F. Workshop on Multimedia Technologies

Title : An e-Video Abstract Model for Flexible Browsing of On-demand Video

Abstract

In this research, we adopt the method of structured scripts to facilitate the parsing and browsing of video programs. With the help of the e-Video abstract model, we are able to create a video database out of the metadata collected from the original script. Users are given a query tool to obtain a browsing map similar to a Web page. By clicking on the map, the selected video component will be streaming down to users' workstation from the video server for non-linear browsing.

Contact author : Yen, Cheng-Huang
Dept. of Management & Information
National Open University
172 Chung-Cheng Road, Lu Chow City, Taipei County
Taiwan, ROC, 24702
yench@mail.nou.edu.tw
Tel. 02-2282-9355-7613
Fax. 02-2848-4944

Keywords : e-learning, structured script, hypermedia, video parsing, video browsing

Most people are used to watch a movie from beginning to end without knowing the outcome in advance. However, browsing an instructional program offers quite different experience since it usually takes a few back and forth reviews before the introduced concepts are fully understood by viewers. From learners' point of view, video programs would be much easier to browse if they can be made just like normal Web pages with well-organized hyperlinks. This is the notion of e-Video. Before the concept of e-Video can be realized, we must seek a way to partition the video from studio production. The approach must be helpful to learners and feasible in terms of cost and efficiency.

Introduction

From learners' point of view, the review of instructional videos is mostly done in a linear fashion. Although non-linear browsing of videos are likely to help more in the learning process, poor support can be found in existing video browsing tools or equipments. In this paper, we describe the development of the e-Video model that enables non-linear and on-the-fly browsing of instructional videos. The browsing of on-demand video is still linear [2, 6]. The notion of video abstract [14] is closer to the concept of non-linear browsing. If the items in the abstract are implemented as hyperlinks, every item can be mapped to a video segment. In other words, viewers are given hyper-linked items as guidance to navigate a video program. Now, the question is where these items are coming from. The answer is content-dependent metadata.

Video data can be analyzed and structured to identify components that are then used as the basis for creating indices [10]. Indexing video data leads the way to efficient retrieval and browsing. Metadata is an important technique to support video structuring and indexing. Figure 1 provides a simple taxonomy for metadata types. Tags embedded in multimedia data are good choices of indices. In the domain of instructional videos, we have chosen content-dependent metadata since they are likely to provide better guidance for video browsing. Most studio production has relied on scripts to provide major content. However, conventional scripts are non-structured in the sense that audio and video part is synchronized in a linear manner. Script writers do not foresee future needs for video partition or indexing. The so-called structured scripts are written to provide video content as well as metadata for later video processing after the studio production [12]. In our approach, the use of tags in scripts plays an important role. The tags are sources of metadata.

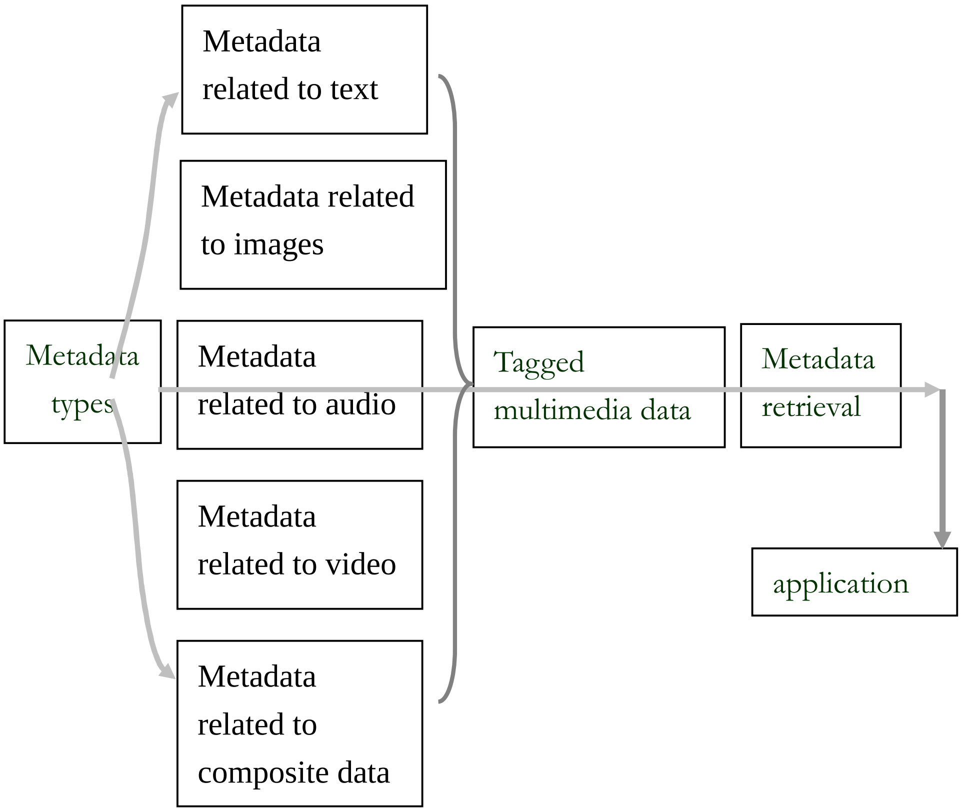

Figure 1. different categories of metadata

The tags embedded in structured scripts come from a traditional symbolic language for scripting. We add a few extensions to this language to help automate the partition of video programs. The tags from structured scripts form a searchable name space. The time code corresponding to each tag becomes the exact point of partition. Therefore, we can define the attributes of a video component by its associated tags and time codes. The e-Video database can then be created from the set of video components. The query against the e-Video database is content-based in the sense that the tags are well-defined and are related to the structure as well as the content of the program topics. The result of the query is a browsing map constructed by inter-relating retrieved video components. Our research is divided into two major parts as depicted in Figure 2. Structured scripts deal with the use of tags in the script stage. The e-Video abstract model bridges the storage world with the application world [11].

The original intent of our research is to facilitate non-linear browsing of video without much monetary spending in both software and hardware establishment. For most distance education institutions, the change made in the script stage is feasible since the entire video production process has been conducted according to disciplined guidelines. The rules of structured scripts can be made as simple as possible in order to be used as a general guidance for script writers. Therefore, our approach can easily be integrated into the existing process. For existing videos to be part of the video database, we suggest a reproduction of the original scripts in order to create indices from tags. Whether this method is feasible would be up to an evaluation on cost and efficiency.

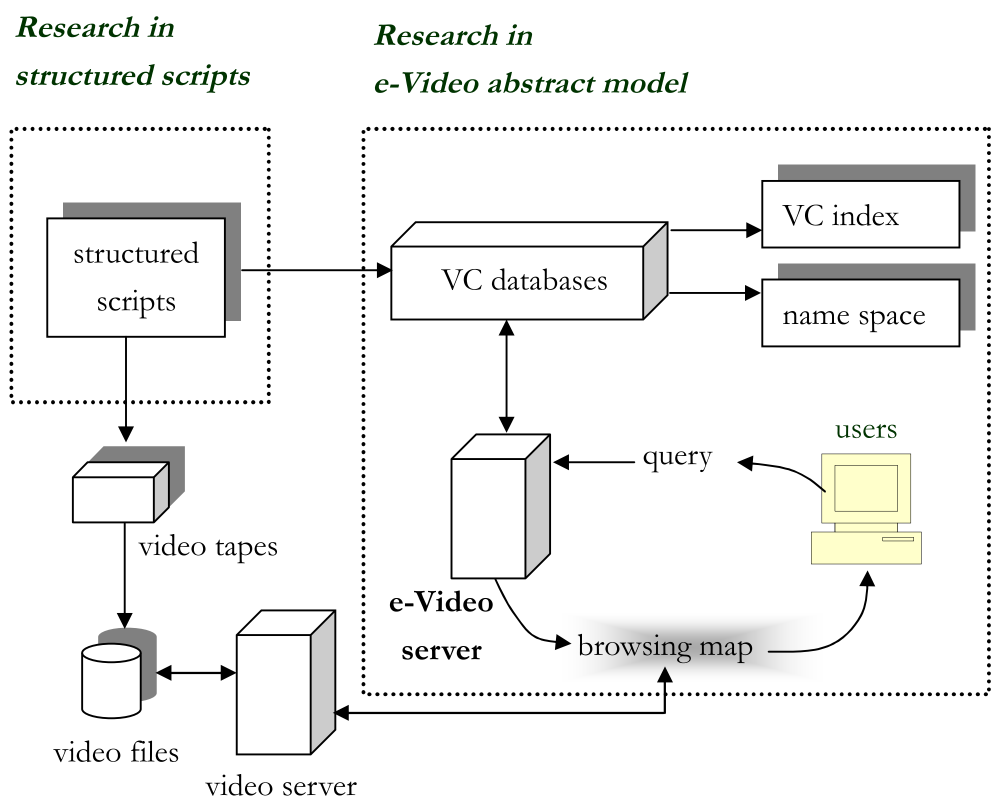

Figure 2 major components of e-Video research

Related Research

Much effort has been devoted to flexible and easy use of existing audio-visual information [3, 4, 14]. The notion of video abstracting [13] tries to construct video abstract based on video content. The result is similar to a tree-like structure made of video frames. Video abstract has minimum storage requirement. Video databases use video abstract to organize large quantities of video files. Another research direction relies on text-based information retrieved from video programs. For example, automated re-construction of text from a PowerPoint presentation appeared in a video is a typical example [7]. Audio recognition is also a way to produce text information from video programs.

The planned production of video in distance education institutions requires formal scripts and strict control of audio and video at the level of broadcast quality. Simply digitized the produced video for retrieval does not fit the needs of most learners [5, 7]. Automatic audio recognition is not quite feasible since the scale of quantity will soon exceed affordable limit of cost. Changing the way the scripts are written and making good use of embedded metadata is a much better choice when all circumstances have been considered.

Methodology

Figure 3 summarizes our approach. Structured scripts provide metadata that can be parsed, abstracted and analyzed to produce indices or video summary. The retrieval or query of video can be done in a way that supports non-linear browsing. Before we are able to comment more on our approach, let's first explain via a script example.

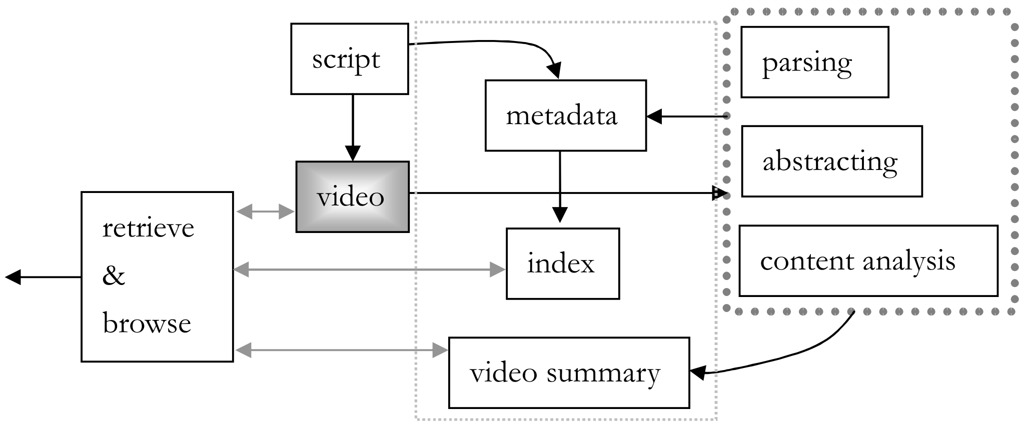

Figure 3 video data indices for video data retrieval

Table 1 lists a sample script. The symbols used in scripts are traditional [8, 9]; e.g., SP for superscription, TC for tele-card, etc. We add extensions like DTC for divider TC, DVC for default video component, IAC for improvised audio component, etc. Structured scripts can be composed in a script editor. The script editor is a kind of computer software that is capable of generating more output from structured script. Based on a script, the video director knows how to conduct studio production. Based on a structured script, a computer technician knows how to place the video into a video database for non-linear browsing.

Table1. a script sample

Video part	Audio part
DTC Getting to know Java	(5 seconds music)
	Java is a programming language and most of us have heard of it for a long time.
TC Java features	
	Java is similar to C/C++ in syntax and semantics with some features of its own :
	1. Java is object-oriented.
	2. Java programs are portable. (Java's ability to provide
SP	cross-platform support is explained in Figure 1.)
Figure 1 Java platforms	3. Java does not support the use of pointers.

Figure 4 shows a list-like output of the script in Table 1. Every element in the list is composed of an audio part and a video part. Since video programs normally use time codes to identify a spot in the video timeline. We should be able to identify each list element with a starting time code and an ending time code.

tag

(DTC Getting to know Java + (5 seconds music))(DVC + Java is a programming language and most of us have heard of it for a long time.)(TC Java features + Java is similar to C/C++ in syntax and semantics with some features of its own : (1. Java is object-oriented. + IAC) (2. Java programs are portable. Java's ability to provide (SP cross-platform) support is explained in Figure 1.(Figure 1 Java platforms)) (3. Java does not support the use of pointers. + IAC))

Figure 4 List-like output of structured scripts

The list-like representation is suitable for computer processing. From the conceptual level, we can generate a video abstract based on the tags appeared in the structured script. Figure 5 shows such an abstract. Each node in the abstract corresponds to a video segment. This is how the browsing prototype of e-Video is constructed.

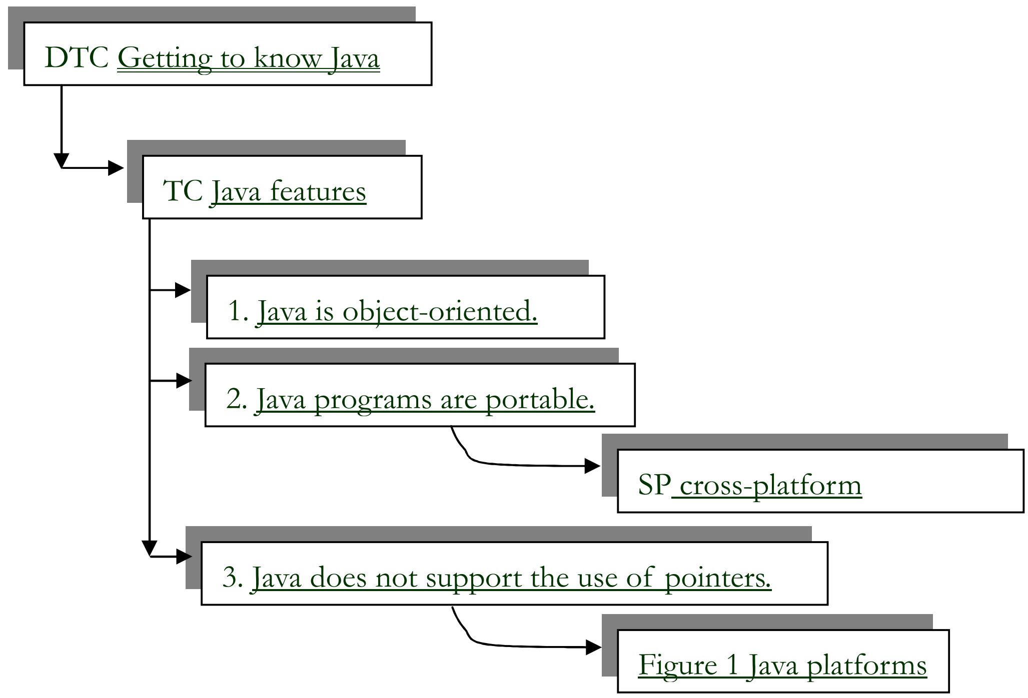

Figure 5 the hierarchy of tags in structured script

Figure 5 is an example of linear partition of video. Structured scripts make such partitions more beneficial to learners. Actually, there are different ways for partitioning. For example, a pair of time code, $<tc_1, tc_2>$, denotes a video component. A group of pairs of time codes denotes a stratum; e.g., $stratum_x=\{<tc_3, tc_4>, <tc_5, tc_6>\}$. Stratafication groups video components based on certain relationship [1, 13]. The notion of browsing maps is a generalized model for different types of partition methods.

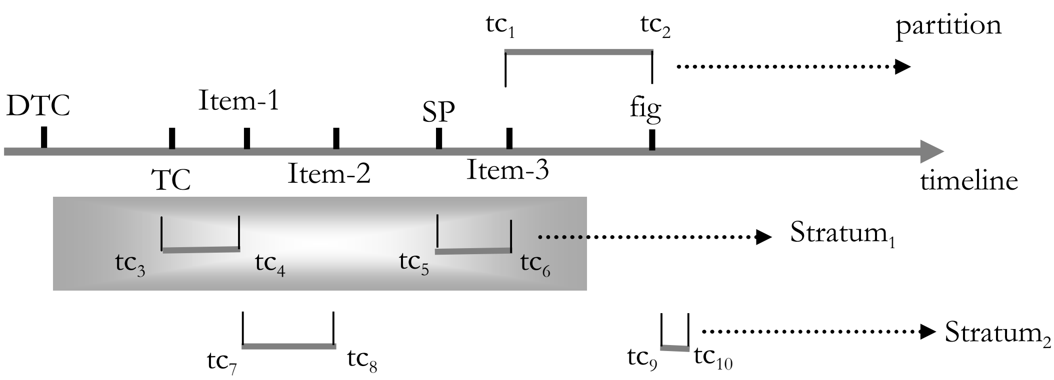

Figure 6 the method of video partition

At the storage level, the video file corresponding to each program needs not be divided. The hyperlinks in the browsing map record accurate time codes to retrieve video components with low-level file access techniques. This allows our methodology to survive in a moderate data rate environment. Compared to other audio-visual processing methods, our research has proposed a feasible and efficient way for distributing video programs. And more importantly, the browsing flexibility matches the needs of users very well in the coming era of e learning. Figure 7 demonstrates the layered concept in e-Video. Video files are at the physical layer. Time codes serve as indices for random access for video files. Structured scripts are part of the abstract layer and hyperlinks correspond to the application layer.

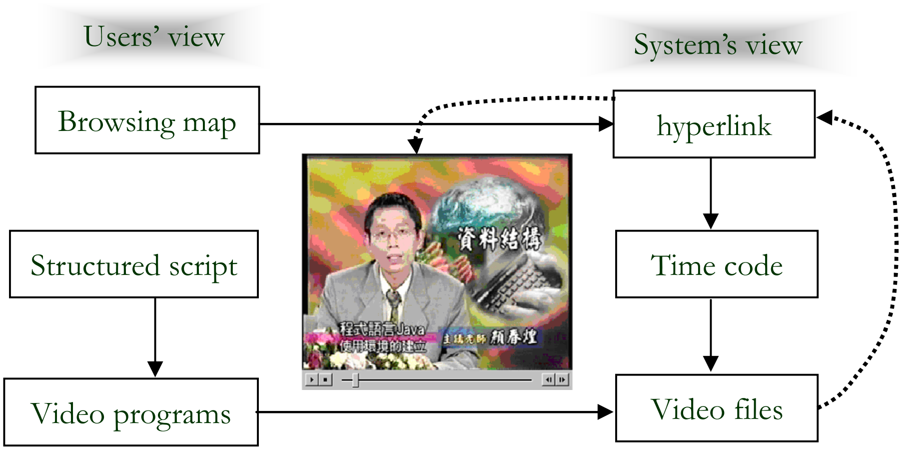

Figure 7 layered concept in e-Video

Since video programs may exhibit many different topics in different ways, a turning point in a video normally raises a change in video scenes. To make these turning points more evident, structured scripts adopt the following scripting policies :

1. Each video component must have a name.
2. The names of video components are highly relevant to introduced topics.
3. The audio and video part has evident turning point.
4. The design of video component should be consistent with or closely related to the structure of the content.

An Abstract Model for Structured Scripts

The collection of digitized audio-visual programs will soon build up a large information warehouse [1, 2]. However, e-Video users interact with systems through browsing maps. Therefore, there must be a model that provides interfaces between the warehouse and browsing maps. The key to the model is the notion of video components. A video component can be a video file, a video segment or a single shot. In our research, a video component, as known as a VC, is defined as a self-sufficient video clip. The abstract model represents a VC as follows.

VC : (id, name, tc_start, tc_end, offset, tag_type, tag_no)

- id : id captures content-independent information; e.g., the score number.
- name : name is used to create a name space. Video components are allowed to have same names. However, id is unique.
- tc_start and tc_end denote the starting point and the ending point of a VC respectively.
- offset denotes the relative starting position of a VC in a video output. The relative ending position should then be offset+(tc_start-tc_end).
- tag_type corresponds to the tag that divides the VC. tag_no denotes whether the VC is contained in another VC.
- Search space and browsing maps are grouped sets of VCs.

Relationships between VCs

Items in browsing maps are associated based on relationships among VCs. In other words, there exist different types of relationships that make the construction of browsing maps easier.

1. Containment relationship : VC_i contains VC_j iff VC_i.id=VC_j.id and VC_i.tc_start <= VC_j.tc_start and VC_i.tc_end >=VC_j.tc_end
2. Overlap relationship : VC_i overlaps with VC_j iff VC_i.id=VC_j.id and VC_i.tc_start < VC_j.tc_start and VC_i.tc_end <VC_j.tc_end
3. Locality : locality is identified by id, tag_type or tag_no.
4. null relationship : denotes the situation when there is no relationship identifiable from the abstract model.

Among attributes of VCs, id and tag are used as index keys and time codes are used for addressing. The VCs in browsing maps contain these indices for retrieving video segments. The scripting policies help increase the number of relationships existing among VCs. In our model, relationships are automatically identified instead of being defined or inserted.

About Querying and Browsing

In the e-Video abstract model, a query can be viewed as the process of constructing a result browsing map from a global browsing map. If the VC database is implemented as a relational database, every VC becomes a data record and SQL syntax can be used to query the database. However, the intent of an e-Video query is to help learners. Therefore, the browsing map obtained from the query fits better with learners' needs than pure SQL queries. Figure 8 depicts the separation of query and browsing interfaces.

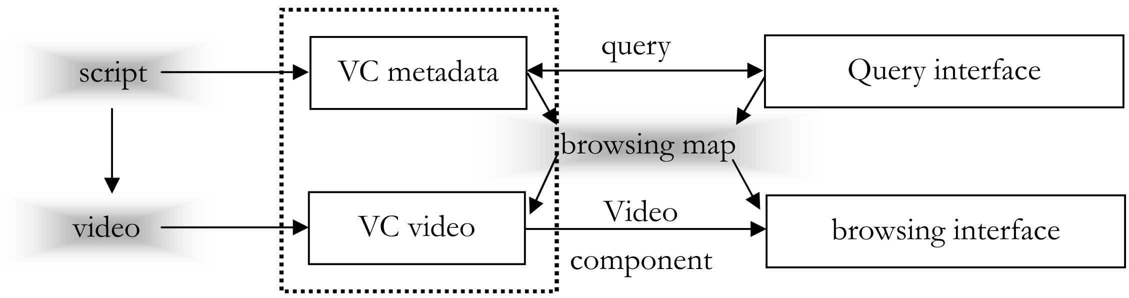

Figure 8 the separation of query and browsing interfaces

Implementation Considerations

In our approach, the entire process of video production starts with structured scripts. Theoretically, a well-defined script can be transformed into a list-like data structure without difficulty. A valid script is written according to scripting policies. Script editors can be used to check whether a script is well-defined. Most instructional video programs contain limited amount of content and thus avoid possible difficulties during the processing of VCs due to unacceptably high time complexity of processing programs.

Conclusions and Future Work

Structured scripts change the way video programs are viewed by learners. It also provides a basis for porting video learning resources to computer and network platforms. The e-Video abstract model integrates database techniques with video processing applications. There are several potential areas where e-Video can be applied or extended though there are also new issues arise. For example, suppose the video components belong to different people, legal issues should be resolved. Besides instructional video programs, e-Video may also be suitable for other types of video programs where its market value can be enhanced.

References

1. Baker, A. Multimedia and Imaging Databases. Morgan Kaufmann, 1996.
2. Chang, S. et al. Ed. Video On Demand Systems : Technology, Interoperability and Trials. Kluwer Academic Publishers. 1997.
3. Chua, T. ``A Video Retrieval and Sequencing System,” ACM Transactions on Information Systems, Vol. 13, No. 4, Oct., 1995, pp. 373-407.
4. Guan, L., et al., Ed. Multimedia Image and Video Processing. CRC Press, 2000.
5. Horton, W. Designing Web-based Training. Wiley, 2000.
6. Hwang, J., et al., ``Video Browsing for Course-on-Demand in Distance Learning,” Int’l Conf. on Image Processing, Vol. II, pp. 530-533, Santa Barbara, CA, Oct. 1997.
7. McCormack, C., et al. Building a Web-Based Education System. Wiley Computer Publishing, 1997.
8. Rubin, B. ``Structured Content Modeling for Cinematic Information,” SIGCHI Bulletin, Vol. 21, No. 2, 1989, pp. 78-79.
9. Smith, A. ``Parsing Movies in Context,” USENIX, 1991, pp. 157-168.
10. Thuraisingham, B. Managing and Mining Multimedia Databases. CRC Press. 2001.
11. Yen, C. H. 2001. `` e-Video for Better Learning – Create an Abstract Model for Structured Scripts,” Vol.4 , No. 3, Special issue on e-learning. , Communications of IICM. pp. 53-62.
12. Yen, C., ``The Production of Web-based Interactive Video From Structured Script," ICCE/ICCAI 2000, Taipei, Taiwan, Nov., Volume 1, pp. 855-861.
13. Zhang, H. J., ``Video Parsing, Retrieval and Browsing: An Integrated and Content-based Solution,” ACM Multimedia, 1995, pp. 15-23.
14. A Special Issue on Visual Information Management. Communications of the ACM. Dec. 1997, Vol. 40, No. 12.