Merging Web Brower and Interactive Video – A HyperVideo System for e-Learning and e-Entertainment

Yi-Chun Liao, Han-Bin Chang, Hui-Huang Hsu, Timothy K. Shih
Multimedia Information Network Lab
Department of Computer Science and Information Engineering
Tamkang University
Taiwan, ROC
g8190221@tkgis.tku.edu.tw, ham@mail.mine.tku.edu.tw, {hhsu, tshih}@cs.tku.edu.tw

Abstract

Hypervideo is an instance of hypermedia. A video with hyperlinks in designated areas of certain frames is just like hypertext in the WWW environment. A video can be viewed as a main document the producer wants to present and it can include many hyperlinks. Thus, it can be presented with more auxiliary materials. For effective browsing and navigation, semantic annotation is also essential for the visual content of movies. It is very important to show any particular object that the movie producer wants to highlight in educational videos or commercials. Synchronization problem needs to be solved when the different objects are rendered in a multimedia player. We propose an interactive and structure-based video authoring system, in which we can describe the video object and make multistory movies. The system provides a manual object-based interface for the film producer to select any meaningful object and annotate it. We also provide a video story construction function to interact with the video viewer. A hypervideo player is also developed to play the video file made by the authoring system.

Keywords: Hypervideo, e-learning, video interaction.

1 Introduction

Recently, technology has reached a level where vast amounts of digital information are available at a low price. At the same time, powerful computation and “meaningful” multimedia content make communication between human and machine possible. Advance in the video processing technologies have simplified the editing process to automatically create highly semantic video data. For example, one of the most active researches is the video-content analysis and retrieval. The digital form allows processing of the video data to generate appropriate data abstractions that permit flexible video-database organization and enable content-based retrieval of video. In many existing researches, it is desirable to identify syntactic and semantic components in the video material. One example of video context is the presence of an object in the video. Ideally, a video can be automatically annotated as a result of machine interpretation of the semantic content of the video. Although visual content is a major source of information in a video program, an effective strategy in video-content analysis is to use attributes extractable from multimedia sources. However, how I get the price or specifications of the product while watching a commercial? How to extract the information based on video content?

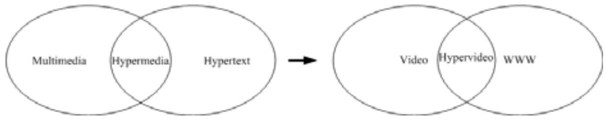


Figure 1 Hypermedia and hypervideo

Traditional video presentation uses a video play with forward, backward, and scrolling bar in a linear way. Such a presentation mode results from context-restricted content and a traditional video structure. Although the provided video technology has high compression rate as MPEG-1, MPEG-2 [1] and H.26x [2] and the hardware has more different concepts as VCD2.0 and DVD [3], the presentation of video data is processed as a stand-alone data type that restricts the function of video content.

When we talk about hypertext, multimedia, and hypermedia, there is a relation between them as shown in Figure 1[4]. The nonlinear information link is a major property of hypermedia and hypertext and gives the media viewer an opportunity to decide his/her reading path. Using the main concepts of hypermedia and hypertext for multimedia documents, we can use video as a multimedia document like WWW and it is referred as hypervideo [5]. In the hypervideo, the “Link” or “Jump” functions facilitate the video clip as a hyperlink-based map.

The MPEG-4 scene description language BIFS (Binary Format for Scenes) is used to make the goal of the interactive audiovisual scenes. The standard defines a set of the binary tools that provide the coded representation of individual audio-visual objects, text/graphics and synthetic objects. A set of the authoring tools can be found for editing the textual format (XMT) that will be compiled into MPEG-4 to be played in a MPEG-4 player. XMT-A and XMT-O are two textual formats that represent different level scene description languages.

XMT-A is developed to describe the low features in MPEG-4 scene description language, and the complexity is like the VRML as we would like to use it to build an interactive scene. XMT-O is a high level language based on W3C SMIL with dynamic, interactive properties for multimedia presentation. It is preferred to author the interactive video environment with a simple and easy format. Not all functions of MPEG-4 are supported in XMT-O or XMT-A, since replicating the tools of MPEG-4 would set an extra complicated context.

In [9], the XMT-O-based authoring tool is proposed to edit the spatial and temporal relations that deeply reduce the complexity of authoring. A useful and powerful editing tool is needed for an amateur in such visual manipulation work in which “drag and drop” is more suitable than a text-based or XML-based editor. Visual-based and text-based methods are used in this tool. Moreover, xMedia objects in MPEG-4 Systems specification are used. Combining the properties of SMIL and xMedia specification, the behaviors of the object is defined according to the timeline and the triggering of events. That is, in the tool, the objects construct such unnatural scene from a different viewpoint.

The concept of hypervideo is extended into Detail-on-Demand hypervideo by including the function of the video summarization [10, 11, 12, 13]. Hyper-Hitchcock is a hypervideo editor, in which the user semi-automatically get video clips and the editor can automatically generate a multi-level summary video. Finally, the user has a tree-like graph that uses the key frame in a higher level of the graph. If you are interesting in a key frame clip, you can get a complete video clip by the hyperlink between them.

Some video-based applications are used in the language learning environment and focus on the interactivity between the teacher and the student. Yoshiaki Hada [15] used XML to

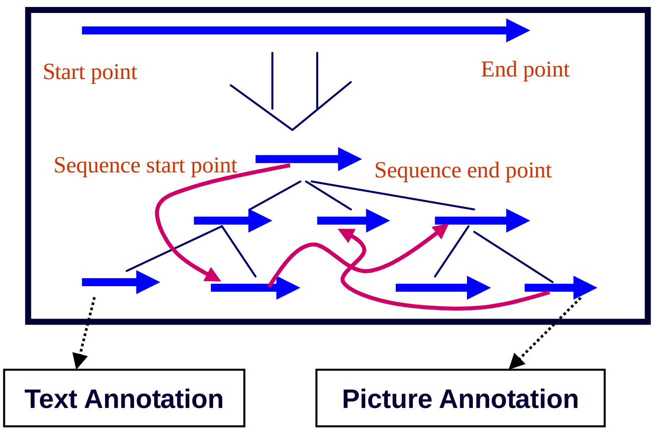


Figure 2 The browsing sequence of the hypervideo

extend a traditional videoconference system so that the system can record learning scene as a video file for teachers. The system can add whatever the teacher wants to revise and comments into conversational video.

Although, some outstanding systems have had good performance on object detection and video segmentation [6, 7, 8, 14]. However, given a video file, another aspect of video processing is that a user may not be interested in the entire clip, but rather portions of the clip. And the audience may interests in something in certain frames. We propose a novel semantic video authoring system which focuses on assisting the video author in annotating based on video semantic level. The concept of hypervideo will be applied in our system to link the different multimedia object. Figure 2 shows that the browsing sequence of our system will not be like the traditional video that only has a liner browsing sequence. And any kind of media object will be used to exhibit the required information of the video that has an anchor to hyperlink. The hyperlink will increase the complexity of authoring and browsing video as so many anchors will be set in a video clip and those anchors will move while the video clip is played. We will address the issues on (1) where, (2) when, and (3) at which object is suitable to be hyperlinked in a video clip.

Hypervideo is a new video processing technique similar to hypertext. In a Web page, useful information can be obtained right on a click without time and space restriction through hyperlinks. The available video producing tools cannot perform such hyperlinks on video. Not all functions of hypervideo are supported in current hypervideo systems or detail-on-demand systems. One of our crucial issues is to let the editor select an object to set an anchor. This approach differs from the current system and it is a necessary in a semantic viewpoint.

The rest of the paper is organized as following. Section 2 first presents in detail our approach to enhance functionality of the hypervideo and show the interaction model by using Petri Net. In section 3 introduces some key technologies of the implemented system. Section 4 concludes the paper.

2 Object-based Annotation and Multistory Hypervideo

Searching a requested clip in video takes much time. Generally, a video file for a movie is about two hours. The existing video annotation approaches are sequential and time-consuming. For example, there are 1500~2000 shots in a movie. How can the film producers annotate them one by one? The trivial approach cannot provide more powerful annotation, in which we often can use a user-defined shot as a description unit. For this, there are two major components in our system: video object annotation and video structure constructing. Video object annotation is an advance tool to present an object that has extra information. In video structure constructing, the user can make a branching story movie easily.

Proposed Method

In the MPEG specification, we are promised to define a data stream called the user-data stream [16]. Based on codec rules, if given a video file, we put the additional information as a data stream. Those additional data can be anything such as a text, a video clip, a URL link, or a still image. Even some commands will be used to control video playing if the decoder can read those commands (shown in Figure 3).

In Figure 3, the extension data is composed of many data units that are attached to the original video file. These user_data units are related to the objects or clips in the original video.

2.1 HyperVideo Presentation: Definitions

Building a hypermedia system has major problems need to be overcome such as: the synchronization of the continuous and discrete media, the structure of media presentation, and the interface layout.

In our past research, we proposed a video presentation system with inter-object synchronization based on the Petri-net mechanism [18, 19]. Figure 4 shows a general interface of the multimedia presentation system includes three major components: a video, an image (presentation slide), and a slide index, such a presentation system is used in distance learning popularly [19, 20, 21]. The system was proposed using SMIL [22] with RealPlayer [23] or Microsoft ASF with Microsoft MediaPlayer [24]. The method is to add hyperlink markers into a video sequence that will bring out a slide while the video with marker is played. And the slide index can be used to jump to the different video clip. Synchronization problem needs to be dealt between the video and slides. Such a system is a subset of the hypervideo system, but it only suits for the web environment.

In our proposed hypervideo system, it is needed to manage the different media objects and control those objects according to the time schedule. For some unsolved limitations, our system can efficiently present the semantic entries to viewers, and we leave the underlying semantic structure of the video to the film director based on which the film director wish to depict to the audience. The system has the features as follows:

- Common File Format:
  In our system, the format of the annotated video is not restricted, and the exported file can be play in a common video player. Because of the proposed system does not modify the original video data (shown in Figure 3).
- Interactive Story:
  Like a DVD movie, you can have a seamless presentation [17]. In a DVD format disk, the audience has some selection buttons in a still image when a DVD movie is played. In our system, the audience interacts with the movie story, if the film director allowed and made a branch point. For the flexibility, we do not use a still image to achieve interaction, because the presentation will be paused.
- Video object annotation:
  The object can be annotated when the file producer wants to highlight the object. For a video producer, it is possible that the defined object has a multimedia description. For example, in a commercial, the producer can insert a video clip for a product for sale. And the viewer can view the additional information if the viewer is interested in the product.
- Video Structure Constructing:
  For example, an instructor can provide the same teaching video materials with the different learning paths for learners. We use the tree structure to present the video story.
- Video Browsing: Extracting structures for video is a fundamental task to facilitate user’s browsing. According to the last feature, the producer has

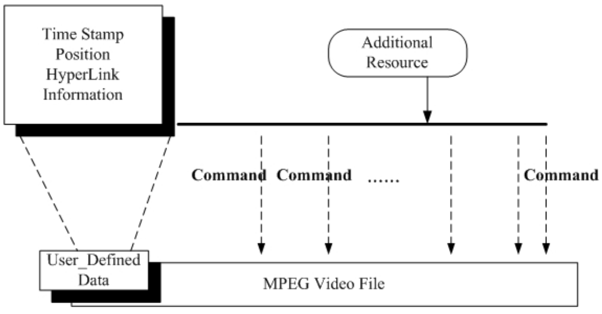


Figure 3 The user_data is attached to the Raw video data

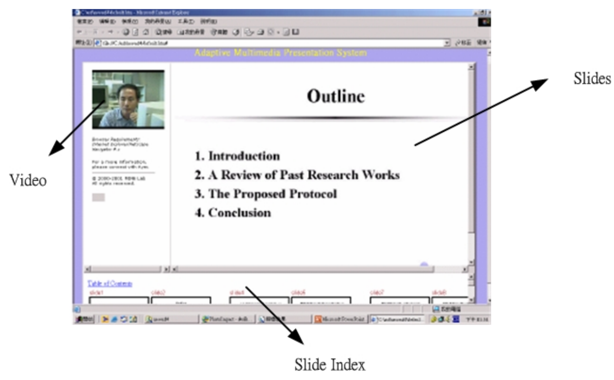


Figure 4 The multimedia presentation system

constructed a tree-liked structure. The proposed system uses it as the video browsing structure.

2.1.1 Interaction and Synchronization Properties

According to Figure 5, we build a timeline-based graph in which we present four different media: audio, image, text, and Video. Here we emphasis on synchronization between the media streams and get a Petri net specification of the slide show in Figure 6. In Figure 6, the scenario includes two reference points. The synchronization of the objects is verified at two reference points.

The Petri Net is particularly well suited to model concurrent processes and control their synchronization and has been extended. “Object Composition Petri Net” (OCPN) [26] and “Extended OCPN” (XOCPN) [27] are two well-known synchronization models. However, they lack detail description of synchronization and interactions from the user in an interactive multimedia system. We need a more powerful model to characterize properties of hypervideo: interaction and synchronization. Another extension has been proposed based on OCPN, called “Distributed Object Composition Petri Net” (DOCPN)[25].

Definition (DOCPN).[25]

A Petri Net structure, D, is a five-tuple, D=(P, T, I, I_p, O)

P={p_1, p_2,…,p_n } is a finite set of places, n≧0.

T={t_1, t_2, …,t_m} is a finite set of transitions, m ≧0.

P∩T=Φ.

I: T->P^∞ is the input function.

I_P: T->P^∞ is the priority input function.

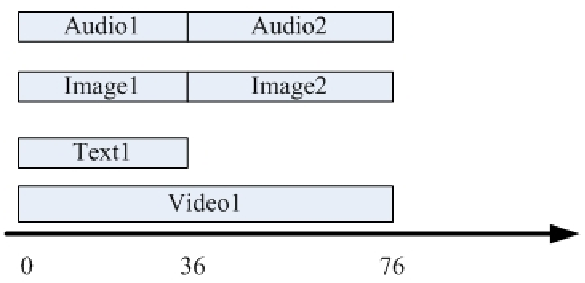


Figure 5 The timeline of different media

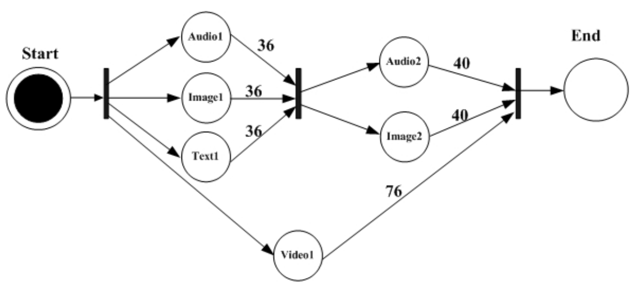


Figure 6 Petri net specification of the slide show

O: T->P^∞ is the output function.

Fire rules:

(1)If all input events are complete, ready and non-priority, a transition only will fire.

(2)If an arrival input events is priority, a transition can fire without waiting for other non-priority input events.

Figure 7 demonstrates the following actions:

First, a major video stream includes three sub-video clips and displays on the screen. At T_0, the viewer turns off an image and triggers a text Text1 at T1. Because the viewer needs to pay attention for this text, the player will be paused the major video from time T1 to time T2. At T2, the viewer closes the text and then the video continues to play video1; Video2 succeeds to display automatically. We explore synchronization and interaction issues at reference points.

And if the auxiliary media is an image, the major video does not to be paused; the viewer will see an image in the foreground, and the major video continues to play in the background. The other auxiliary will affect the playing of the major stream. For example, if a video is triggered, we suspend the major video and a frame of the major video substitutes for the major video at T3 of the Figure 7.

Using the above DOCPN definition, we get a DOCPN map in Figure 8 and define the places and functions in table 1.

In the UI(1,1) of Figure 8, the viewer starts a text that is a priority interaction PUI(1). UI(1,3) Triggers off a text TO(3,1) and suspends the video. UI(1,4) closes the text that triggers off major video to play.

2.2. Multimedia Object

For the proposed authoring tool, the following terms need to be described again:

Object An object is defined by users by using the editor to highlight. An object can be viewed as an anchor.

Node We use the node to construct our tree in Figure 9. A node could be a video clip, an

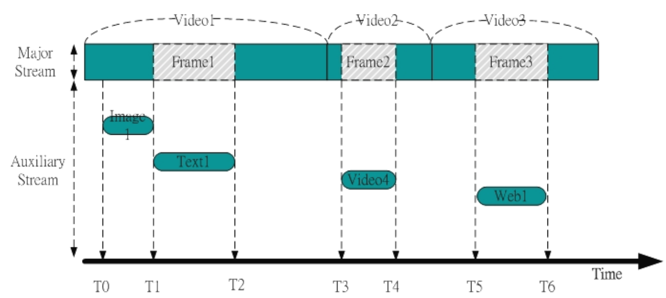


Figure 7 A timeline scenario of hypervideo system

Table 1 The definition of DOCPN map

Places	Total initial place	a unique starting-mark
	Total final place	a unique ending-mark
	Conductor	Control the signals GC
	CIP(Client initial place)	the starting-mark at client
	UIP(User interaction initial place)	Control the user interaction signal
	GC(a)	Global clock sent at time a
	UI(b)	User interaction at time b
	SCIP	Starting place of the reaction
	SCEP	Ending place of the reaction
Priority Input function	PUI(priority user input)	User input events with priority
	PGI(priority global clock input)	Global clock input with priority
Non-priority input function	Is the same as the definition of the Petri net	
Output function	Is the same as the definition of the Petri net	

image, text or a web page.

Scene It differs from the term in the tradition video processing tool. The duration of the scene is determined by an editor user. Actually, we do not focus on how to detect a scene completely and exactly. A scene is a unit while we would like to weave the different video story script.

HyperLink Link structure describes the connection between major video material and other media.

One of the issues is that a user can insert the any kind of material into a video if it is necessary for him/her. In Figure 9, we take a video Clip B as an example for constructing a hypervideo tree. Clip B includes additional information for two video clips-a resource R1 and a hyperlink L1. Video M includes two materials and a hyperlink. One of these materials can be used to link to another video that is placed at higher level. To limit the complexity of the navigating tree, all auxiliary materials of the auxiliary media will be disable. That is, for example, if we trigger Clip M to explain Clip B, we can not see the other auxiliary materials R1 and Video3 to explain M. For some reasons, we skip some video clips to view Clip M immediately by using a hyperlink L1 of Clip B.

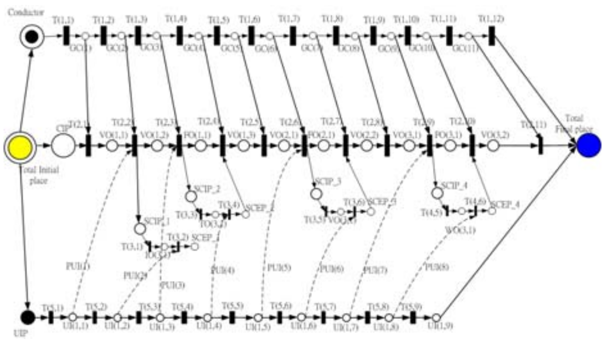


Figure 8 The DOCPN map with the user interaction and hyperlink events

2.3 Temporal and Spatial Relations

Temporal Rules

In such a case, when we watch a scene, how long the duration of the presentation of an object will attract the eyes of a human? The major problem is that a not-long-enough period cannot attract the audience's attention and its function.

- Rule1: the duration of a defined object should be more than a threshold.

Spatial Rules

In designing such a small screen size of 320x240 or 640x480, we found restrictions that come from the audience and users in our experience as following:

(1)The size of the marked area that a user uses to make hyperlink should be limited to a proper size.

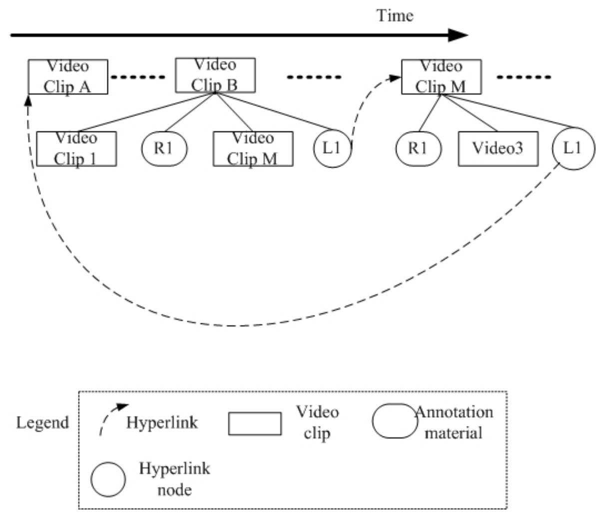


Figure 9 An example of Hypervideo tree

Considering the presentation speed of a video it might not be convenient for an audience to click the link if the marked area is too small.

(2)Two objects can not be close to each other in a frame.

Here we define the rules to solve the spatial problem:

- Rule2: the size of the marking area should be large than a minimum area. I
- Rule3: if the marked areas cross over each other and overlapped area is more than a value, the foreground is enabled and the background is disabled.

Spatial and Temporal Rules

Now we consider a case that there are two marked areas in Frame k and the two areas are not overlapped; in frame k+n, the area size is more than a threshold (as shown in Figure 10).

Combining the above-mentioned rules, we get spatial-temporal rules for hypervideo Authoring:

Rule4: the early marked object is the foreground object if the two objects cross over each other.

3 System Architecture

As shown in Figure 11, in the proposed system, the key issue is how the annotation data can be played into video sequence exactly. We design a hypervideo presentation engine for a hypervideo player. The presentation engine comprises three components:

- Navigation manager: the major component of presentation engine is navigation manager that lets user browse the video sequence and receives the user input (mouse click for video link). And this component also can control the process of the video decode while the user forward or backward the video.
- Video decoder: this component is responsible for decoding video like a MPEG decoder. Actually, we apply the video decoder that installed in the OS directly and can decode a MPEG video sequence.
- Video Render: we use this component to render a video that comes from the output of the navigation manager and the video decoder. So if the one of the inputs to the video render is interrupted, the video render will output the general video sequence without using the hypervideo function. That is, the output data of our hypervideo authoring tool also can be played in other video players.

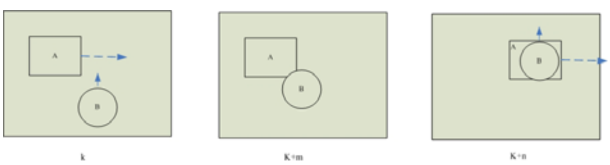


Figure 10 The cross over of two objects

According to our definition, in Section 2.2, a video clip is defined by the user to represent a scene that contains the objects with hyperlinks. Our hypervideo authoring tool consists of two major functions: the annotation function and the hypervideo constructing function.

Video annotation is usually used to enhance the semantic of the video object in the research issue of MPEG-4[28] and MPEG-7[29, 30] and Video retrieval [6, 7]. Before Annotation, it is a big problem that which objects will be recognized, tracked, and annotated. There are many outstanding researchers focus on this challenge and systems are available [8]. Here we will adopt the manual and intuitive way to reduce the complexity of authoring.

Given a raw video file, the producer usually cuts the necessary clips that are story units in a movie. The story unit will be used as the maximum presentation time of an annotated object. That is, the user can define objects in a story unit, but an object cannot be annotated in two sequential units.

The proposed hypervideo authoring tool can be divided into two major components:

- **Object annotation**: We will focus on which objects the user would like to highlight to the viewer. Exactly marking every object in a video sequence is time-consuming and not necessary. For example, in a video, a car might appear several times, and only one is needed to be addressed with other media. We keep

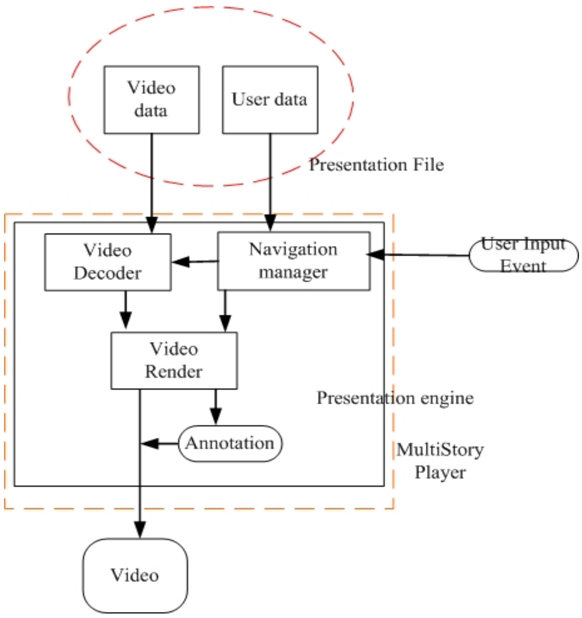


Figure 11 The architecture of proposed player

the work of marking and annotation to the authoring user.

- **Multistory constructing:** Multistory is a kind of hypervideo. However, the user plays the role of the director to organize those clips that he/she cut. Multistory enhances the capability of hypervideo of enriching the linear playing sequence. As in Figure 12, sequence A is an original video with six clips. After our reorganization, we got three different sequences with three cases:

  Case 1: Reducing- the user of the authoring tool can make a story with reduced presentation time. It is useful that this case may be an overview of the original video sequence.

  Case 2: Repeating- for some reasons, a clip can be reused twice or more times. For example, Scene 1 shows a warning message, and the video producer would like to show it before the regular clip starts to play in Sequence C. It is a common case in a TV program to include some limited shows.

  Case 3: Reverse- in the sequence D, we changed the original order and repeated scene6.

As mentioned above, the three cases are basics and the user can do other changes. The key point is that the different scripts create different stories. It means that the big video file will be downloaded once and get playing scripts to view different sequence in distance learning environment.

Usually, the commercial video editing software must take time to merge these clips, and those tools are not suited for an amateur user. In our system, the complicated steps have been simplified. The user just draws and drops without redundant steps and unnecessary functions. Figure 13 shows the process in the proposed system.

First, the author uses the drawing tool to select an object which will be annotated. The system builds some default properties (shown in Table 2). The field is composed of personal annotation attributes and automatically computed attributes.

The personal annotation attributes come from the author's description for the selected object. In the Table 2, the example is used in a distance learning class. Here the significance attribute is created by the instructor to remind students. And the information security classification attribute is required with a default value. This attribute lets the viewers get this annotation information immediately without any password or other secret setting.

Making story tree function lets an instructor organize the video story immediately. Video story is presented as a tree structure in our system. If an instructor wants to make a tree, he/she must initial a root node and then insert the child node of the root. Of course, he/she can delete a node if he/she does not think the node is needed.

Finally, that additional information will be formatted as a data unit. The attached information can be used for many things. For example, it is stored with database and is analyzed to generate sharable learning materials.

3.1 File Format and Example of Learning

For example, in a traditional class room, the environment is very simple. It is impossible that there is a professional cameraman to record the learning video file. So that there must are redundant clips or unsuitable clips in the raw video file. In Figure 14, the raw video data consists of four general clips and one redundant clip and one instructor's comment.

The instructor found that the original playing sequence is not suited to other learners. He/she modified the sequence and deletes some learning course. The revised sequences are described as follows:

(1)Scene 3 and Scene 4 are original, but presentation order is put in the beginning.

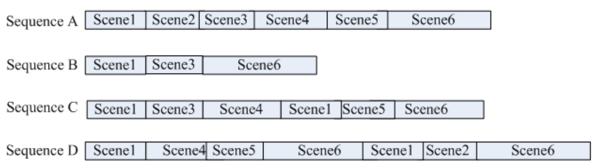


Figure 12 Different playing sequences with the same video material

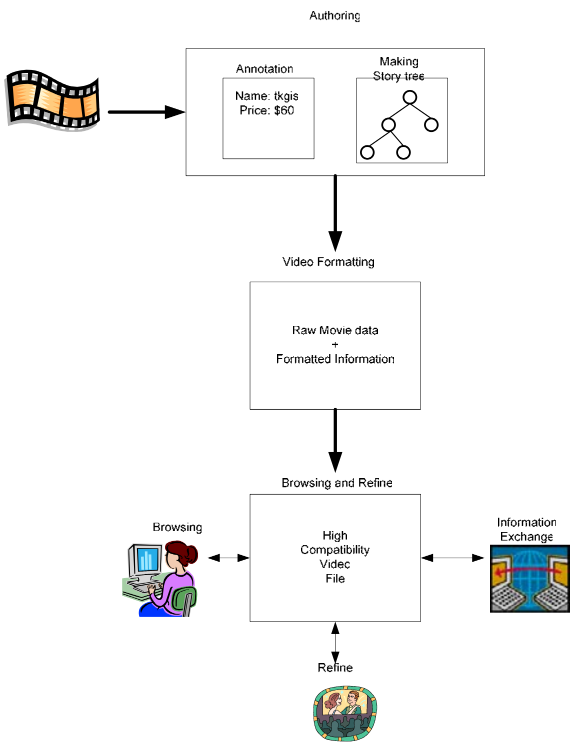


Figure13 Editing process of the hypervideo authoring system

Table 2: Example of attribute list of a video object and the personal textual description of each field.

Attributes list(Text)
Personal Annotation Attributes
Category: Distance Learning Course-Data Structure. Title: Chapter 1 Introduction. Date: 2004/09/09 Author: Mr.Liao Keywords: Tree, Data structure Significance: 1 :
Automatically Computed attributes
Start Point: End Point: :

(2)The instructor puts two course illustrations after Scene 3 and Scene 4. And he/she cuts the illustrations short to fit time restriction.

(3)Because Scene 3 is a key point, the instructor repeats it.

(4)The instructor thinks that he/she can make an instruction for next course. So the original comments of Scene 1 and Scene 2 are cut and re-order.

(5)Scene 1 and Scene 2 are put at the end of the video sequence.

There are two advantages in previous description: a) the instructor can make many scenarios without cutting the real video file directly. b) Those scenarios can be attached to the video file immediately.

3.2 Decoding the Video

The difference between our multistory player and the common player is that in our system the default output video is a MEPG compatible format file. Our Annotation data will be regarded as user-defined data, and the common video player will not show these data if the player lacks our control filter.

The presentation data and the user data are put into one video file. For a common video player, the presentation will be decoded like a general mpeg file with video and audio. The user data will be discarded. But the user data is controlling and annotating data. Our multistory player involves a filter to read those data and personalizes them as a user-defined stream.

The multistory player includes two major components. One is the presentation engine which is regarded as a general decoder. The other component is the navigation manager in which we design two functions. The navigation manager is responsible for reading the user data and controlling the presentation.

3.3 Implementation

3.3.1 Editor

Figure 15 shows the user interface of the video editor. The details are described in the following.

- Object annotation:

  Area A is a view and drawing area in our system. The user can just load a video file and plays it in this area like using a video player. Then using the “mark in” and “mark out” buttons to log the duration that annotated object will present. After determining it, the user draws a notable region and then makes a description for it. This duration also can be used to be a story unit in constructing the story. These pieces will be put into Area B, in

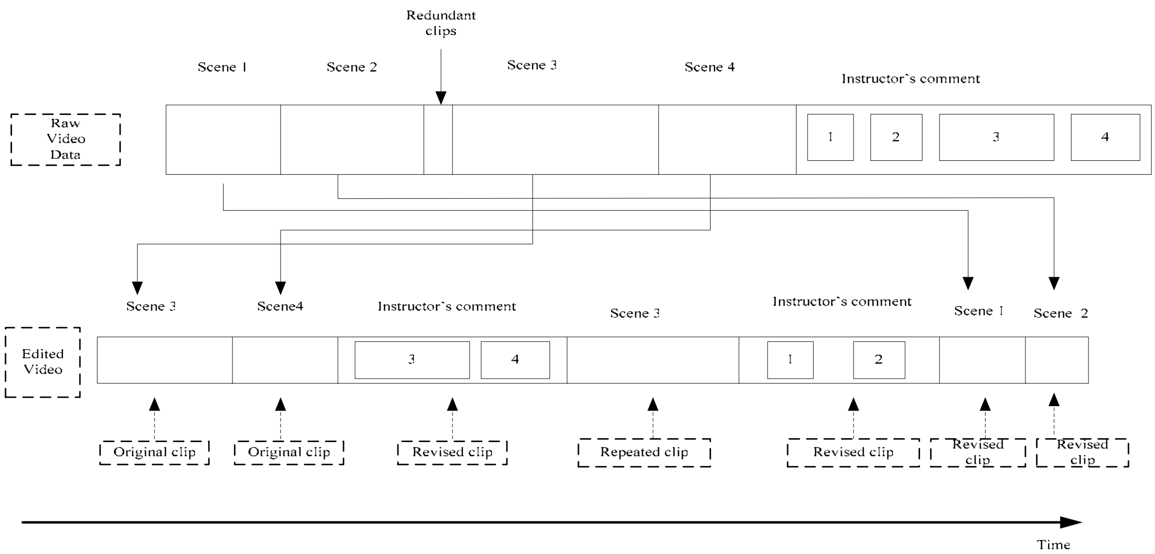


Figure 14 Comparing the raw file and edited video file

which the user stores the story units that are temporarily cut from the original video file. In the Area C, we can preview the auxiliary media such as an image.

- Constructing the story:

We provide a video story board for the user to construct and browse the hierarchy of video in Area D. When a user wants to make a new story tree, he can pick a piece of video from Area B or cut one from the original file.

How can the user connect these pieces? The branch point shall be defined in those pieces, and the user shall determine the duration that includes a region or an object to be a "jump point". Our story board is not simply a collection of one frame, because it is unreasonable to use a frame as a branch point. It is impossible that an audience can catch a frame and select it while a video is playing.

In the user interface, the video producer must have a default traverse. Otherwise, the player will plays the whole video file if no hyperlink is clicked.

3.3.2 Player

As mentioned in the previous section, additional video information can be played only in our system. The developed video player is shown in Figure16. There are a yellow region and a red region that include the additional information. The regions might have been marked as a branch point by the authoring tool. If so, the audience can also click them to jump to other clips. In Area C of Figure 15, an image hyperlink is inserted into the video. So the player presents the inserted image (on the right hand side of Figure 16).

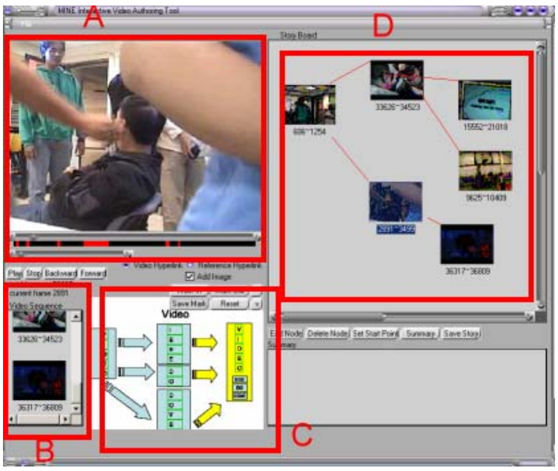


Figure15 The hypervideo authoring tool

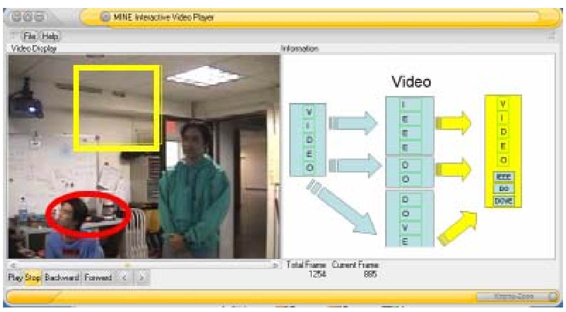


Figure16 The hyrpervideo player

4 Discussions and Conclusions

Video material is one of major components in e-learning and e-entertainment area. It also was used in simple way. For further use, more contents are given by hyperlinks in a hypervideo system. That is, more information can be attached into a video that is useful for e-learning and e-entertainment. Structural video provides different learning paths. An instructor can design many learning paths for different level students and those learning structures can be shared in the network.

A hypervideo system that provides a semantic level annotation and constructs the video presentation structure is introduced. The system is different from other video presentation systems. The major focus is on the objects that are interesting and useful to the video producer and the audience. A video player is also developed for the video viewer to view the annotated film efficiently. The prototype of the hypervideo authoring system can be found at http://www.mine.tku.edu.tw/video/IVAT.htm

In the future work, we will evaluate different interaction methods, for example, by a remote control or by a gesture. We also will add SCORM into out system.

References

[1] MPEG Homepage: http://www.chiariglione.org/mpeg/

[2] International Telecom Union Homepage: http://www.itu.int/home/

[3] Jim Taylor, "*DVD Demystified*", McGraw-Hill Professional (2nd edition), December 22, 2000.

[4] Ralf Steinmetz and Klara Nahrstedt, "*Multimedia: Computing, Communications and Applications*," Prentice Hall PTR, (1st edition), July 27, 1995.

[5] Nitin Sawhney, David Balcom, and Ian Smith, "*HyperCafe: narrative and aesthetic properties of hypervideo*," Proceedings of the the seventh ACM conference on Hypertext, March 1996.

[6] Shih-Fu Chang, William Chen, Horace J. Meng, Hari Sundaram, and Di Zhong, "*VideoQ: an automated content based video search system using visual cues*," Proceedings of the fifth ACM international conference on Multimedia, November

1997.

[7] M. Bertini, R. Cucchiara, A. Del Bimbo, and A. Prati,"*Object and event detection for semantic annotation and transcoding*," Proceedings of International Conference on Multimedia and Expo, Vol. 2, 6-9 July 2003.

[8] N. Dimitrova and Hong-Jiang Zhang, "*Applications of video-content analysis and retrieval*," Vol. 9, Issue 3, IEEE Multimedia, 2002.

[9] http://www.research.ibm.com/mpeg4/Projects/authoring.htm

[10] F. Shipman, A. Girgensohn, and L. Wilcox, "*Creating navigable multi-level video summaries*," Proceedings of the International Conference on Multimedia and Expo, Vol. 2, 6-9 July 2003.

[11] Andreas Girgensohn, Lynn Wilcox, Frank Shipman, and Sara Bly, "*Extending to multidimensional interfaces: Designing affordances for the navigation of detail-on-demand hypervideo*," Proceedings of the working conference on Advanced visual interfaces, May 2004.

[12] John Doherty, Andreas Girgensohn, Jonathan Helfman, Frank Shipman, and Lynn Wilcox, "*Detail-on-demand hypervideo*," Proceedings of the eleventh ACM international conference on Multimedia, November 2003.

[13] Andreas Girgensohn, Frank Shipman, and Lynn Wilcox, "*Hyper-hitchcock: authoring interactive videos and generating interactive summaries*," Proceedings of the eleventh ACM international conference on Multimedia, November 2003.

[14] Wendy E. Mackay, and Glorianna Davenport, "*Virtual Video Editing in Interactive multimedia Applications*," Vol. 32, No. 7, Communications of the ACM, July 1989.

[15] Yoshiaki Hada, Hiroaki Ogata, and Yoneo Yano, "*XML-based Video Annotation system for Language Learning Environment*," Proceedings of the Second International Conference on Web Information Systems Engineering, 3-6 Dec. 2001.

[16] Joan L. Mitchell, William B. Pennebaker, Chad E. Fogg, and Didier J. Legall, "*Mpeg Video: Compression Standard (Digital Multimedia Standards Series)*," Kluwer Academic Publishers, October 1996.

[17] Jim Taylor, "*DVD Demystified*", McGraw-Hill Professional (2nd edition), December 22, 2000.

[18] Timothy K. Shih, Lawrence Y. Deng, J. H. Wang, and Andy Ay-Hwa Liou, "*Maintaining Persistent Look-and-feel for Roaming Student with Mobile Agent in Distance Learning*," in Proceeding of the 7th International Conference on Distributed Multimedia Systems (DMS' 2001), Taipei, Taiwan.

[19] Timothy K. Shih, and Lawrence Y. Deng, "*Distributed Multimedia Presentation with Floor Control Mechanisms in a Distance Learning System*," in Proceeding of the 2000 International Conference on Information Society in the 21 Century: Emerging Technologies and New Challenges, Japan.

[20] Kuo-Yu Liu, Natalius Huang, Bo-Hung Wu, Wei-Ta Chu, Herng-Yow Chen, "*The WSML System: Web-based Synchronization Multimedia Lecture System*," ACM Multimedia'02, France

[21] Sugata Mukhopadhyay and Brian Smith, "*Passive Capture and Structuring of Lectures*," ACM Multimedia'99, Orlando, FL, USA.

[22] Synchronized Multimedia Integration Language (SMIL) 2.0 Specification, http://www.w3.org/.

[23] http://www.real.com/

[24] http://www.microsoft.com/

[25] Sheng-Uei Guan, Hsiao-Yeh Yu, and Jen-Shun Yang, "*A prioritized Petri net model and its application in distributed multimedia systems*," IEEE Transactions on Computers, Vol. 47 , Issue 4 , April 1998.

[26] T.D.C. Little and A. Ghafoor, "*Synchronization and Storage Models for Multimedia Objects*," IEEE J. Selected Areas Comm., Apr. 1990, pp. 413-427.

[27] M. Woo, N.U. Qazi, and A. Ghafoor, "*A Synchronization Framework for Communication of Pre-orchestrated Multimedia Information*," IEEE Network, Jan. Feb. 1994, pp. 52-61.

[28] J. M. Martinez, "*Authoring 744: writing descriptions to create content*," Multimedia, IEEE, Vol. 10 , Issue 4 , Oct-Dec 2003.

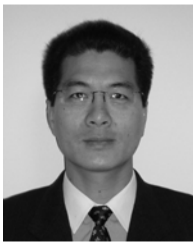
[29] Mark van Setten and Erik Oltmans, "*Demonstration of a distributed MPEG-7 video search and retrieval application in the educational domain*," Proceedings of the ninth ACM international conference on Multimedia.

[30] Ching-Yung Lin, Belle L. Tseng, Milind Naphade, Apostol Natsev, John R. Smith, "*MPEG-7 video automatic labeling system*," Proceedings of the eleventh ACM international conference on Multimedia.

Biographies

Yi-Chun Liao is now a Ph.D. student at Computer Science and Information Engineering at Tamkang University, Taiwan, R.O.C. His research interests include Multimedia Computing and Networking, Distance Learning, Software Engineering. Yi-Chun Liao received his BS and MS degrees in Computer Engineering from Tamkang University in 1998 and 2001, respectively.

Han-Bin Chan*g* is now a Ph.D. student at Computer Science and Information Engineering at Tamkang University, Taiwan, R.O.C. His research interests include Multimedia Computing and Networking, Distance Learning, Software Engineering. Han-Bin Chan*g* received his BS and MS degrees in Computer Engineering from Tamkang University



Hui-Huang Hsu received the B.E. degree from Tamkang University, Taipei, Taiwan in 1987, and the M.S. and Ph.D. degrees from The University of Florida, Florida, USA in 1991 and 1994, respectively. All the degrees are in electrical engineering.
He worked as a research assistant in the Department of Computer and Information Sciences at The University of Florida throughout his Ph.D. study and stayed there as a postdoctoral research fellow from January to May in 1995. From August 1995 to July 2003, he was with Chung Kuo Institute of Technology (1995~2000), Takming College (2000~2001), and Chinese Culture University (2001~2003) all located in Taipei, Taiwan. He joined the Department of Computer Science and Information Engineering, Tamkang University in August 2003, where he is now an associate professor. He has participated in the organization of several international conferences and workshops. He is the editor of a book titled "Advanced Data Mining Technologies in Bioinformatics" to be published by Idea Group Inc., Hershey, PA, USA in 2005. His current research interests are in the areas of bioinformatics, multimedia processing, data mining, and e-learning.

Dr. Hsu is a member of The Institute of Electrical and Electronics Engineers (IEEE) and a life member of The Institute of Information and Computing Machinery (IICM), Taiwan.

Timothy K. Shih is a Professor of the Department of Computer Science and Information Engineering at Tamkang University, Taiwan. He is a member of ACM. As a senior member of IEEE, Dr. Shih joins the Educational Activities Board of the Computer Society. His current research interests include Multimedia Computing and Distance Learning. He was a faculty of the Computer Engineering Department at Tamkang University in 1986. In 1993 and 1994, he was a part time faculty of the Computer Engineering Department at Santa Clara University. He was also a visiting professor at the University of Aizu, Japan in summer 1999, a visiting researcher at the Academia Sinica, Taiwan in summer 2001, and an adjunct faculty member of Xidian University, China. Dr. Shih has edited many books and published over 340 papers and book chapters, as well as participated in many international academic activities, including the organization of more than 30 international conferences and several special issues of international journals. He is the founder and co-editor-in-chief of the International Journal of Distance Education Technologies, published by Idea Group Publishing, USA. He is also the associate editor of IEEE Transactions on Multimedia and ACM Transactions on Internet Technology. Dr. Shih has received many research awards, including research awards from National Science Council of Taiwan, IIAS research award of Germany, and several best paper awards from international conferences. Healso received many funded research grants from both domestic and international agencies. Dr. Shih has been invited to give more than 10 keynote speeches and plenary talks in international conferences, tutorials in IEEE ICME 2001 and ACM Multimedia 2002, and talks at international conferences and overseas research organizations. Publications, demonstrations and contact address of Dr. Shih can be retrieved from
http://www.mine.tku.edu.tw/chinese/teacher/tshih.htm.