Implementation forEducation Data Management System based on Remote Control

Byeongtae Ahn*

Liberal & Arts College, Anyang University, Anyang-shi, South Korea. Email: ahnbt@anyang.ac.kr

Abstract: The research is subject for effective remote education using multimedia, it is necessary to develop efficient management techniques of video information. This requires real-time processing of video information which should be managed and retrieved in a compressed forms. The main technology of compressing video is currently MPEG-2. This implies that it is very important to manage and retrieve video compressed in MPEG-2, and then to process the video in real-time for the remote education environment using multimedia. This paper is to develop the management system of video information which is one of the most critical requirements in remote education systems for managing and retrieving MPEG-2 video
Keywords :Video, Remote Education, Mpeg, Semantic, Image

I. INTRODUCTION

Recently, as the demand for retrieval of moving picture information has increased rapidly and various object-oriented database systems have been developed, these can be utilized in multimedia systems. Accordingly, research on a technique for efficiently storing and retrieving video information and modeling of a video database is being conducted [1].

However, the management of moving picture information using such a multimedia DBMS is a method of managing the bitmap or wave pattern to be searched in an uncompressed state. However, due to the nature of moving pictures, it is difficult to store, search, or transmit uncompressed natural images [2]. Therefore, in order to solve these problems and put them into practice in a video information management system, a technology capable of searching and transmitting video information in real time in a compressed state is required, and a data model capable of describing a general video document reflecting this is required. These demands mainly use one of two approaches for retrieving video information. In other words, there is a tendency to select and use content-based search and annotation-based search [3]. Content-based search automatically extracts feature data such as shape, color, texture, movement, etc. from moving image data and searches based on this [4]. On the other hand, annotation-based search is a method of first grasping the meaning of video data, expressing it using natural language, and searching based on this. In this paper, we present a general data model that can extract the structure of a general video document and give meaning to the indexing of a video stream. In addition, a compressed video information management system (CVIMS, Compressed Video Information Management) that extracts text information about a video compressed with MPEG-2 technology, stores it in a database system, and enables search using query words or representative images (key frames). System) was developed. In addition, by analyzing general video data, an integrated video data model (IVDM) that can use both annotation-based and content-based search was presented [5].

The structure of this paper is as follows. Chapter 2 introduces related studies, and Chapter 3 describes the generic data model for video documents presented in this study. In Chapter 4, the design contents of the compressed video information management system (CVIMS) developed using the proposed model are described, and in Chapter 5, the actual CVIMS is implemented. Finally Chapter 6 presents conclusions and future tasks.

II. RELATED WORK

How to search for video data can be divided into a search based on content and a search based on annotations [6].

Recently, a multi-layered video model (MuVi) was proposed for a search in the form of integrating these two techniques [7]. However, the video data itself, the assigned meta information, and the extracted feature information are mixed to form a straight layer, and modeling for an annotation-based search, which is a concept layer, is insufficient.

In this paper, unlike the MuVi model, which divides the video's physical structure into a meaningful concept layer, based on the logical data of the video, the user's needs can be reduced from a wide range to a narrow range through several steps at a higher level. A method of segmentation to give annotations (used for annotation-based search), and segmentation so that features of video data can be extracted at the lower level (used for content-based search) was used [8]. <Table 1> compares and analyzes the MuVi model that is most similar to the IVDM model proposed in this study.

Table 1Compare of IVDM &MuVi

	MuVi	**IVDM**
Search techniques supported	Annotation-based search, content-based search, integrated search	Annotation-based search, content-based search, integrated search
Integrated Search Approach	A sub-concept hierarchy is introduced to solve the difference in semantics that occurs when the content-based is directly mapped to the annotation-based.	Reducing the search range from a wide range to a narrow range (upper level: annotation-based search, lower level: content-based search)
Video segmentation process	Abstraction process (Physical data -> semantic data)	The process of materialization (Logical data -> physical data)
Query type	5 things	4 things
Advantages	Independence by layer, content-based search by feature vectors and concepts	Independence by query type, database schema structure can be created only with IVDM
Complement	More detailed description of the conceptual hierarchy and data structure is required.	A sophisticated framework is needed to exclude subjective annotations.

Recently, there has been active research on an efficient index generation technique for moving video data compressed with MPEG [9]. These are mainly methods of constructing an index by analyzing the contents of a compressed video and automatically selecting a key frame. However, when making a search system by analyzing the contents of an actual video, the efficiency is somewhat inferior because the contents are not yet reached at the stage where the details are parsed and used for search.

Therefore, in this study, we developed an extended video data model including metadata based on MPEG-2 and Dublin Core based on the previously introduced video data model, and implemented a system that can search video information based on this model. 10].

III. Extended Video Data Model

In order to manage the video itself and metadata, which is a description of the content and structure of the video, through a database, a general data model for video should be provided [11]. However, this general model need not exist for each application. The general video data model proposed in this paper uses a segmentation approach to define the structure of video documents by defining the level of abstraction.

This enhanced Generic Video Data Model (EGVDM) is a frame that provides functions of structuring video data, free annotation of video data, and sharing and reusing video data. It's a walk. In EGVDM, moving picture data is a contiguous group of frames called a stored video segment [12].The proposed EGVDM was made with the concept of structural components related to the frame sequence of the moving picture medium based on the film theory and segmentation of the moving picture medium [13]. The concept of structural components is further subdivided into complex units, sequences, scenes, and shots, and these subclasses are defined in hierarchical relationship with each other. A shut consists of one or more consecutive frames and appears as a spatiotemporal successive action. A scene consists of several shots, and a sequence is composed of these scenes. The collection of related sequences again constitutes a compound unit, and the compound unit can refer to itself at any level. In this EGVDM, an annotation can be added to an arbitrary frame sequence. Since these annotations are independently composed of arbitrary structural components, thematic indexing is sufficient to describe the structure of a video document [14].

Frame sequences are classified into annotation objects and key frame objects. The key frame object is composed of image, image type, frame number, size, and location information by extracting a specific representative image from the frame sequence, and the annotation object is composed of subclasses of object annotation, person annotation, location annotation, and event annotation. do. Object annotation consists of object type and object description. In this paper, object annotation is defined using metadata based on Dublin Core. That is, object annotation is defined as title, subject, identifier, relation, right, language, document format, etc. [15].

Therefore, EGVDM satisfies this demand by defining the contents of the video document as a logical concept. That is, the connection between the moving picture stream and the medium physically stored in the database is configured as a set of events on the stored moving picture segment, and the same moving picture medium is allowed to be used in various ways depending on how it is expressed in the database. In addition, by using annotations and structural information related to the video stream, it is possible to access different video media that are stored and used in the same video media.

IV. Design of CVIMS

Compressed video information management that extracts key frames from video data compressed in MPEG-2 based on the video data model presented above, gives caption and picture description form annotations, and stores them in the database in text format for management. Designed the system (CVIMS).MPEG-2 video compression files are largely composed of three types: I-frame, P-frame, and B-frame [16]. Among these, I-frames are frames compressed using only spatial compression techniques without using temporal compression techniques. Therefore, the I-frame can be a reference frame because it can be decoded independently and can be accessed randomly. So, in CVIMS, it is assumed that all I-frames in a video file compressed with MPEG-2 can be candidates for key frames. Based on this assumption, CVIMS provides a method for users to directly select key frames by extracting I-frames from MPEG-2 compressed video [17]. In addition, the search for each video is not searched by actual frames, but caption information for each key frame is created, structured together with the key frame, and stored in a database, and then the search is performed using the caption information. The relationship between the index structure and caption for MPEG-2 video is shown in Fig. 1.

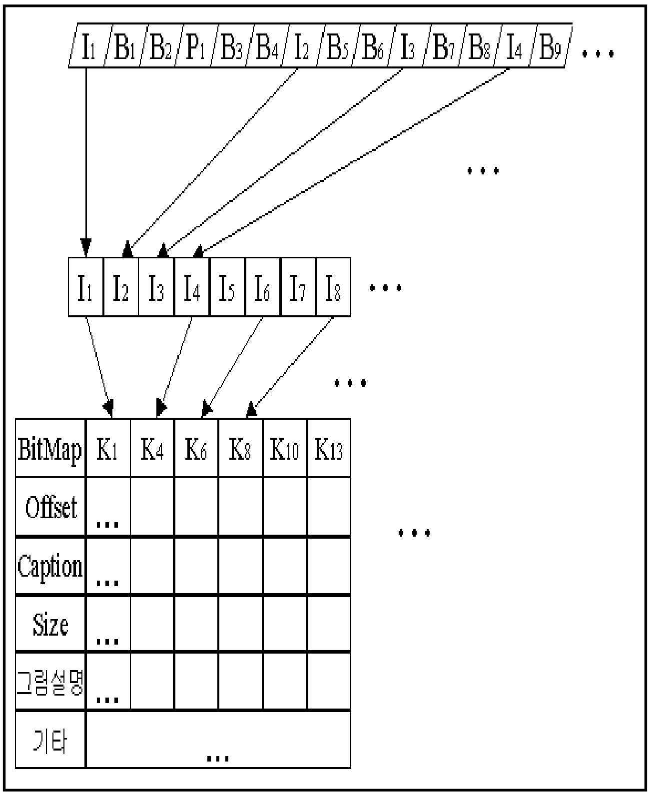

Fig. 1. Index Structure of MPEG-2

CVIMS is a caption and picture description editor that can index user interfaces and MPEG-2 videos, a query processor that handles various user queries, a video display that displays query results, and a database that manages index data and video data. And it is composed of a storage server that stores MPEG-2 video [18]. CVIMS is largely composed of user interface, video processor, and management data manager. The video processor is again composed of a query processor, a caption/picture description editor, and a video display. The query processor is composed of a caption query machine, a picture description query machine, and a query machine that combines caption and picture description according to the query type. This receives the user's query and searches each object managed by the management data manager to bring the desired result. In the caption/picture description editor, a key frame is selected from a list of pre-decoded I-frames, and caption information and picture description information for each key frame are created and stored in the database. The video display is a part that displays the result of a query, and is classified into a thumbnail display that outputs as an icon in the form of a thumbnail picture, and an image display that displays an actual video. Lastly, the management data manager manages the information stored in the database, and manages various index information and caption/picture description information created by the editor [19].

CVIMS is built as a client/server environment, and the server extracts I-frames from MPEG-2 video, selects key frames, and manages various information about key frames by structuring them in a database. For the interface with the user, the client is composed of a module that edits the user's request, that is, caption and picture description information for a key frame, a module that processes user queries, and a display module that shows the result. The connection between the client and the server was made by using the ODBC driver of VC++, and MPEG-2 files are provided from the storage server through the ATM network. Fig. 2 shows the client/server

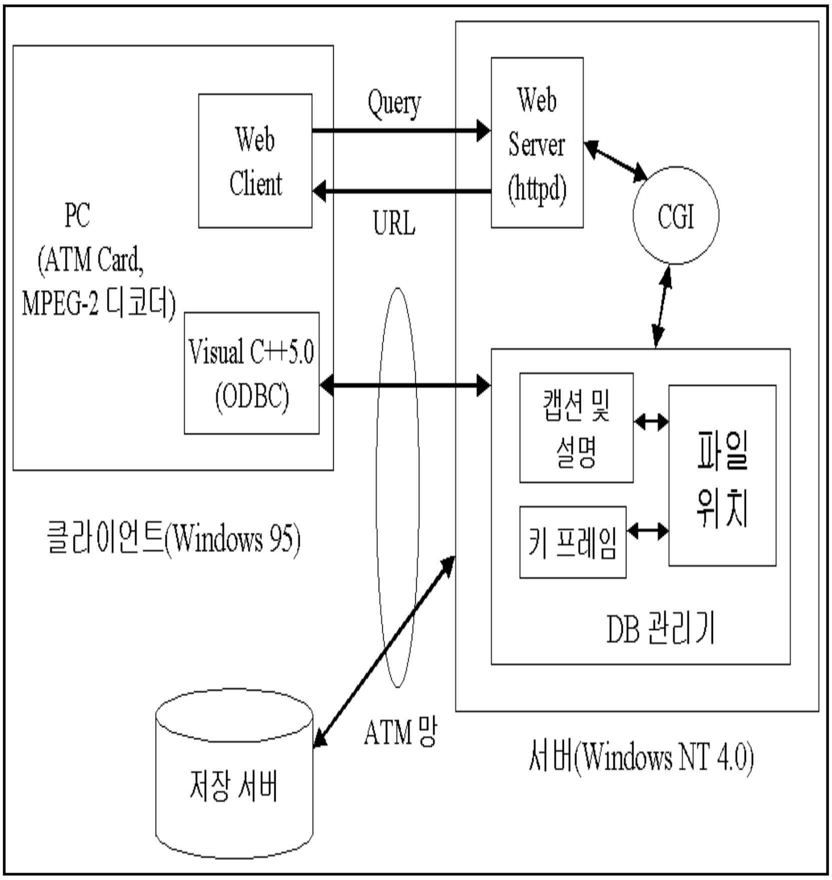

Fig.2. CVIMS Client/Server Environment

structure diagram of CVIMS.

V. Implementation Result of CVIMS

Chapter 5 shows the implementation results of the CVIMS designed in Chapter 4, centering on the user interface screen. Fig. 3 shows the screen when the user executes CVIMS [20]. The composition of the screen is largely divided into three categories: an editing window for editing, a search window for querying, and a display window for showing videos.

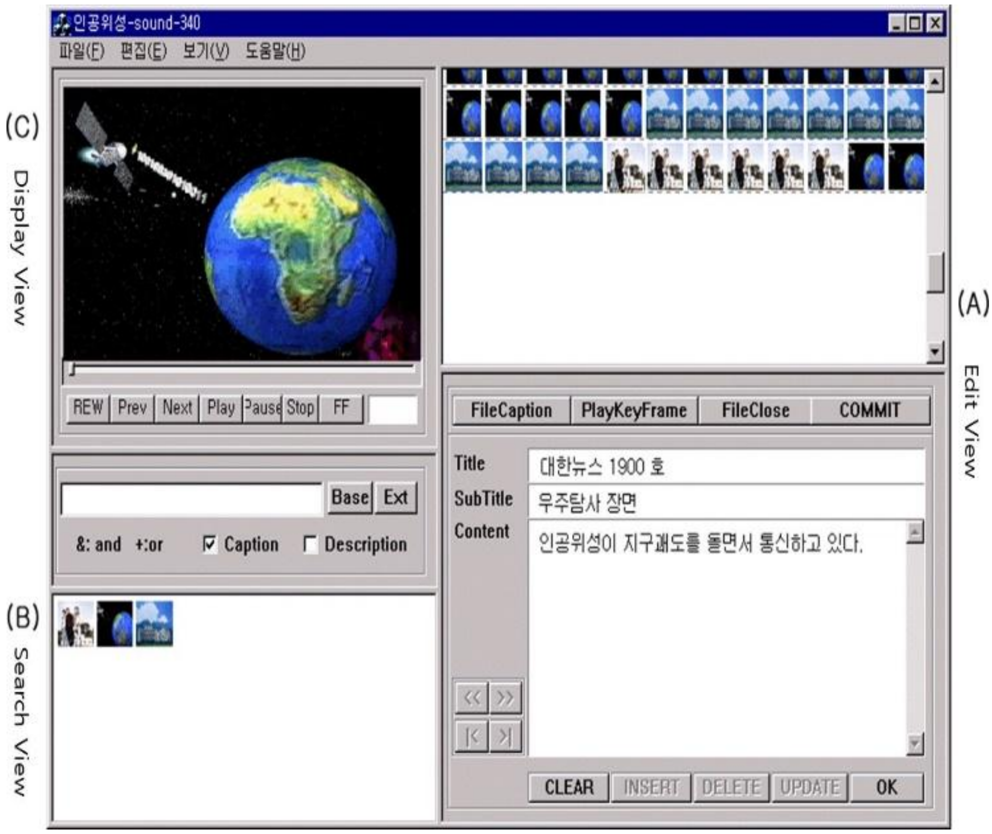

Fig.3. User Interface Screen

The editing window is shown in Fig. It is shown in 3(A). Each of them is described in terms of function. First, the editor loads the MPEG-2 video file to be edited in File Open. Then, information about the I-frame is displayed in the I-frame list window as a bitmap image in the form of a thumbnail [21]. The editor sets the I-frames that can be key frames in this I-frame table and their scope with the mouse. Then, if you click the play button on the screen, the video for the selected area is displayed on the video display screen, and the editor edits the caption information and picture description information for the key frame selected in the caption edit window and picture description edit window while viewing this video. Fig. 3(B) is a window for the search subsystem. The function of the search sub-system is largely divided into basic search and extended search, and the basic search is a search for general users to find desired information using caption and picture description information [22]. Here, and operation and or operation are supported, and the check box is used to determine whether to search with caption information or picture description information. 3(C) shows the video display window. The information found in the search is displayed as a bitmap image in the form of a thumbnail, and when a user clicks a specific bitmap here, the actual video represented by this bitmap is displayed on the display window.

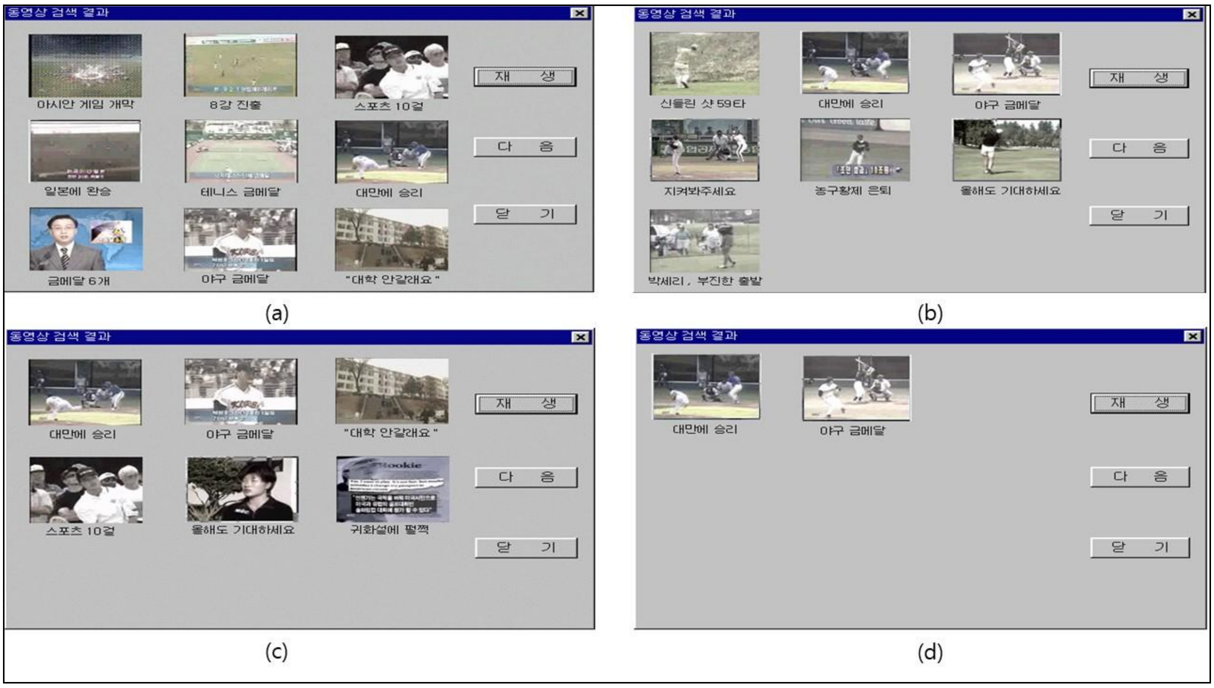

Fig.4. Query Result of Video Information

Fig 4. shows the query result interface of video information. (a) shows all sports events. (b) shows the results related to golf and baseball. (c) shows topic news that occurred in October. (d) shows only pure baseball results.

VI. Conclusion & Future Works

In this paper, we implemented EGVDM (Enhanced Generic Video Data Model) by extending the video data model, and based on this, we designed and implemented a prototype of a compressed video information management system (CVIMS) that can manage MPEG-2 compressed video. . And based on this model, we designed an object-oriented database schema using news videos as an example.

As a future task, not only index information but also video data itself should be structured and stored in a database. In addition, it is necessary to make an effort to standardize the category in the keyframe and the type in the SI_frame among sub-scene structures for content-based search.

REFERENCES

1. A. Ono, M. Amano, M. Hakaridani, T. Satou, and M. Sakauchi, "A Flexible Content-Based Image Retrieval System with Combined Scene Description Keyword," IEEE, pp. 201-208, June 2019.
2. B. Bruegge, J. Blythe, J. Jackson and J. Shufelt, " Object-Oriented System Modeling with OMT," OOPSLA '92, 27(10), PP. 414-427, Oct. 2019.
3. D. Gall, "MPEG: A Video Compression Standard for Multimedia Applications," CACM, 34(4), pp. 47-58, April 2019.
4. E. Oomoto and K. Tanaka, "OVID: Design and Implementation of a Video-Object Database System," IEEE Trans. on Knowledge and Data Engineering, 5(4), pp. 62-72, 2019.
5. H. Aoki, S. Shimotsuji, and O. Hori, "A Shot Classification Method of Selecting Effective Key-Frames for Video Browsing," ACM Multimedia 96, pp. 1-10, 2019.
6. H. Frater and D. Paulissen, Multimedia Mania, Abacus, 2019.
7. I. Center, "Query by Image and Video Content: The QBIC System," IEEE Computer, 28(9), pp. 23-32, Sept. 2019.
8. J. Monaco, "How to Read a Film," The Art, Technology, Language, History and Theory of Film and Media, Oxford University Press, 2019.
9. J. Meng and S. Chang, "CVEPS - A Compressed Video Editing and Parsing System," ACM Multimedia 96, pp. 43-53, 2020.
10. J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen, Object-Oriented Modeling and Design, Prentice-Hall, 2020.

11. J. Smith and S. Chang, "Searching for Images and Videos on the World-Wide Web," Technical Report #459-96-25, Center for Telecommunications Research, Columbia University, New York, August 2020.
12. J. Smith and S. Chang, "VisaulSEEK: a Fully Automated Content-based Image Query System," ACM Multimedia '96, Nov. 2020.
13. K. Hirata and T. Kato, "Query by Visual Example Content-based Image Retrieval," Advances in Database Technology(EDBT '92), pp. 56-71, 2020.
14. M. Davis, "Media Streams: Representing Video for Retrieval and Repurposing," Ph. D. Thesis, Massachusetts Institute of Technology, 2020.
15. R. and J. Gray, "Similar-shape Retrieval in Shape Data Management," IEEE Computer, pp. 57-62, Sept. 2020.
16. R. Hjelsvold, "Video Information Contents and Architecture," In Proceedings of the 4th International Conference on Extending Database Technology, pp. 28-31, March 2020.
17. R. Hjelsvold and R. Midtstraum, "Modelling and Querying Video Data," Proceedings of the 20th VLDB Conference, 2020.
18. [18] S. Smoliar and H. Zhang, "Content-Based Video Indexing and Retrieval," IEEE MultiMedia, pp. 62-72 2020.
19. S. Stevens, "Next Generation Network and Operating System Requirements for Continuous Time Media," In Proceedings of the Second International Workshop for Network and Operating System Support for Digital Audio and Video, November 2020.
20. T. Smith, "If You Could See What I Mean... Descriptions of Video in An Anthropologist's Notebook," Master's Thesis, MIT, 2020.
21. V. Kobla and D. Doermann, "Compressed Domain Video Indexing Techniques Using DCT and Motion Vector Information in MPEG Video," SPIE, Vol. 3022, pp. 200-211, 2020.
22. V. Ogle and M. Stonebraker, "Chabot: Retrieval From a Relational Database of Images," IEEE Computer, 28(9), pp. 40-48, 2020.