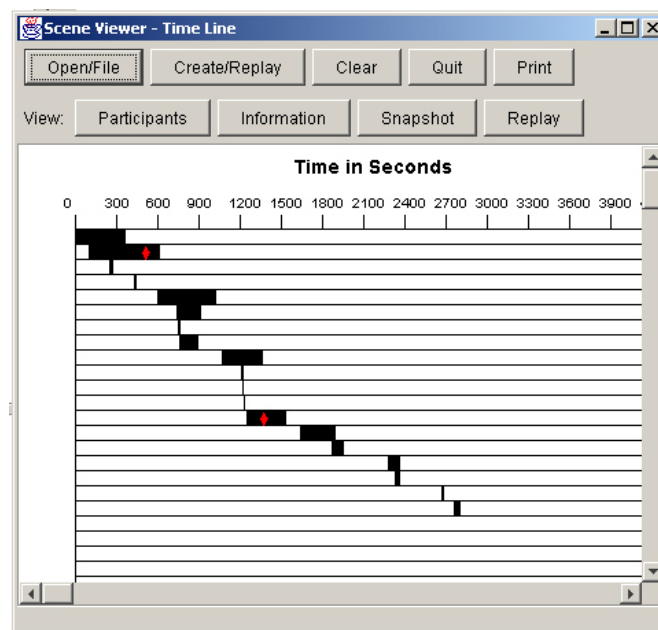


Figure 5: Viewing scenes and capture markers

This interface can also be used to select scenes to be reviewed using the 2D management interface discussed earlier, or exported to 3D Studio Max for final rendering. To export the data to 3D Studio Max, the positions of all the participants (or if required a subset of the participants) are saved to a text file along with time stamps.

Importing Replay Data Into 3D Studio Max

Character studio is a plug-in for 3D Studio Max that facilitates humanoid animation. One method for importing movement data is to create a “footsteps” file. This file contains the information about a sequence of footsteps necessary to drive a bipedal motion simulation. This creates a realistic animation of a skeleton character progressing down the specified path. The skeleton is normally bound to a mesh to drive the final animation. Artists can layer additional movements on top of the automatically generated motion to add details and subtleties to the walk.

Each footprint has the following properties: a relative location, a rotation (describing the direction the foot should face), and timing information (the duration of footprint as a whole, together with the time the foot is in contact with the ground). In addition

there are several scale parameters the effect the animation as a whole. These allow motion originally created for a biped of a certain size, to be imported onto a biped of a very different size, whilst still maintaining the relative footstep motion.

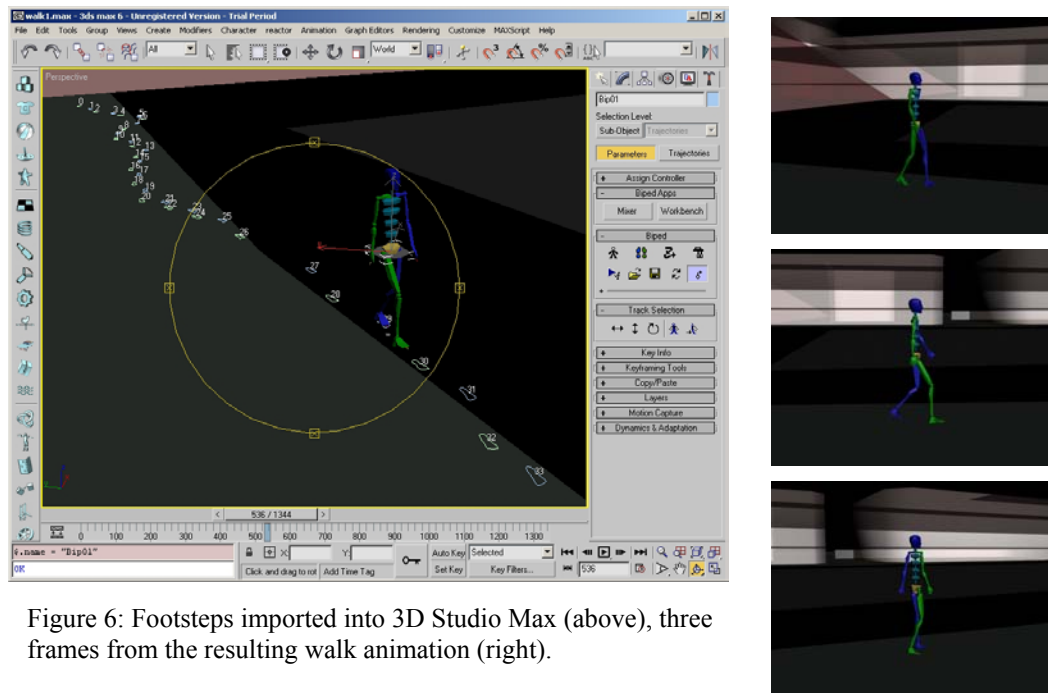


Figure 6: Footsteps imported into 3D Studio Max (above), three frames from the resulting walk animation (right).

Summary of requirements for record and reuse

Having explored the production process for mixed reality television shows and also demonstrated some early prototype production tools, we can now draw out some of the requirements that this application raises for underlying record and reuse infrastructure.

- Record online canonical view – we need to make recordings of the online system state (movements, text messages etc) as seen by a central server that is considered to provide the canonical view of the experience.
- We need to make video recordings with instrumented cameras so that we can cross index them – both temporally and spatially – with the system state recordings.
- We need to be able to analyse the recordings to identify potential scenes of interest.
- We need to be able to jump directly to key moments in the recordings and then repeatedly replay a short scene, viewing it from different (virtual and physical) camera positions.
- We need to be able to overdub some part of the recordings. This requires being able to replay the current recording, act against it and save a new version. This needs to be non-destructive.

- We require the ability to edit the recordings, for example subtly changing the trajectory of an online player through the model, to remove certain players entirely, to edit down larger recordings into smaller ones (in much the same way tradition audio/visual media would be) and to possibly merge or join recordings together.
- We need to export selected parts of the recordings (both in terms of selected times and participants) in a format that is useful to animation packages.
- We need access to the state of given types of item at a given point in time. Maybe in a similar fashion to Equip's current pattern matching system, except a time is also specified.

References

Benford, S., Greenhalgh, C., Craven, M., Walker, G., Regan, T., Morphett, J. and Wyver, J., Inhabited Television: broadcasting interaction from within collaborative virtual environments, *ACM Transactions on CHI*, December 2000, ACM Press.

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

Craven, M., Taylor, I., Drozd, A., Purbrick, J., Greenhalgh, C., Benford, S., Fraser, M., Bowers, J., Jää-Åro, K., Lintermann, B, Hoch, M., Exploiting Interactivity, Influence, Space and Time to Explore Non-linear Drama in Virtual Worlds, *Proceedings of CHI'2001*, 30-37, Seattle, US, April 2-6, 2001, ACM Press.

Greenhalgh, C., Flintham, M., Purbrick, P., Benford, S., Applications of Temporal Links: Recording and Replaying Virtual Environments, *Proceedings of IEEE Virtual Reality VR 2002*, Orlando, Florida, 2002.

Greenhalgh, C., Purbrick, J., Benford, S., Craven, M., Drozd, A. and Taylor, I., Temporal links: recording and replaying virtual environments, *Proceedings of ACM Multimedia 2000*, L.A. October 2000.