AN MPEG-4 BIFS-BASED AUTHORING TOOL FOR MULTIMEDIA CONTENT
S. M. Tran\textsuperscript{1,2}, M. Preda\textsuperscript{1}, F. J. Preteux\textsuperscript{1} and A. Gschwindt\textsuperscript{2}
\textsuperscript{1} ARTEMIS Project Unit, Institut National des Télécommunications, Evry-France.
\email{tran.son@int-evry.fr}.
\textsuperscript{2} Budapest University of Technology and Economics, Department of Broadband Infocommunication Systems, Hungary.

Abstract

In this paper, we propose an authoring tool referred to as ICCT (Interactive Content Creating Tool). The tool serves to easily generate as well as dynamically control a multimedia and interactive MPEG-4 content represented as a hierarchical scene graph. In the parametric mode of ICCT, scene-designer – using a user friendly interface for setting properties of objects in the scene - can create an enriched multimedia content with various types of objects: audio, video, image, text … Some predefined dynamic behaviors for certain types of objects are available, such as positioning the video clip in the scene, start / pause the media, … The maximum power of interaction between the viewer and the scene as well as between objects of the scene can be obtained with the programmatic mode of the ICCT. In this mode, with the aid of script, scene-designer can flexibly define a scene, whose final multimedia content can be presented to viewer in various manners depending on request from viewer and also on the mutual interactions between objects of the scene. The combination of the two modes makes ICCT useful for both novice and experienced scene-designer to exploit the power of scene-composition and interactivity provided with the recent MPEG-4 standard in order to create sophisticated video applications.

1 Introduction

The boom of the Internet together with the rapid development in digital store capacity makes multimedia applications more and more popular. As a feedback, this new technology and related uses also expose some new requirements on the quality of multimedia applications. Among these others, consideration shall be put on a higher compression of image / video, the availability of composing different media data and the support of interactivity between users and applications. The emergence of the recent MPEG-4 standard brings efficient solutions to these new challenges. As a member of the MPEG family, the MPEG-4 standard inherits and improves all the performance of its predecessors, offering the possibility of efficient transmitting and / or storing a huge amount of digital image / video. However, the standard jumps a great step forward with graceful key-techniques, such as advanced audio coding, object-based video compression, deployment of wavelet-based texture encoding and mesh-based representation. The most promoting feature of MPEG-4 is probably the concept of the multimedia scene supported by the definition of Binary Format for Scenes (BIFS) as a part of system information, acting as an intermediate layer between media data and the final built-up content. BIFS enables the construction of the scene graph where the nodes of the graph are the media objects. Thus, it provides a flexible way to manipulate various types of media in an MPEG-4 scene, such as scheduling, coordinating in temporal and / or spatial domain, synchronization, processing interactivity, and so on. With respect to the complexity of the scene graph – the number of the node types – MPEG specifies profiles and associated applications [9] that MPEG-4 can address. This mechanism allows easy and structured implementations of compliant decoders as well as authoring tools.

Syntactically, the MPEG-4 BIFS is largely based on Virtual Reality Modeling Language (VRML) [4], [5]. It adopts all the 3D nodes from VRML, and at the same time proposes several 2D as well as 3D nodes. BIFS can be considered as a binary encoded version of an extended subset of VRML, which can represent roughly the same scene as with VRML in a much more efficient way at the cost of computer-readable-only composition: binary representation of a multimedia scene. Therefore the so-called eXtensible MPEG-4 Textual format (XMT) is defined for representing MPEG-4 scene description by using a textual syntax. XMT allows authors to exchange their content with others, who may use different authoring tools. XML facilitates interoperability with both the Extensible 3D (X3D) being developed by the Web3D and the Synchronized Multimedia Integration Language (SMIL) from the World Wide Web Consortium (W3C) [11].

Unfortunately, the advanced feature of MPEG-4 derived from BIFS, \textit{i.e.} the composing functionality for creating an interactive multimedia scene, has not yet be focused well enough in recent applications. For instance, only the MPEG-4 audio video coding performance, having high quality at low cost of bit-rate is exploited in DivX applications [15] and players [14], [16]. Such audio / video applications are compliant with the “Simple 2D” scene graph profile. More complex and advanced BIFS feature is still unknown or unfamiliar to most of video service providers. As an effort to reduce the distance between service providers and the already available technology, we introduce here an authoring tool, called Interactive Content Creation Tool (ICCT). With an user-friendly graphic interface, the tool offers a high abstraction level of editing a multimedia scene without requiring any knowledge about the underlying system. With the aid of the script compiling add-in, ICCT also provides designer with a flexible and efficient way to create a scene supporting sophisticated interactivities. The resulting
scene in both mode is multiplexed together with other audio-visual material to create an MPEG-4 formatted file, which can be a target of playback, storing or broadcasting. This final multimedia content is compliant with the “Advance 2D” profile of MPEG-4.

The aim of our work here is twofold. First, we introduce an authoring tool, which aides service / application provider smoothly migrating from a simple conventional audio-video scene to a full interactive multimedia scenario. Second, also as a demonstration of the capability of the developed ICCT, we present the class of applications that the ICCT addresses by illustrating some typical interactive applications based on MPEG-4, through which we help readers having a broader look into the advanced features of the recent MPEG-4 standard. Our applications are according with standardized MPEG-4 profiles and we present them in an increased order of complexity.

The rest of the paper is organized as follows. The next section presents a brief overview of the MPEG-4 scene description in the framework of system information. In Section 3, we discuss in detail the structure of ICCT. With the aid of this tool, a large scale of enhanced video applications can be generated as described in Section 4. The enhancements can be either the well-known feature as movie with subtitle; or real breakthroughs for video application, such as parallel video streams, addition of graphics and interactive controls. The last section aims at evaluating the performance of such multimedia contents and outlining open perspectives for future developments.

2 Overview of BIFS structure and functionality in MPEG-4 scenarios

MPEG-4 addresses the coding of audio-visual objects of various types: natural video and audio objects as well as textures, text, 2D and 3D graphics, also synthetic music and sound effects. In order to reconstruct this rich multimedia content at a client’s terminal, it is not sufficient to transmit only the compressed audio-visual data to that terminal. However this is the most common case in the current implementations of the MPEG-4 standard (Figure 1-a).

Figure 1: Different scenarios for video applications supported by the MPEG-4 standard.
Additional information is needed for enriched applications in order to combine media data at the terminal, to construct and to present to end-users a meaningful multimedia scene. The emergence of BIFS is dedicated to take over this role. Similar to VRML, BIFS describes a scene with a hierarchical structure that can be represented as a graph. Nodes of the graph form various types of objects, such as audio video, image, graphic, text, etc. The structure is not necessarily static; nodes may be added, deleted or modified. The nodes themselves expose a set of parameters through which, aspects of their appearance and behavior can be controlled. For instance, the audio part of a normal movie application can be referred to as an AudioClip node, which is a child of Sound2D node. With the former node, one can choose the sound-track (for different language) of the given movie through its'url' parameter. With the latter node, the volume of the sound can be set through intensity parameter. To fully describe a multimedia scene, a collection of hundreds of BIFS nodes are standardized and continually extended [1]. From this rich resource, to create an arbitrary scene, a subset of nodes are deployed and encoded into an elementary stream. This BIFS stream is a separate but not independent stream. It may contains several so-called media nodes, taking charge of displaying video clips, audio clips, background image,...and therefore requiring further compressed audio / video data (media data) from other elementary streams. The url field of media nodes maintains this relation, which is outlined as dash arrows in Figure 1-c. At a clients' terminal, different types of information are extracted from MPEG-4 multiplexed stream or MPEG-4 formatted file and decoded by proper decoders. The resulting BIFS information, conveying the idea / intention of scene-creator, is involved first to reconstruct the scene. The media data is employed only when the associated media node is required by the scene. In other word, media data are no longer directly “visible” to viewers in BIFS enabled MPEG-4 application: what viewers see, hear and interact with, are exclusively BIFS nodes. An example of such a scenario is shown in Figure 1-c. With BIFS information, scene-designers now have a tool to force the scene reconstruction at the decoder to adhere to their intention. Furthermore, they can let viewers interact with BIFS node (modify nodal parameters) to a certain extend. The resulting multimedia scene is therefore driven by both service provider and viewer, which is impossible with the predecessors of the MPEG-4 standard.

3 Structure of ICCT

Although the implementation of complex BIFS information can offer a video application with enhanced interactivity as shown previously, no such feature exists in most of the current MPEG-4 files [15]. Most of the MPEG-4 players [14], [16] are only dealing with “Simple 2D” Scene Graph profile. This limitation is mainly due to the fact that a scene-designer can not describe a scene efficiently without a proper tool: he / she shall browse among hundreds of predefined BIFS nodes in order to find some proper ones for representing an auditory / visible event in a scene. This process is only possible if the designer has a deep knowledge about the underlying BIFS structure. The consequent requirement is not always and not necessary the case for service providers in general.

Our Interactive Content Creator Tool (ICCT) is used to generate and integrate BIFS information for an enhanced video application. The output of ICCT is an MPEG-4 formatted file, which can be played with BIFS enabling player or used as a target of broadcasting. The tool serves as a solution for the aforementioned problem regarding to the implementation of BIFS. With two operating modes: parametric and programmatic, ICCT proves to be a useful tool for both novice and experienced scene-designer.

3.1 Parametric operating mode

This mode of operation is dedicated for scene-designers without any knowledge of the underlying BIFS node. Hence, the term “parametric” means with purely adding / setting operations; the scene-designer can create a customized multimedia content. There are two main functional units: the Import media data and the Scene composition unit (Figure 2). Designer uses the former one to insert all necessary compressed media data (compliant to the MPEG-4 standard) to ICCT. These media data (they are the collection of tracks in Figure 3-a) create valid resources, which may be addressed by media nodes in the scene. The Scene composition unit – generating and encoding BIFS information - is again separated into two layers: the Graphic interface layer and the Scene node layer, which respectively correspond to interactions with author and to mapping into MPEG-4 scene graph nodes. In the upper layer, higher abstraction level of objects can be found, which are uniquely corresponding to auditory / visible events in the scene, such as a Video clip, an AudioClip, a Background image, a logo Graphic..... In the lower Scene node layer, these objects will be actually mapping to a hierarchical structure of several BIFS nodes and encoded accordingly to generate BIFS information. As an example, an AudioClip object (it is responsible for the sound of video application) created in the Graphic interface layer is translated to a Sound2D node, which in turn consists of an AudioClip node in its source parameter. The parameters of the AudioClip are applied to the connected counterparts, which may not belong to the same BIFS node; the volume of the sound is mapped to the intensity of Sound2D node while its start time constrains the startTime of AudioClip node. The predefined mapping methods make the BIFS layer transparent to designer, therefore he / she now can concentrate only on the temporal / spatial behaviors of the objects in order to
construct a multimedia scene (Figure 3-b). The elementary stream created from the imported media data and BIFS nodes are finally mixed into an MPEG-4 formatted file for latter playback, storing or broadcasting.

Figure 2: The two-layered structure of the authoring tool in parametric mode.

3-a: Import media data  
3-b: Setting temporal / spatial properties of audio-visual objects

Figure 3: The control panel of the ICCT in the parametric mode.

3.2 Programmatic operating mode

For experienced scene-designers, who are familiar with BIFS nodes and can work directly with Scene node layer, the previous mode of ICCT may restrict them to limited ways of mapping objects in the scene to BIFS nodes. Instead, they need such an authoring tool, which generates the binary encoded BIFS information from their own settings directly to BIFS nodes and multiplexing it with other media data into a MPEG-4 formatted file. The programmatic mode of the ICCT serves experienced scene-designer in this way. Figure 4 shows the functional blocks of ICCT in this mode. The Import media data unit remains the same as in parametric operating mode. The Scene composition unit now plays a function of script compiler: its input is a script-based description of BIFS nodes, its output is the encoded BIFS information. The syntax of script and its parser / encoder are employed from the BIFS encoder software of the MPEG-4 Working Group [3]. This mode maximizes the possible ways to produce various types of scenes. Together with the support of Script BIFS node in [3], ICCT offers designer a programmatic way of describing multimedia scene, which makes the interactivity available in multimedia scene. In the current version of ICCT, programmatic mode does not support the visualization of temporal / spatial nodes appeared in scripts. The scene temporal and spatial preview as in Figure 3-b is accessible only in parametric mode.
4 Usage of ICCT for creating various MPEG-4 applications

This section will demonstrate the use of ICCT to create several multimedia scenes. Regarding the complexity of the integrated interactivity, the resulting applications are classified into three groups: enhanced low bit-rate video applications, simple interactive videos and full interactive applications. Generating process of one member with typical features for each group will be discussed in detail.

4.1 Enhanced low bit-rate video applications

A typical application in this group consists of one video, one audio and a subtitle positioned at the bottom of the scene as in Figure 5. To generate this application with ICCT in the parametric mode, a designer shall follow steps below:

- Import MPEG-4 compressed audio and video stream into ICCT with the Import Media Data functional unit. The audio- and video stream must be compressed in compliance with the MPEG-4 standard. In our case, in order to playback the MPEG-4 formatted file with Envivio player [13], these audio / video streams have the syntax specified respectively in [6], [7].

- With the aid of the Graphic Interface layer, designer shall create several objects to construct the scene:
  - Video clip referring to MPEG-4 compressed video stream,
  - Audio clip referring to MPEG-4 compressed audio stream and several Text objects one-by-one referring to subtitle displayed at a certain time.

- Spatial and temporal properties of the objects shall be set to obtain the scene with a proper location of object in time and in 2D coordinate of the player’s screen. For instance, the spatial positions of the Texts shall be under the Video clip’s; the temporal position of the Texts (the startTime and stopTime for their appearing and disappearing time) should be synchronized with the time of their associated video frames.

- By saving the scene to file, designer activates the operation of the Scene node layer, which will map all the objects to proper hierarchy graphs of BIFS nodes for binary encoding. For example Video clip object will be mapped to a graph, whose root is a Transform2D node, this node consists of Shape node in its children parameter, the Shape node is in turn built up on Bitmap and Appearance node. The Appearance node is continually separated into Material2D, MovieTexture and TextureTransform node. The construction of this tree-structure is transparent to the designer, automatically processed in accordance with some predefined rules. Consequently, all properties set by the designer in the Graphic Interface layer find appropriate parameters of certain BIFS nodes in the graph and are encoded properly into the binary format. Finally, the BIFS information is generated in the binary format, multiplexed with the imported media data and exported into an MPEG-4 formatted file.

The resulting MPEG-4 file has several advantages against the same application produced with MPEG-2 (DVD application). Firstly, it has better quality of audio and video at a lower bit-rate (smaller size of file). It is due to a better compression of media data offered by MPEG-4. Secondly, with the implementation of BIFS node, subtitle information can be carried out in a standard but flexible way: the position of text, font style of text can be modified easily. Another important feature, which is not supported in MPEG-2, is that several Video clips (possibly Audio clips) can be defined in a scene for parallel playback of several video streams (audio streams). It is the case of video broadcasting with sign language aid for deaf viewers (Figure 1-d).
4.2 Simple interactive video application

Applications in this group and in the following group demonstrate the interactive features, which are currently not available with any video encoding standard, but MPEG-4. A simple interactive video application is firstly illustrated. We created a multimedia content, in which user can browse along a series of images, one by one with the aid of forward / backward “button” (Figure 6-a). The term “button” in a video clip is a new concept and is supported exclusively in video compression with MPEG-4 BIFS. It means an active surface in the scene, which can be animated to simulate the “clicking”, “releasing” action with the pointing device (cursor or keyboard) supported by player. Then some related functions will be activated. To facilitate the new behaviors of video application, MPEG-4 BIFS supports some sensor-like nodes. Their states will be changed upon triggers from viewers (in the current application, the trigger is clicking action on forward / backward button) or from other BIFS node. By the mechanism of routing – linking a parameter of one BIFS node to another – trigger can be transmitted through all other nodes of the scene, generating a global reaction. The essential requirement of such applications is that every node in the hierarchy tree can be connected to one another; some disguised BIFS nodes – having no spatial / temporal location, serving as a trigger transmitter only – can be created and manipulated. As a result, the programmatic mode of ICCT proves to be a more powerful tool against parametric one.

For the sake of simplicity, we only explain here the main steps in the process of creating this type of application:

- All images must be inserted into ICCT with the Import Media Data functional unit. MPEG-4 player supports JPEG format, therefore the imported images must be converted accordingly.
- By using the syntax of script in [3], each media data (compression information for image) is associated with an ImageTexture node. It serves as a Texture parameter of an Appearance node. The material parameter of every Appearance is set to Material2D node, whose transparency parameter is equal to 1 as its default value (the image then is invisible).
- Two TouchSensor nodes are attached to two Shape nodes, which are based on Circle node. They create the appearance of the forward / backward button.
- For appropriate reaction driven by triggers from TouchSensors, one Script node is used to evaluate triggers from TouchSensors (which button was clicked) and generate proper action (the transparency parameter of the current ImageTexture is set to 0, the one of the next or the previous ImageTexture shall take value 1 regarding to whether trigger is from forward or backward button). Some other conditions are also checked by Script node, for example if the current ImageTexture is the first one in the series and the backward button is clicked, setting the previous image visible is skipped.

The functionality of the forward / backward button can be extended to any possible interaction of the user with the scene, which is only limited by the imagination of the application designer, such as selecting a language for current video clip, toggling on / off subtitle, starting a certain short explaining video for a current view,…These simple applications of this group open a revolutionary aspect for video application in general: video applications become more intelligent, they support interaction with users in their structure, i.e. conditional behaviors of video clip are encoded within the bit-stream in the same manner as a download program in computer domain. It shortens the gap between computing and video broadcasting technology, which is a tendency in the multimedia era nowadays.
4.3 Full interactive application

If applications in the previous group are in the half way between video broadcasting and computing technology, applications in this group is the complete merging of video application and computer technique. The video application in Figure 6 demonstrates this convergence. In fact, it is a Memory game, which is already familiar to computer user. The target of the game is to find all the matching pairs of image-cards, which are shown with their faces down. By two consecutive selects you can turn them over. If the exposed images are the same, they will be removed. Otherwise the image-cards will be covered again and user shall remember their position for other try. Obviously, this simple game can be programmed with any computer language, such as Java, C++. We demonstrate that it can also be done with MPEG-4 BIFS, a feature of video compression technology. The difference here is that the game is not an executable program but an MPEG-4 BIFS enabling file – the latest file-type of movie container. To exploit the movie nature of the game, the back sides of image-cards are parts of a movie clip for some advertisements; When all pairs of images are found and removed, the game becomes a conventional movie file: the main movie clip will be played. The structure of this application created by ICCT in programmatic mode can be shortly described as follows:

- Video and audio clips are inserted into ICCT with Import Media Data functional unit. They are 2 pairs of video and audio streams for the advertising and main movie respectively. Beforehand, these media streams are encoded accordingly to the MPEG-4 standard ([6], [7]). There are 16 JPEG images, which are also imported into ICCT.
- Two Shape nodes and one TouchSensor node are grouped together within a Transform2D node to create every clickable card having two sides. The first Shape node consisting of a Rectangle node and an ImageTexture node represents the front side of the card. The ImageTexture node maps each card to an associated JPEG data. For displaying the advertising movie clip in piecewise manner, we build the second Shape node of every card with a MovieTexture node and an IndexedFaceSet2D node. The former is used for addressing the video data as a whole, while the latter defines the small mapped area of the video, which is represented by the parent Shape node. There are also another Shape node and a Sound2D node in the scene, which correspond to the video and audio of the main movie. By setting the transparency attribute of the Material2D node in each Shape node as in subsection 4.2, we can show either front or back side of any card to user.
- There is one central Script node, which is the target for routing from all TouchSensor node. The Script node exposes the logical operations of the game: upon each trigger from a certain TouchSensor node, it decides which side of the associated card is visible to user. It remembers the state of the game such as which image was selected by used as the first choice, how many pairs of image-card were found, …. The behaviors of the Script node is predefined in its url attribute. The value of this attribute actually is a script-program, which has the syntax specified in [4].

Figure 6: Interactive multimedia contents based on MPEG-4 BIFS.

The presented Memory game is just an example to demonstrate a new feature of the MPEG-4 standard: the computing power, which is implemented by means of routing and programming Script nodes. Originally, it provides a tool for flexibly composing and manipulating a scene for movie application. But the mentioned scene can be such a rich multimedia content with complex interactivity that we can hardly separate from a computer program. The separate role between passive media data files, basically including all audio video material, and
active programs, used as a tool for users manipulate data, is already merged into one entity with the MPEG-4 standard.

5 Conclusion and perspectives

MPEG-4 is the first standard, which enables the scene-related information transmitted together with video / audio data. Its specifications shorten the gap between conventional movies and enriched multimedia content. However, the advanced features seem to be left aside in recent MPEG-4 implementations and applications, mainly focused on video / audio compression issues. It is mainly due to the complexity of encoding BIFS and the lack of efficient authoring tools.

The work presented here tends to fulfill this gap appearing in the recent MPEG-4 implementations. We first propose a set of mapping rules between the multimedia objects and the specific MPEG-4 representations. Then, we implemented the ICCT which allows, by accessing a user-friendly interface, to build multimedia applications without any knowledge related to the internal representation structure of an MPEG-4 scene. In this mode of ICCT, multi-media content is object-oriented. Spatial / temporal attributes and several other properties can be attached to each one and can be easily manipulated with adding / setting operations. These typical instructions no longer exist in the programmatic mode of ICCT, where a scene is defined with script. The advantage of the second mode is to provide author with the powerful programmability to describe sophisticated multimedia scenarios with complex interactive links. The current version of ICCT does not support the visualization of spatial / temporal of objects in the programmatic mode.

In the next version of the authoring tool, the programmatic mode will be enhanced with visualization for easier object-manipulation. The semantic of its language will be more human understandable. It will get closer to the level of the forth generation of programming language: consisting of more setting, dragging and dropping operators applied to high abstraction level objects and less programming with some syntax language. MPEG-4 player supporting 3D profile also have to be implemented. It is another advanced feature of MPEG-4, describing a mixed scene with 2D and 3D objects. The more features are supported by player, the more load (CPU time, memory resource) is put onto the client side. Therefore, optimization of the utilization of BIFS nodes will be an issue of our future developments.

References