Design and Implementation of Interactive Contents Authoring Tool for MPEG-4

Hsu-Yang Kung, Che-I Wu, and Jiun-Ju Wei

Department of Management Information Systems
National Pingtung University of Science and Technology, Pingtung, Taiwan
{kung, joey, m9156021}@mail.npust.edu.tw

Abstract. This paper proposed and implemented a QoS-Aware MPEG-4 authoring tool with provision of content interaction control, which is named the Q4AT system. Q4AT provides friendly user interface in a WYSWYG manner for users to create their favorite MPEG-4 scenes. The characteristics of Q4AT are as follows. (i) The description of a scene is based on the standard BIFS language. (ii) The creation of the object description is based on the syntactic description language (SDL). (iii) All descriptions are VRML-based. A VRML-based system is compatible to different browsers and operation systems. Therefore, the author could simplify the steps of creating an MPEG-4 scene. (iv) Q4AT supports the functions of the quality of service by the proposed object priority differentiation control. Using Q4AT, authors are transparent to the MPEG-4 processing layers and are capable of concentrating on the editing function usage of the Q4AT. Q4AT makes it possible to easily create a complicated MPEG-4 scene and reduce the difficulty of MPEG-4 media creation and presentation.

1. Introduction

MPEG-4 is a compression standard developed by ISO/IEC MPEG (Moving Picture Experts Group) and is totally different from MPEG-1/2. MPEG-4 standard improves the efficiency of media compression and tries to provide the applications integrating multimedia, communications, and computers [1]. The fundamental infrastructure of MPEG-4 is object-oriented and media operations (presentations) described in an MPEG-4 scene [2], [3]. An MPEG-4 scene combines the object descriptions of natural and synthetic media objects, e.g., the text, image, 3D, audio, and video objects, which can be manipulated by interaction operations [4], [5], [6]. To achieve a user-oriented MPEG-4 scene presentation, a user friendly MPEG-4 authoring tool is urgently required [7], [8]. Using a well-design authoring tool, a film maker could dynamically change the structure of an MPEG-4 scene to achieve the desired presentation contents [9], [10].

In the paper, we proposed and implemented a QoS-Aware MPEG-4 authoring tool with provision of content interaction control (Q4AT). Q4AT is developed using VC++, OpenGL, and VRML languages. *Q4AT* uses the standard BIFS language and the syntactic description language (SDL) to create the descriptions of a scene and the media objects, respectively. All descriptions are VRML-based to be compatible to different browsers and operation systems. To support multimedia communications, *Q4AT* provides the QoS specifications via the proposed object priority differentiation control. *Q4AT* provides friendly user interface in a WYSWYG manner for users to create their favorite MPEG-4 scenes.

The rest of this paper is organized as follows. Section 2 describes the system architecture of Q4AT. Section 3 illustrates the implementation results and usage of Q4AT. Section 4 concludes this paper.

2. The System Architecture of Q4AT

Figure 1 depicts the system architecture of *Q4AT*. The processes of creating a standard MPEG-4 file (.mp4) are composed of authoring actions, e.g., the Edit actions, the Preview actions, and the Save actions. For example, an author news (opens) an mp4 file to edit, previews the authoring results using the VRML-enabled window, and saves the final results in the mp4 format [11]. The main components of the *Q4AT* system are (i) the Editing Board component and (ii) the Q4AT Function component, which are described as follows.

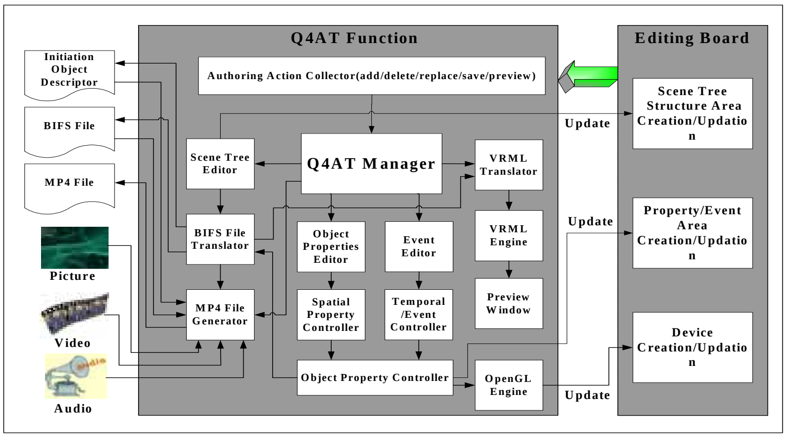

Q4AT Function
Editing Board
Initiation Object Descriptor
BIFS File
MP4 File
Picture
Video
Audio
Authoring Action Collector(add/delete/replace/save/preview)
Scene Tree Editor
Q4AT Manager
VRML Translator
BIFS File Translator
Object Properties Editor
Event Editor
VRML Engine
MP4 File Generator
Spatial Property Controller
Temporal /Event Controller
Preview Window
Object Property Controller
OpenGL Engine
Update
Update
Update
Scene Tree Structure Area Creation/Updatio n
Property/Event Area Creation/Updatio n
Device Creation/Updatio n

Fig. 1. The system architecture of the Q4AT system.

2.1 Editing Board Component

Figure 2 illustrates the user interface, which is Editing Board component of *Q4AT*. There are five control modules composed of the editing component, which are (i) Pre-define Button, (ii) OpenGL Device, (iii) Property Area, (iv) Scene Tree Structure Area, and (v) Preview Window. We describe the functionality of control modules as follows.

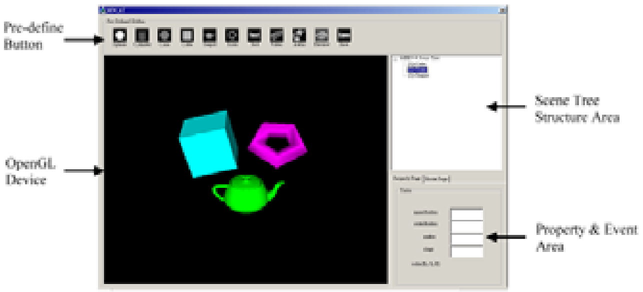


Fig. 2. The user interface of Q4AT

2.1.1 Pre-define Button

OpenGL Device is responsible for rendering an MPEG-4 scene in the WYSWYG way [12]. OpenGL Device translates and displays all editing actions of a scene author. Figure 3 illustrates the display result after editing media object using OpenGL Device. An author can change the coordination and view points of objects. When the properties of objects are changed, OpenGL Device re-displays the modified object at the same time.

(i). Object Button. Object Button provides the functionality for an MPEG-4 author to create various media objects via the defined icons. The icons of Object Button include Sphere, Cylinder, Cone, Cube, Teapot, Torus, Text, Video, and Audio. An author uses the icons of Object Button creates (editing) an complicated MPEG-4 scene, which is composed of the 2D、3D、picture、text、audio, and video objects. Noted that there are corresponding video and audio files, which is specifies by the author, for the creation of video and audio objects. The scene author adds these objects to OpenGL Device and specifies the properties of media objects.

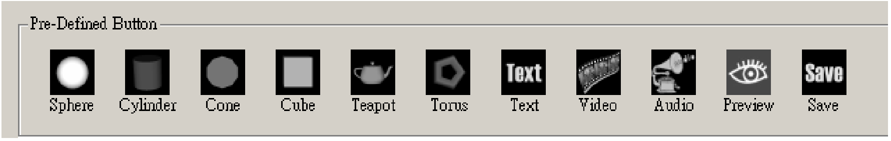


Fig. 3. Pre-define buttons of Q4AT.

(ii). Preview Button. An author uses Preview Button to view the designed scene in advance of the next editing operation. The control policy of Pre-view Button is to trigger the VRML preview window and pop-up from Q4AT.

(iii). Save Button. An author uses Save Button to save the editing results of the designed MPEG-4 scene. When Save Button is clicked, MPEG-4 File Generator activates to collect attribute information of objects and MPEG-4 File Generator packs the related object files into a standard .mp4 file.

2.1.2 OpenGL Device

OpenGL Device translates and displays all editing actions of a scene author. Figure 4 illustrates the display result after editing media object using OpenGL Device. An author can change the coordination and view points of objects. When the properties of objects are changed, OpenGL Device re-displays the modified object at the same time.

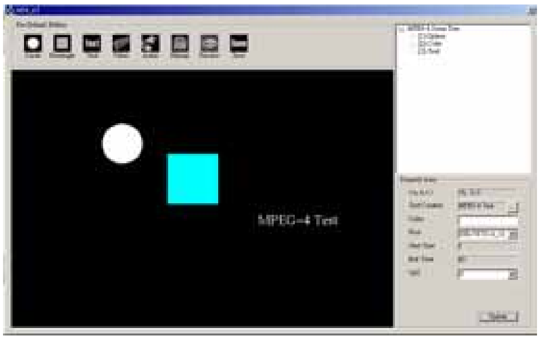
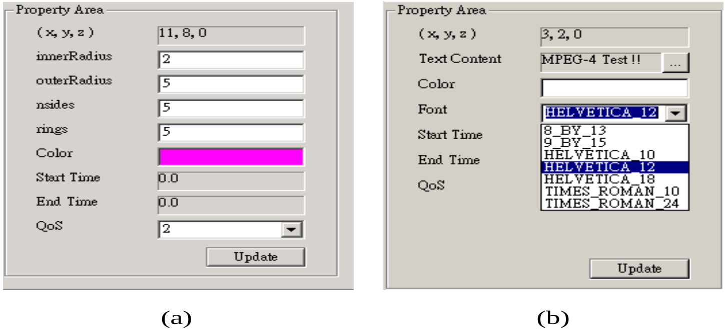


Fig. 4. A display example using OpenGL Device.

2.1.3 Property Area

Property Area provides different objects of an MPEG-4 scene with the assignment of object attributes. Different objects may have same properties, e.g., color assignment, spatial/temporal values, and QoS priority, and also have heterogeneous properties, e.g., text font, encoding types of audio/video files, and sphere radius. The Property Area displays the designed object properties and gives the author the limitation of authority for editing different kind of media objects. OpenGL Device directly renders the modification of objects. Figure 5-(a) and -(b) illustrates a dialog example of using Property Area for specifying the attributes of a sphere object and a text object, respectively. In Property Area, some fields for specifying an object are changeable and some files are not. An author can modify the attribute files of a sphere object, including innerRadius, outerRadius, insides, rings, color and QoS.



Property Area

(x, y, z)	11, 8, 0
innerRadius	2
outerRadius	5
nsides	5
rings	5
Color	
Start Time	0.0
End Time	0.0
QoS	2

(a)

Property Area

(x, y, z)	3, 2, 0
Text Content	MPEG-4 Test !!
Color	
Font	HELVETICA_12
Start Time	
End Time	
QoS	

(b)

Fig. 5. The dialog windows of Property Area for (a) sphere object and (b) text object.

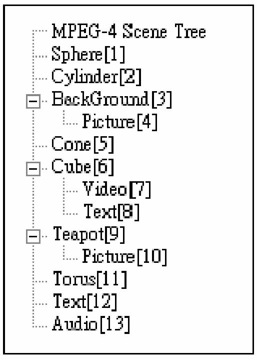
2.1.4 Preview Window

Scene Tree Structure Area displays the tree structure of an MPEG-4 scene, which illustrates the detailed information about media objects, including the object id, the object name, the object type, and the temporal/spatial relationship between other media objects. Figure 6 depicts the functions of Scene Tree Structure Area and is described as follow.

(i). As an author uses Pre-define Button to add a medium object on OpenGL Device, Scene Tree Structure Area creates the corresponding tree node. A node of the scene tree contains the object information, e.g., object ID and object type.

(ii). When an author clicks a node of Scene Tree Structure Are, Property Area displays its properties to set attribute values of the node, i.e., the medium object.

(iii). An author can drag a node of scene tree to change the relationship between other nodes. To ensure the attribute consistence between tree nodes, Q4AT automatically monitors the legality of scene objects to prevent some violation situations.



```
MPEG-4 Scene Tree
Sphere[1]
Cylinder[2]
BackGround[3]
    Picture[4]
Cone[5]
Cube[6]
    Video[7]
    Text[8]
Teapot[9]
    Picture[10]
Torus[11]
Text[12]
Audio[13]
```

Fig. 6. An illustration example of Scene Tree Structure Area.

2.1.5 Preview Window

Preview Window is a pop-up window for scene authors to preview the results of scene realization. Preview Window was manipulated via VRML. Using VRML, the problem that OpenGL Device can’t integrate with the video objects is solved. Furthermore, VRML supports the capability of content interaction. A scene author is capable of dynamically and conveniently adjusting the status of a scene in a WYSWYG manner.

2.2 Q4AT Function Component

The Q4AT Function component is responsible for executing and realizing the authoring actions. All authoring actions can be considered as the transmission of action messages between Q4AT Function component and the author i.e., Editing Board component. The Q4AT Function is mainly composed of the Authoring Action

Collector, Object Property Controller, BIFS File Translator, VRML Translator, and MP4 File Generator, which provide an author with a transparent environment to easily create a desired MPEG-4 presentation. We describe the main control modules as follows.

(i). Authoring Action Collector gathers the editing actions of a scene author and sends editing actions to Q4AT Manager. Q4AT Manager distributes actions messages to the related Editors to manipulate media object.

(ii). Object Property Controller gets action messages, including the adding, deleting, properties modification, refreshing actions, from the Q4AT Manager. Then, Object Property Controller transmits the gathering messages to OpenGL Device and BIFS File Translator for the displaying and translation of a scene.

(iii). BIFS File Translator detects changes of object properties and keeps up the latest properties prepared for VRML Translator and MP4 File Generator. The main purpose of BIFS File Translator is to create the scene description and object description of MPEG-4 standard, and then generates the BIFS file.

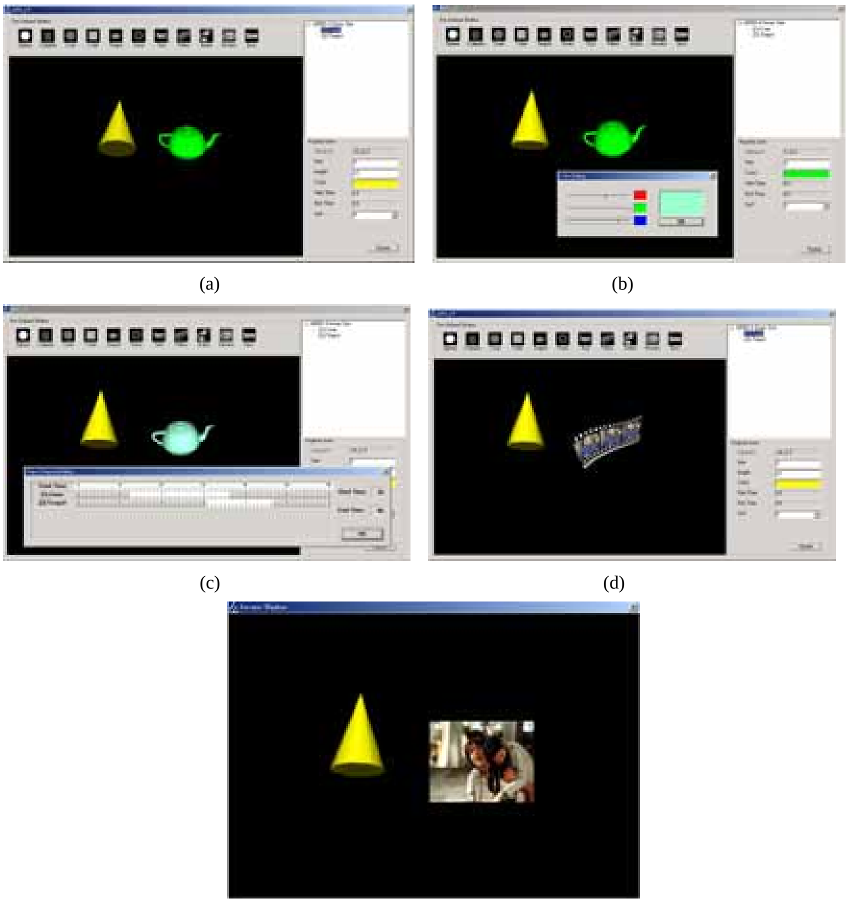
(iv). VRML Translator gets the BIFS file from BIFS File Translator and translates the BIFS file into the formats of VRML syntax used for Preview Window.

(v). The main function of MP4 File Generator is to build the standard MPEG-4 file (.mp4). MP4 File Generator gathers all media files of the corresponding media objects and packs the media files with scene/object description into a standard MPEG-4 file.

3. Usage of Q4AT

The steps of realizing an MPEG-4 scene are to add media objects into a scene, to modify the properties of objects, to preview the authoring results, and to finally save the editing results. Figure 7 illustrates an example of editing an MPEG-4 scene using Q4AT. Figure 7-(a) shows the authoring results of adding two objects into OpenGL Device. A scene author uses mouse to easily drag media objects in a scene and to change their spatial attributes or viewpoint. An author uses Scene Tree Structure Area to add a corresponding node at the same time. There are two ways to modify the properties of a medium object. (i) An author can use the mouse to click the object in OpenGL Device. Or, (ii) an author clicks the node of Scene Tree Structure Area to modify the properties of the node. These two actions will trigger the Object Property Area to display corresponding properties of nodes, and then scene author modifies the changeable attributes in Object Property Area and change the attribute values directly. Note that not all properties in Object Property Area are modifiable, for example, the spatial value of a medium object is unchangeable. Figure 7-(b) depicts the color dialog for modifying the colors of an object. Figure 7-(c) shows the timeline dialog for editing the start/end time of a medium object. OpenGL Device displays all modifications in real-time for the scene author.

Due to the critical limitation of integrating video objects by using OpenGL, Q4AT uses a 2-D rectangle to express a video object and provides VRML Preview Window to solve the problem of OpenGL. Furthermore, the designed interaction functions are useful and convenient for scene authors to adjust the deployment of scene objects by using VRML. Figure 79-(d) shows the display example of a video object in OpenGL Device. Figure 7-(e) illustrates the preview result of using VRML Preview Window to preview the deployment of scene objects.



(a)

(b)

(c)

(d)

(e)

Fig. 7. (a) Adding two media objects. (b) A Color dialog. (c) A TimeLine dialog. (d) Adding video object into a scene. (e) Using VRML Window to preview.

After authoring the MPEG-4 scene, the BIFS File Translator of Q4AT translates the designed scene tree into the scene.bt file, which contains the standard BIFS commands with text format. Q4AT adopts the MP4Box encoder [13] to interpret the BIFS commands in the scene.bt file and then encode the BIFS commands into the scene.mp4 file. The scene.mp4 file is just the scene and object description file with binary format. The related media content files are pointed in the scene.mp4. Figure 8 depicts the processing procedure of generating a recognized .mp4 file and the processing steps are as follows.

(i). The acceptable medium format for the video objects is the .media format. Firstly, Q4AT adopts the DivxEnc converter [14] to translate the original medium file with .avi format into the intermediate file with .m4v format. Secondly, Q4AT adopts the MP4Creater60 converter [15] to translate the intermediate medium file from the .m4v format into the .mp4 format. However, the .mp4 format file created by MP4Creator60 is only the MPEG-4 simple profile format, which can't be used for the MPED-4 core profile environment, i.e., the multiple object authoring and presentation environment. Therefore, Q4AT adopts the MP4Box encoder to translate the .mp4 file into there files with the .nhnt, .info, and .media formats. The generated medium file with .media format can be realized by MP4Box for video and audio objects.

(ii). The acceptable medium format for the audio objects is the .media or .mp3 format.

(iii). The acceptable medium format for the image objects is the .jpg or .bmp format.

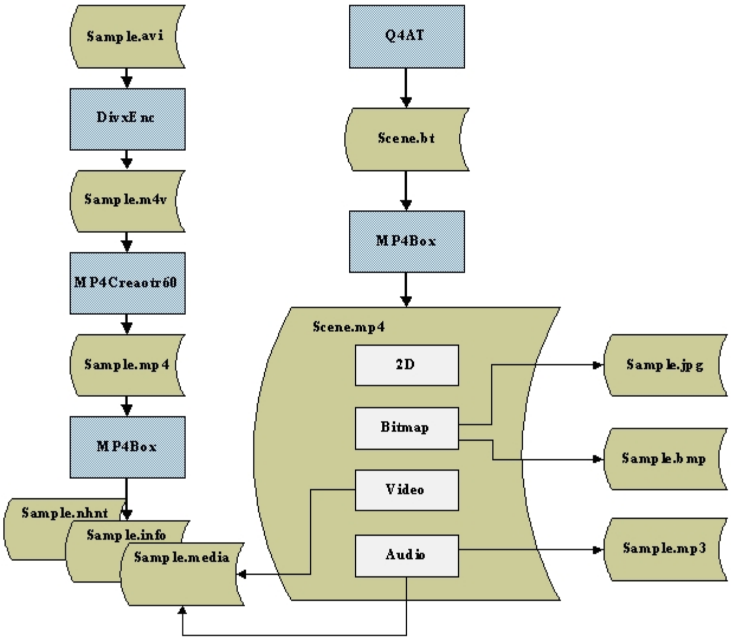
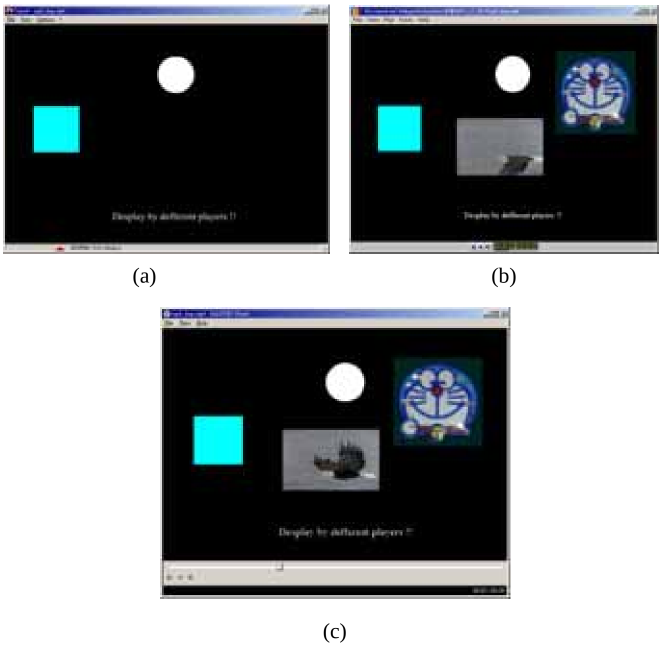


Fig. 8. The processing procedure of generating a .mp4 file.

To prove the feasibility of Q4AT, this paper uses the well-known MPEG-4 players, including the Osmo4 of ENST [16], the M4Play of IBM [17], and the MAXPEG of DigiMax [18], to present the generated .mp4 file. Figure 9-(a), -(b), and -(c) depicts an MPEG-4 file presentation results, which is authored by Q4AT, by using the Osmo4 player, the M4Play player, and the MAXPEG player, respectively. Figure 10 depicts the designed MPEG-4 scene presented on a PDA. The MPEG-4 player is the Osmo4, which is WinCE-based and is the only available MPEG-4 player for handheld devices.


Display by different players !!
Display by different players !!
Display by different players !!

(a) (b)

(c)

Fig. 9. The MPEG-4 presentation illustrations using (a) Osmo4 player, (b) M4Play player, and (c) the MAXPEG player.

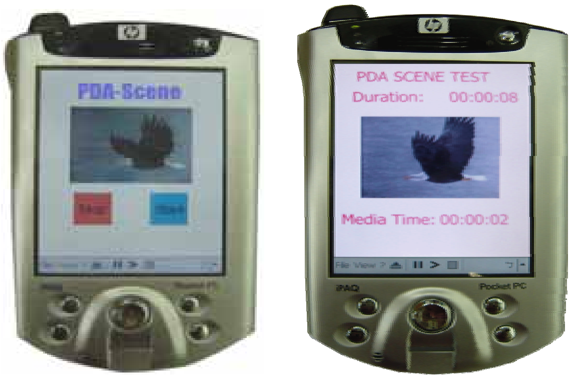


Fig. 10. The MPEG-4 presentation illustration on a PDA

4. Conclusion

In the paper, we proposed and implemented a QoS-Aware MPEG-4 authoring tool with provision of content interaction control (*Q4AT*). The *Q4AT* system provides the friendly user interface and helps the author easily create an MPEG-4 scene composed of heterogeneous media objects. The capabilities of the QoS-aware with the object priority control and the content interaction with the interaction event control are provided in the *Q4AT* system. To achieve a smooth MPEG-4 presentation, the *Q4AT* system designs Object Property Area to specify the temporal/spatial relations of the media objects. The main goal of *Q4AT* is to provide powerful functionality for authors, who don't need have the complicated MPEG-4 knowledge, are capable of creating a standard MPEG-4 file.

Acknowledgement

The authors would like to thank the National Science Council of the Republic of China, Taiwan for financially supporting this research under Contract No. NSC-93-2219-E-020-002.

Reference

1. ISO/IEC 14496-1: 1999 Information technology -- Coding of audio-visual objects -- Part 1: Systems (1999)
2. Eleftheriadis, A.: MPEG-4 systems: architecting object-based audio-visual content. IEEE Second Workshop on Multimedia Signal Processing (1998) 535-540
3. Herpel, C.: Elementary stream management in MPEG-4, Vol. 9, Issue 2. IEEE Transactions on Circuits and Systems for Video Technology (1999) 315-324
4. Viljoen, D. W., Calit,z A. P., Cowley, N. L. O.: A 2-D MPEG-4 multimedia authoring tool. Proceedings of the 2nd international conference on Computer graphics, virtual Reality, visualization and interaction (2003) 151-160
5. Kim, K. Y., Hong, J. W., Park, C. M., and Jung, H. K.: The architecture of MPEG-4 based IPMP authoring system. Proceedings of IEEE International Conference on Multimedia and Expo (ICME '04) Vol. 1. (2004) 447 - 450
6. Liu, D. R., Shieh, M. J., Lee, Y. C., and Chen, W. C.: On the Design and Implementation of an MPEG-4 Scene Editor. Digest of Technical Papers of International Conference on Consumer Electronics (2000) 120-121
7. Bulterman, D. C. A. and Hardman, L.: Structured multimedia authoring, Vol. 1. Issue 1. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP) (2005) 89-109
8. Pourmohammadi-Fallah, Y., Asrar-Haghighi, K., and Alnuweiri, H. M.: Streaming multimedia over the Internet, Vol. 23. Issue 1. IEEE Potentials (2004) 34-37
9. Daras, P., Kompatsiaris, I., Raptis, T., and Strintzis, M. G.: An MPEG-4 tool for composing 3D scenes, Vol. 11. Issue 2. IEEE Multimedia (2004) 58-97
10. Drumm, H. and Kuhhirt, U.: The shadow tool - new visual cues for MPEG-4 authors. Proceeding of 2004 IEEE International Symposium on Consumer Electronics (2004) 77 - 82
11. Kim, K., Lee, I., and Ki, M.: Interactive contents authoring system based on XMT and BIFS. Proceedings of the tenth ACM international conference on Multimedia (2002) 275-278
12. Zhigen, X., Yan, Y., and Chen, J. X.: OpenGL programming in Java, Vol. 7. Issue 1. Computing in Science and Engineering (2005) 51 - 55
13. MP4box, http://www.aziendeassociate.it/cd/gpac/dev/MP4Box.7z
14. DivxEnc, http://www.greatis.com/security/download.htm
15. MP4Creator60, http://www.esato.com/archive/t.php/t-63404
16. Osmo4, http://www.comelec.enst.fr/osmo4/
17. M4Play, http://www.alphaworks.ibm.com/tech/tk4mpeg4
18. MAXPEG, http://www.digimax.com.tw/MPEG4/