

Virtual Presence Mixing 3D Video and Interactive Scenes

**Felipe Carvalho, Peter Dam, Luciano Soares, Alberto Raposo,
Eduardo Corseuil**

Tecgraf – Computer Graphics Technology Group / Department of
Computer Science
PUC-Rio – Pontifical Catholic University of Rio de Janeiro, Brazil
{kamel}, {peter}, {lpsoares}, {abraposo}, {thadeu}@tecgraf.puc-rio.br

Ismael dos Santos

Petrobras Research Centre -
CENPES, Rio de Janeiro, Brazil

ismaelh@petrobras.com.br

Abstract

High quality non-interactive 3D Videos are largely used to document projects, present innovative products and ideas, and to explain complex concepts. On the other hand, in spite of its highly interactive characteristic, virtual reality (VR) suffers due to the non-photorealistic image quality. This research presents an approach for combining 3D rendered videos with interactive scenes. The goal is to overcome the lack of realism in VR models and improve presence and expectation in simulations.

1. Introduction

Professional videos and visualization tools are used in several applications like training, design review among other areas. Our main focus of research is to propose an architecture to organize a visualization process pipeline mixing video and geometric data. It is clear that not always the user wants to navigate in a 3D scene, many times he just wants to animate through an already defined path, and at some places play with some object to test some possibilities.

Videos are a common technology quite useful to document projects, present products and teach science. It is very used in presentation rooms in several companies being simple to deal and easy to anyone playback. High quality VR interactive content allows users to chose different viewpoints and manipulate the environment, but it needs special software and often more advanced user interface devices.

Each one has its advantages and disadvantages [1]. Videos are simple and have high quality images, but follow a linear timeline and lack interaction. Interactive tools, on the other hand, are not photorealistic. If we intend to interact in large models, such as CAD data, for instance, we have to face an even more complex problem since these data sets are quite detailed and take too much processing to render an image in real-time. We propose rendering complex scenes in video and interact just in simpler and located scenes. In order to

interact with the scenes, several features were implemented allowing the user to manage the scene, like moving objects and cameras, start audio clips, launch animations, among other possibilities.

The screenplay in our approach will be defined as a graph, due to the possibility of following different paths to reach one or more intermediate stages and endings, constantly changing from interactive scenes to videos, it is even possible to produce a cyclic graph. We are calling this solution as “interactive video” and this term will be used across the document. One big difference in this system is the non-linearity. It is possible to skip among the videos and interactive scenes, letting the user decide what he wants to do next.

This document is organized into the following sections. Section 2 presents related works about the use of videos and interactive scenes. Section 3 shows the ideas of presence and expectation that are the key elements that lead us to start this research. Section 4 explains how the interactive video is produced. Section 5 discusses the main algorithms and techniques applied in the research. Section 6 present the results achieved. Conclusions and future work follow in section 7.

2. Related Work

Some applications are available to couple dynamic content, like animations to interactive scenes. VRML [2] for instance is a common environment to interact in virtual reality scenes, but video visualization is limited, usually achieve by playing a short video in a polygonal surface. 3D multimedia tools like Acrobat 3D [3] and Director [4] allow this mix between different contents. They can be quite useful to document projects but do not cover the visualization of complex models the industry needs, such as massive industry CAD models.

The idea of using 3D movies and interactive scenes is much explored in the game industry. Many of best-selling video games like Gran Tourismo [5], Final Fantasy [6] and Halo [7] use off-line videos with several special effects and complexity together with the interactive game in lower quality to achieve the game

expectative of higher quality and maintain high frame rates. Nevertheless, the application of these resources in industry is not explored. It is possible to find very professional videos and powerful interactive tools to visualize, but they are not really coupled.

Since the industry uses stereo projection system to visualize their 3D models, it is important to cover this technology. Stereoscopic videos or simply 3D movies are not easy to produce and visualize. Some players are stereo capable, but using an external solution will lead to a loosely coupled solution, and then we decided to reuse an internal development capable to playback stereo videos [8]. The stereoscopic interaction in 3D scenes can be achieved in some tools, most of them commercial like SmartPlant Review [10] and EON Reality [11], but the same problem arises: using an external tool will reduce the control over the solution flow. We decided to support interactive scenes across OpenSceneGraph [9], that is free available and supports our basic needs to produce graphical primitives and interaction. Several features were implemented in the scene graph in order to achieve the interactive behaviors desired for this research.

3. Presence and Expectation

Augmenting the involvement during a virtual experience by holding the attention of the users in conscientious or unconsciously ways can guarantee the success of VR experiences. Presence is closely related to the concept of tele-presence, which term was coined by Marvin Minsky [12].

The present work is under the subjective definitions of presence following the works of Winter and Nunez [13, 14, 15]. Winter pointed two psychological states as the base to feel presence: immersion and engagement [13]. The meaning of immersion is related with the feeling of “being involved”. Engagement is resulted by the attention on a set of events and stimulus.

Besides the subjective meaning of immersion in Winter, a rational definition can be noticed in Slater [16]. Here immersion is defined in the sense of something that is measurable and is related with the technology used to create the sensation of presence.

Nunez [14] redefined presence using the concepts of realism and immersion. The amount of user information necessary during the experience is defined as immersion, and the concept of realism is related with the user's expectations. These definitions, pushed the meaning of presence to a different perspective, where an inference process resulted from the information in the environment is also important. The

information in the environment is also important. The experience here is not only a process of decoding sensorial stimulus, but also something that is being constructed based on expectations. Something is considered realistic if there is a match with a previous expectation. In [15] some experiments were made, following these ideas related with expectation, and positive results were noticed. The experiments aimed the creation of expectations in the subjects using materials before the virtual experience. Better levels of presence were noticed in the group of subjects that read the materials before the experience.

Given the positives results using the idea of creating expectations to conduct a virtual experience and to induce presence, it would be valid to search attractive and rich ways to induce presence during virtual experiences. Photorealistic videos plus 3D interactions would be a different approach from the traditional ones. Using these videos in the sense of creating expectations to better conduct the 3D interaction scenes would increase user's attention.

4. Composing Presentation

The compositing process is divided into two parts, the macro-level and the micro-level. This process consists of a connection of a large variety of elements, to produce the desired scenario flow.

4.1. Macro-Level Visualization

The macro-level contains the play list of videos and interactive scenes and their order to be interpreted by the player. The interactive scenes and videos are presented as nodes in the graph with one entry point and exit points. Nodes are associated to each other by links. A module tool was developed to enable users to connect the nodes as they want (Figure 1) in a simple graphical interface.

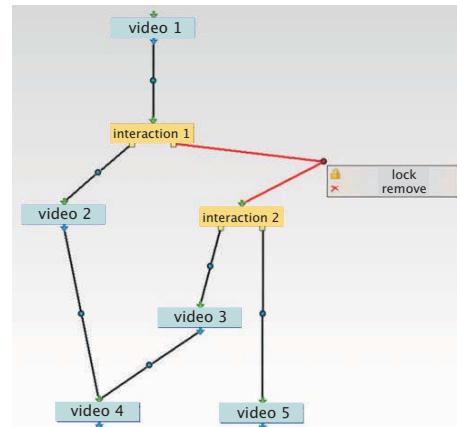


Figure 1. Macro-Level Example Scenario Flow.

This tool allows the interactive video designer to create nodes and to link them according to the desired screenplay. One node must be defined as the starting one. Although more than one ending may exist, we always need to define a starting node.

A video node represents a pre-rendered video. This type of node allows infinite connections into the entry point, but only one connection can exist in the exit point. This means that a video node can only lead to just another node. Note that several videos can be put in sequence, working pretty similar to traditional video edition tools.

An interaction node represents a real-time interaction simulation, thus allowing the spectator to interfere with the content of that scene. An interaction node allows multiple connections into the entry point, as well as multiple exit points connections. The nodes can be organized in such a way that the spectator feels as if he has the power to change the video. The interactions can lead to different exit points, although, must be programmed previously in the micro-level.

The macro-level information can be stored in a XML file. It is possible to define each node configurations, how they are connected, and which one is the starting node. Additional information such as the stereo configuration (mono, anaglyph, active etc), and the links positions are also stored in the macro-level XML file. A macro-level file example is presented in Code 1.

```
<Graph>
<Stereo ConfigFile="settings.xml"/>
<Nodes>
  <Interaction Name="SC01" StartNode="true" File="M1dxm" Pos="10 10"/>
  <Interaction Name="SC02" File="MV01e.xml" Pos="10 20"/>
  <Video Name="Vid01" File="VideoPlayerParams_01xml" Pos="1030"/>
  <Video Name="Vid02" File="VideoPlayerParams_01xml" Pos="1040"/>
</Nodes>
<Links>
  <Link SourceNode=SC01 Pipe="0" TargetNode="Vid01" Pos="10 15"/>
  <Link SourceNode=Vid01 TargetNode="SC02" Pos="10 25"/>
  <Link SourceNode=SC02 Pipe="0" TargetNode="Vid02" Pos="10 35"/>
</Links>
</Graph>
```

Code 1. Macro-level example.

4.2. Micro-Level Visualization

The micro-level defines the internal behavior of videos and interactive scenes. Video micro-levels defines the video file to be read, and specify the start time and time length, since we could just playback part of the video and not the whole content of a video file.

Interactive scenes are an event-action approach to describe behaviors. The set of behaviors will be responsible for the interaction flow during the virtual experience. There are several options, and then an

editing module was produced for the Autodesk 3DStudio Max, to easily produce interactive micro-level scenes.

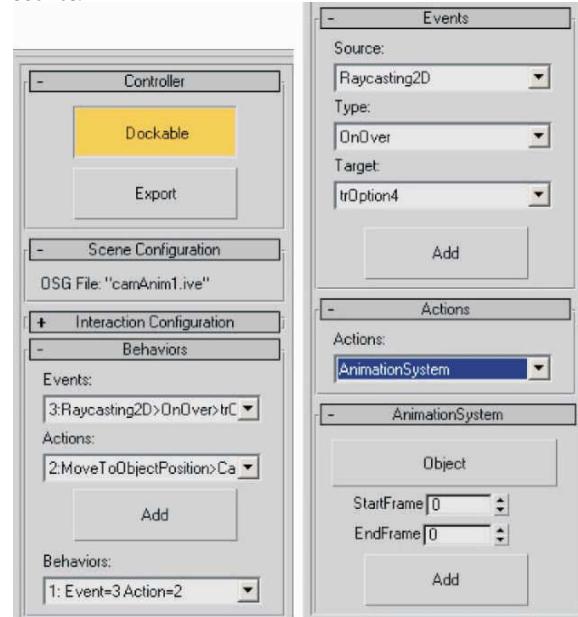


Figure 2. 3DS Max Plugin.

Once our scene is modeled in the 3D modeling application, we must program the behaviors using techniques defined in the system. A scripted plug-in was created for 3DS Max allowing to create events and actions based on pre-programmed techniques and associating these to objects.

With this part programmed, the plug-in exports a micro level XML that holds the configuration and a model file in the OSG IVE file format. The micro-level files are referenced by the macro-level. The micro-level XML is much more extensive than the macro-level, due to the large amount of events and actions. A micro-level file example for an interactive scene is presented in Code 2.

```
<Scene>
<SceneGraphicNameFile=".ivefiles/camAnim1ive"/>
<InteractionTasks>
  <Selection DescriptionFile="SelectionByRaycasting.xml"/>
</InteractionTasks>
<Events>
  <OnOver Id="1" Source="Raycasting2D" Target="trOption1"/>
  <OnClick Id="2" Source="Raycasting2D" Target="plOption4"/>
  <OnStartScene Id="3" Source="Scene"/>
  <Finished Id="4" Source="MoveToObjectPosition" Target="CamOP1"/>
  <Finished Id="5" Source="AnimationSystem" Target="trOption4"/>
</Events>
<Actions>
  <AnimationSystemId="1" Object="trOption1" StartFrame="10" EndFrame="50"/>
  <AudioSystemId="2" Object="CurrentView" Audio="hover2.wav" Loop="false"/>
  <MoveToObjectPositionId="3" Source="CurView" Target="Cam1" Frames="100"/>
  <Exit Id="4" Pipe="0"/>
</Actions>
<Behaviours>
```

```
<Behaviour ID="1" Event="1" Action="1"/>
<Behaviour ID="2" Event="3" Action="2"/>
<Behaviour ID="3" Event="4" Action="3"/>
<Behaviour ID="4" Event="2" Action="4"/>
<Behaviour ID="5" Event="5" Action="5"/>
</Behaviours>
</Scene>
```

Code 2. Micro-level example.

Due the long list of events and actions, we are not going to publish all the possibilities. Here we are just presenting the basic ones. Although, more complex nodes and connections exists, achieving very sophisticated effects.

5. System Architecture

The macro-level is interpreted and used as reference to connect the micro-level nodes. As the start node is detected, the player runs the node interpreter depending whether it is a video or an interactive scene.

Videos are going to be played using the start and length parameters. The interactive scene micro-level is interpreted and a listener mechanism is created to wait for events and trigger the required actions. Devices and interactions techniques are plug-ins of the system. When a new interaction technique is developed it is directly ported to the solution. The interpreter is also a plug-in, responsible to send parameters for the interaction techniques when an action needs to be executed. The interpreter is also responsible to start and finish every plug-in used during the interaction.

The plug-ins exchange messages between them across a message manager that handles the message transport. The messages can carry basic types and pointers to the scene graph nodes (Figure 3).

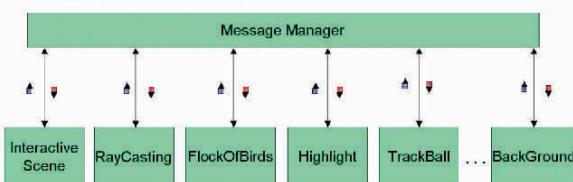


Figure 3. Message Transports.

The message manager is used to control communications of the macro-level (Figure 4). The GManager plug-in start and finish the interactive scenes and the videos. Having all interactive and non-interactive possibilities coupled increases the flexibility of the solution enabling the user, for instance, to cancel a video and skip to the next interactive video node. The structure of these message managers is formed by a message collector which at every frame search for messages by asking all current plugins by asking all current plugins.

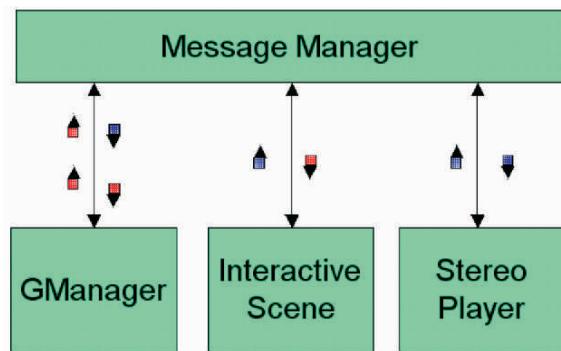


Figure 4. Main Message communication.

The interpreting structure created to handle the interaction flow is composed by a sequence of small trees (Figure 5). Each tree has an event source as root node. On the second level there are types of events that the source emits during the interaction. On the third level there are targets and finally on the last level there are the actions to be executed.

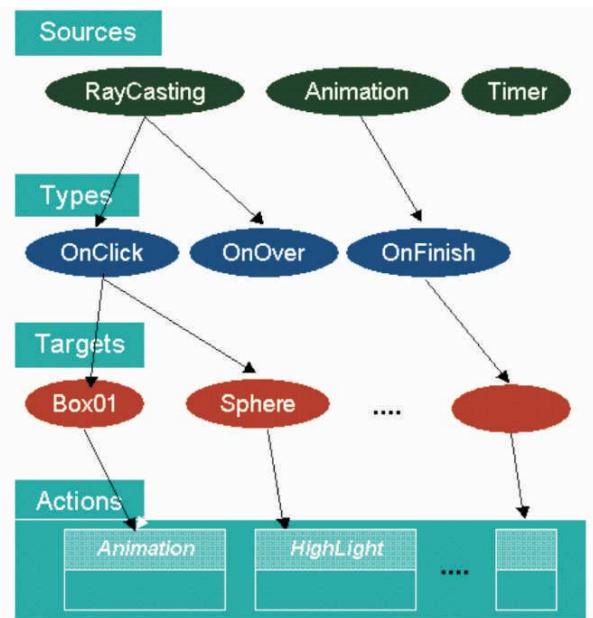


Figure 5. Interactional Flow tree.

The scene description file of the micro level contains 5 basic sections:

- 1) *SceneGraphFile*: Contains the OpenSceneGraph file exported by the 3dsmax. In this file there is information about geometry, materials, light, camera, and animation;
 - 2) *InteractionTasks*: This section acts like an include file directive founded on common programming languages. The included XML files contains

descriptions of interactive tasks using events, actions and behaviors too. Inside the file, that included an interaction task XML file, a label references that task and it is considered as an event source. Example:

```
<Selection
  DescriptionFile="SelectionByRaycasting.xml"/>
```

The label “Selection” is considered as the event source for all the events created inside the file SelectionByRaycasting.xml;

3) *Events*: Contains the list of events. Each one contains the source where it was created, a type and a target object. Events can be triggered inside the techniques by programming hard code, or can be created inside a XML file being triggered by others events;

4) *Actions*: Contains the list of actions. Each one has the name of the technique and a list of parameters to be used in the execution of the action;

5) *Behaviors*: Contains a list of associations between actions and events.

We have created actions and events related with object animation, 3D sound, interpolation of position, path following, object following, object highlight, 2D and 3D text, 3D icons, particles, background image, video stereo player, timers, trackball interaction, treatment of devices such as trackers (optical and a flock of birds), keyboard, mouse, and custom high level interaction tasks (selection, manipulation and navigation) by the association of atomic actions and events.

6. Results

Since we are from the computation area, and not designers, we were initially not completely sure how many interaction resources were important to produce interactive videos. We tested my cases like the one shown in Figure 6 and we defined and implemented a list of features enabling the video designer to produce interactive videos.

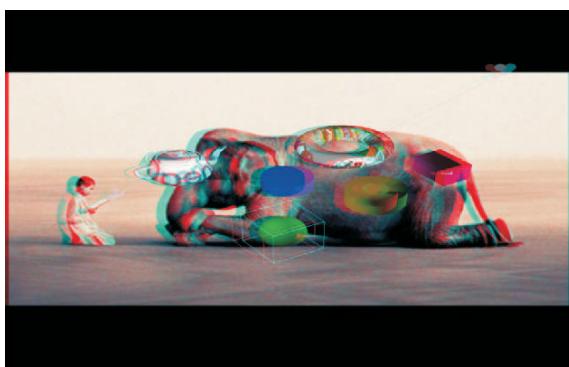


Figure 6. Interactive Video Test Scene.

This solution is being tested in VR facilities. The users evaluated are used to just see high quality videos (Figure 7), and our first observations indicate they get quite impressed when they can change the virtual environment.

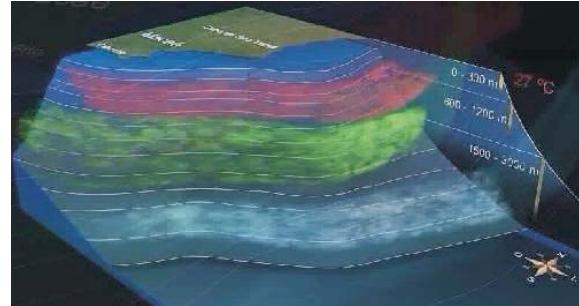


Figure 7. Video Scene.

People producing the interactive scene can see and test all the 3D behaviors inside 3DStudio Max. It is possible because a special player (Figure 8) was developed to assist this creation. This player opens in one of the 3DS views allowing the developer to navigate and test as if he was in an immersive environment.

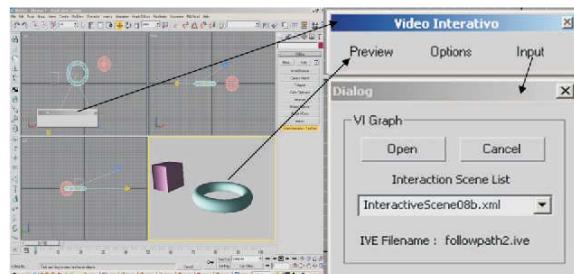


Figure 8. Special Player for 3DS Max.

7. Conclusions

This solution can impact the way people use virtual reality, from videos to real VR applications, increasing presence and not frustrating the audience since the VR quality is not the same as that of professional videos.

Since most of VR facilities in industry are based on planar projections, normal videos can be applied. However, if different projection surfaces are used, such as CAVEs, videos must be produced to fit these needs anyway. For instance, for a six-sided CAVE, it is necessary 6 videos, each one with the correct perspective for each face.

As a future work, we intend to include in the 3D editor dummy icon nodes, to inform the action attached to the scene. We are still studying ways to simplify the micro-level programming for designers, because this might be the toughest part of producing an interactive

video, since it needs a logical sequence to work correctly. We intend to use questionnaires to verify the levels of presence during the virtual experience. This method would be a more formal evaluation.

8. References

- [1] J.M. Sullivan, Rescalable Real-Time Interactive Computer Animations Multimedia Tools for Communicating Math., J. Borwein, M.H. Morales, K. Polthier, and J.F. Rodrigues, eds., pp. 311-314, Springer, 2002.
- [2] Web3D Consortium, VRML97 Functional specification, ISO/IEC 14772-1:1997.
<http://www.web3d.org/x3d/specifications/vrml/>.
- [3] Acrobat 3D,
<http://www.adobe.com/products/acrobat3d>.
- [4] Adobe Macromedia Director,
<http://www.adobe.com/products/director/>.
- [5] Grand Turismo Play Station game,
http://www.us.playstation.com/PSone/Games/Gran_Turismo/OGS/.
- [6] Final Fantasy game, <http://www.square-enix.com/na/title/ff/>.
- [7] Halo XBox game <http://www.bungie.net/>.
- [8] Raposo, A.B., Szenberg, F., Gattass, M., Celes. Visão Estereoscópica, Realidade Virtual, Realidade Aumentada e Colaboração. In: A. M. S. Andrade, A. T. Martins, R. J. A. Macêdo (eds.), *Anais do XXIV Congresso da Sociedade Brasileira de Computação*, Vol. 2, XXIII JAI - Livro Texto, Cap. 7, p.289 - 331. SBC, Brazil, 2004.
- [9] Burns, D. and Osfield, R. 2004. Open Scene Graph A: Introduction, B: Examples and Applications. In *Proceedings of the IEEE Virtual Reality 2004 (Vr'04) - Volume 00* (March 27 - 31, 2004). VR. IEEE Computer Society, Washington, DC
- [10] Intergraph SmartPlant Review,
<http://www.intergraph.com/smartplant/review/>.
- [11] Eon Reality, <http://www.eonreality.com/>.
- [12] Minsky, M., *Telepresence*, Omni 45-52, June, 1980.
- [13] Witmer, B.G., Singer, M.J., Measuring Presence in Virtual Environments: A Presence Questionnaire, *Presence*, v. 7, n. 3, 1998.
- [14] Nunez, D., How is presence in non-immersive, non-realistic virtual environments possible?, *Proceedings of the 3rd International Conference on Computer Graphics, Virtual Reality, Visualization and Interaction in Africa* , 2004.
- [15] Nunez, D., Blake, E.H., The thematic baseline technique as means of improving the sensitivity of selfreport scales, *Presented at Presence 2003*.