

July 10, 2009

RT0860
Human-Computer Interaction 8 pages

Research Report

A Method of Recording Navigable Movies in 3D Virtual Worlds

Tatsubori Michiaki, Akira Koseki

IBM Research, Tokyo Research Laboratory
IBM Japan, Ltd.
1623-14 Shimotsuruma, Yamato
Kanagawa 242-8502, Japan



Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Limited Distribution Notice

This report has been submitted for publication outside of IBM and will be probably copyrighted if accepted. It has been issued as a Research Report for early dissemination of its contents. In view of the expected transfer of copyright to an outside publisher, its distribution outside IBM prior to publication should be limited to peer communications and specific requests. After outside publication, requests should be filled only by reprints or copies of the article legally obtained (for example, by payment of royalties).

1. Introduction

This approach proposes a method for recording navigable movies, with which one can navigate within a movie, in a 3D virtual world like Second Life.

Existing technologies for navigable movies like iCinema's products [http://www.icinema.unsw.edu.au/projects/prj_convostudio_4.html] can be regarded as an extension to Apple's QuickTime VR and require for a navigable movie to record in a 2D movie format like MPEG2 where the cylindrical or spherical panorama view of the recorded target at a moment are captured in each frame of the 2D movie. A movie player allows a human movie viewer to change the direction of camera to watch different angles in the panoramic movie. With a QuickTime VR approach, navigable positions of a camera are fixed or very limited as it requires different set of a movie for each position of the camera.

Another naïve approach to record a navigable movie in a 3D virtual world is to record all the 3D models and moves available in the virtual world of the recording target for the recorded duration. It requires a large amount of memory space and computation cost for copying and storing all the 3D models especially when there are lots of complex 3D models.

Contrary to those existing methods, in our approach, we efficiently record a movie with the following steps.

1. In a recording step, a recorder system roughly and conservatively determines the 3D objects needed to record according to a given navigable range. For the quick perpetuation of the objects, it marks objects in the virtual world instead of storing the copies of them while actually recording any movement of the objects.
2. In a record keeping and trimming step, a perpetuator system interrupts any modification action on marked objects. A trimmer system periodically disposes unneeded marks or redundantly saved objects.

By these steps, this approach enables reducing memory space for recording the shape data of a number of 3D objects.

2. Background and Motivation

2.1 A 3D Virtual World

A 3D virtual world allows multiple users to operate on the objects in the world and to see the rendered 3D models of the world as if they were there. **Figure 1** depicts the simplified architecture of an existing 3D virtual world like Second Life. The virtual world is maintained by a server (3D World Server) and the 3D models of objects in the world are stored in some storage connected to the server. Multiple clients access the server through a network like Internet to provide their human users with 3D virtual world experiences through their graphical user interface. The users can directly operate on the virtual worlds by creating, deleting, and modifying 3D objects. They can also do so indirectly by taking some actions through their avatars or commanding interfaces which raise some events that existing 3D objects with programmed scripts may react to.

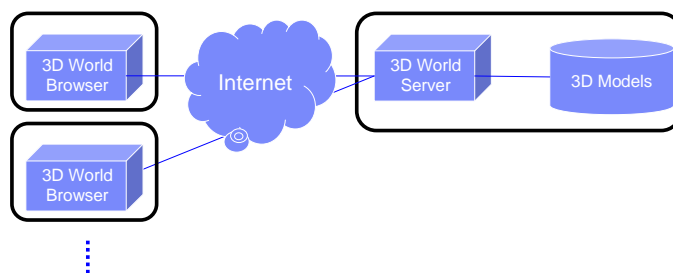


Figure 1. The simplified architecture of an existing 3D virtual world.

The 3D objects in the world periodically changes their position, direction, and shape according to some events raised by users' operations, the system's timer, and interaction among themselves. For example, a user may control its avatar in the world, which itself is 3D object(s), to move in the world with the gesture of walking. The avatar may hit a ball object, which moves according to a virtual physical formula set by scripts attached to the object or the world. The ball may hit other objects like boxes and these objects may react to the event of hitting.

2.2 A Navigable Movie

First we overview a navigable movie system in such a 3D virtual world. A navigable movie is a movie within which one can navigate through the past virtual world as illustrated in **Figure 2-1**. **Figure 2-2** depicts the architecture of a 3D virtual world with an invented navigable movie recorder. A movie player (3D Movie Player) allows a movie browser (3D Movie Browser) to change the direction of camera to watch different angles in the panoramic movie. With existing navigable movie players like QuickTime VR, navigable positions of a camera are fixed or very

limited but the invented movie player allows its user to change the camera position as if one is moving in the world.

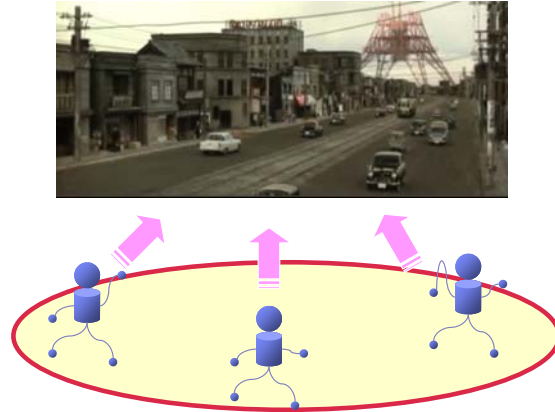


Figure 2-1. A navigable movie where a user can change the camera position and direction.

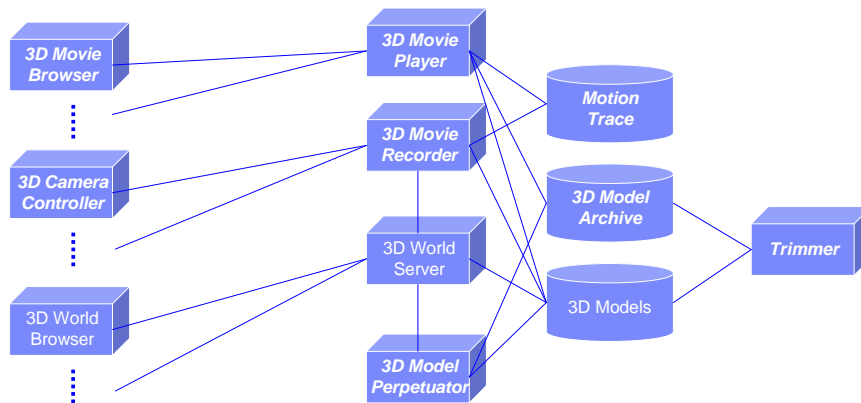


Figure 2-2. The architecture of a 3D virtual world with an invented navigable movie recorder.

2.3 Our Motivation

A naïve approach to record a navigable movie in a 3D virtual world may record all the 3D models and moves available in the virtual world of the recording target for the recorded duration. It requires a large amount of memory space and computation cost for copying and storing all the 3D models especially when there are lots of complex 3D models as illustrated in **Figure 2-3**.

Our motivation is thus to effectively record and play navigable movies.

- The **recorder** traces any movement of the 3D objects in the navigable range while recording, which starts and stops according to the user's instruction, and stores the record in storage (Motion Trace).

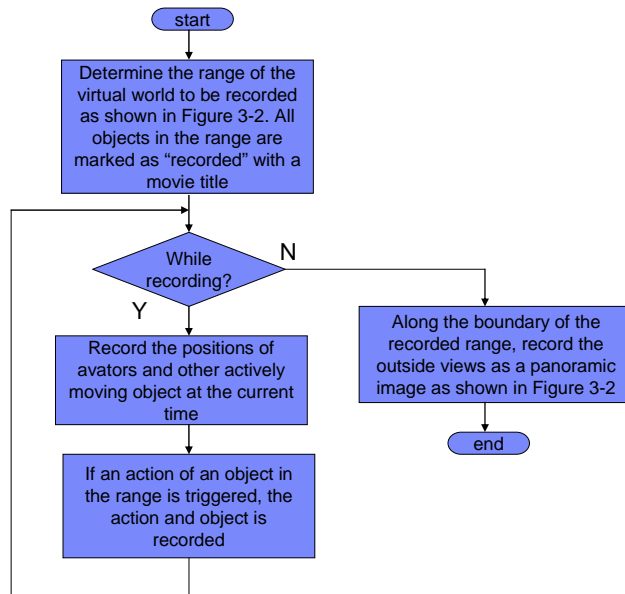


Figure 3-1. The flow chart of the recording steps.

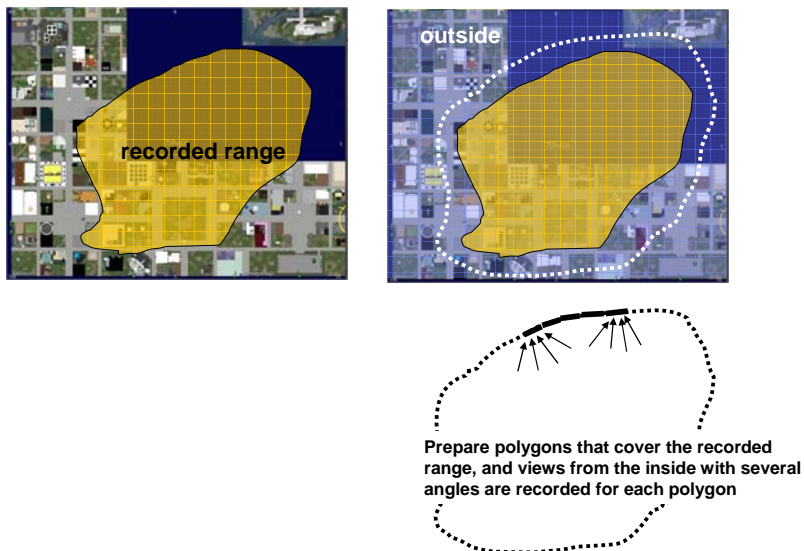


Figure 3-2. Recording range.

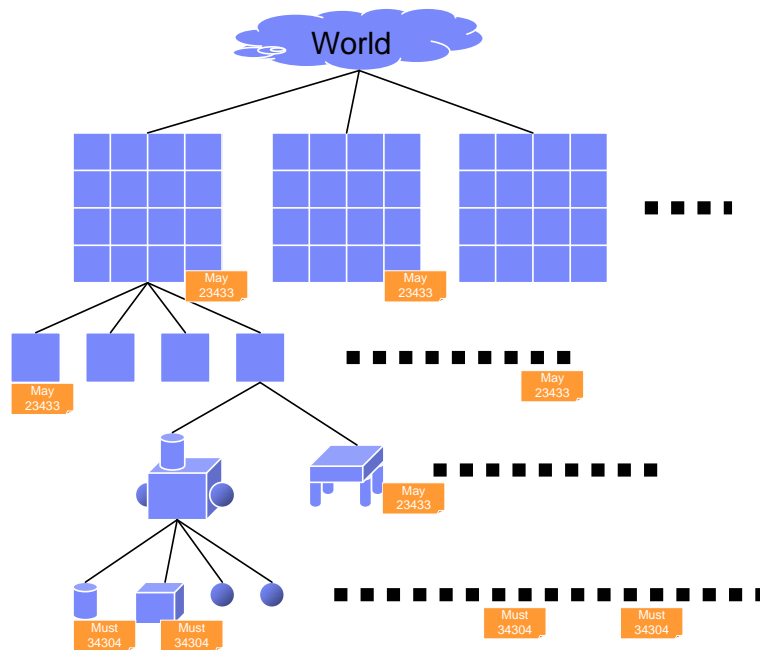


Figure 3-3. Roughly marked objects.

3.2 A Keeping and Trimming System

1. When someone modifies the shape of an active 3D object in the virtual world, a **perpetuator** system (3D Model Perpetuator) checks if the object is marked as recorded.
2. If the active object is marked as recorded, the **perpetuator** creates a copy of the object to replace the object with the copy, and then saves the original object (as a veteran 3D object) in a movie archive space (3D Model Archive).
3. Periodically, a **trimmer** system (Trimmer) more precisely but incrementally checks if active 3D objects roughly marked as recorded really need to be recorded to remove the redundant marks from the objects.
4. Similarly, the **trimmer** checks if veteran 3D objects roughly marked as recorded really need to be recorded to discard the redundantly saved objects.

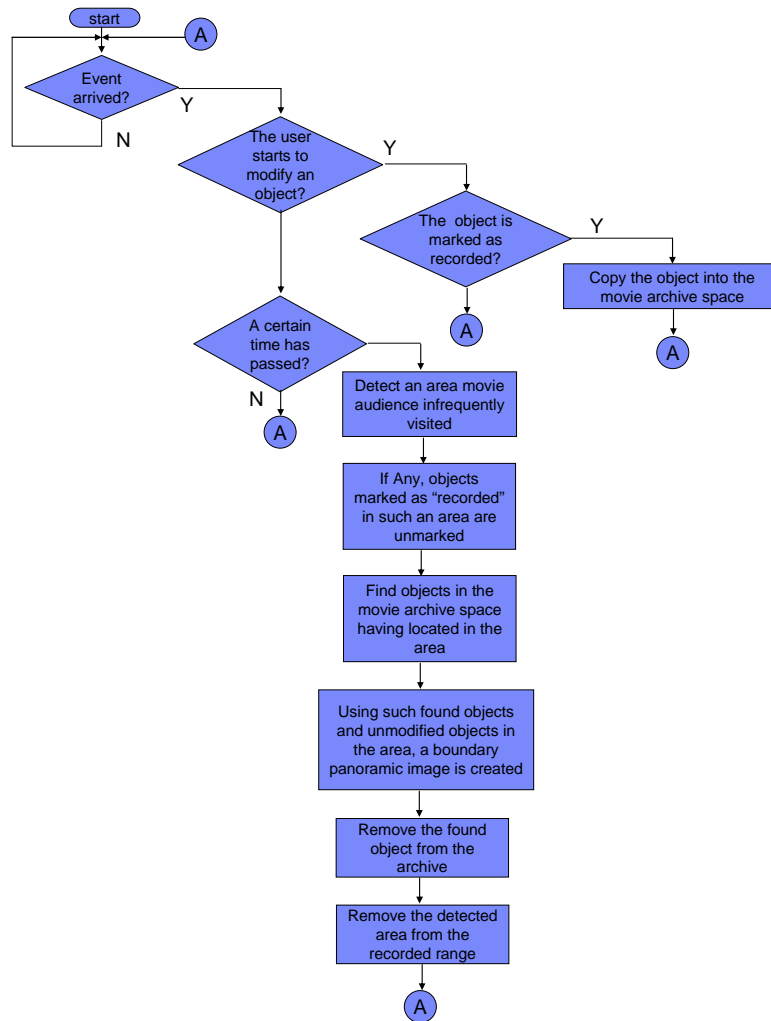


Figure 4-1. The flow chart of maintenance steps.

3.2 A Playing System

1. A 3D movie **player** system allows its user to start and stop a navigable movie through a remote 3D movie **browser** system and allows some control.
2. While playing the movie, the **player** allows controlling a camera for the browser view of the 3D movie, by changing the position, direction and zoom of it in the range allowed when recording the movie.
3. According to the camera view and the current playing time, the **player** determines the 3D objects needed to render by calculation based on the recorded activities.
4. If an object to render is in the archive, the **player** retrieves the 3D object model of the object from the archive storage otherwise it retrieves the data from the active virtual world.
5. The **player** may send the 3D models of objects to the **browser** but it may not if they

were already sent to it.

6. The **player** sends the movement of 3D objects to the **browser**.
7. The **browser** sets up the 3D objects and renders them according to the camera view.

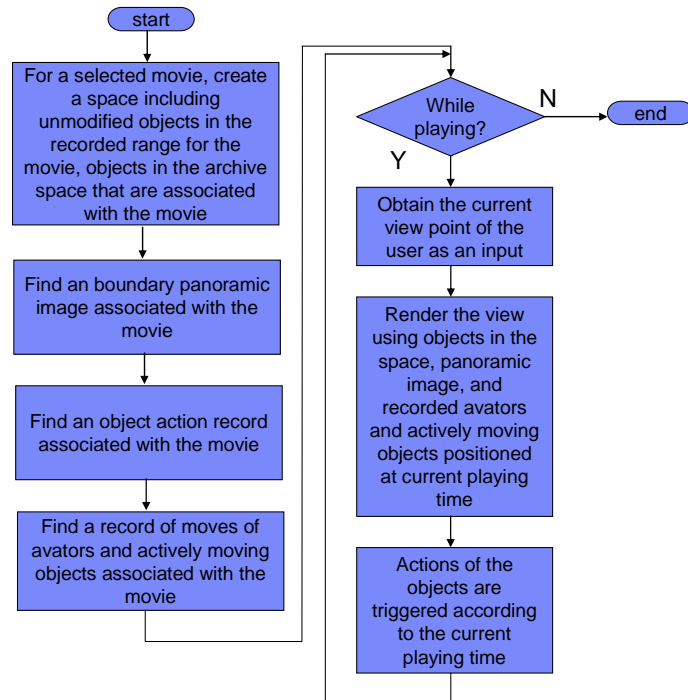


Figure 5-1. The flow chart of playing steps.