Toward The MEdiaSys VIdeo Search Engine (MEVISE)

Frédéric Andrès, Kinji Ono, Shin'Ichi Satoh

NACSIS, R&D Department, Tokyo, Japan

{andres,ono,satoh}@rd.nacsis.ac.jp

Nicolas Dessaigne

NACSIS, R&D Department, Tokyo, Japan dessaigne @mcsun4.rd.nacsis.ac.jp

Multimedia searching over Internet has gained substantial popularity Abstract in the past two years, pushing the emergence of multimedia portals. The key issues are how to do access to multimedia data and so how to represent multimedia data to ease and to speed up such accesses. Access ways are evolving from keywords and related metadata to more intelligent representations (shadow semantics such information relationships inside the video itself). Furthermore, technologies such as Java, along with the growing number of Web browsers that can execute Java applets facilitate distributed multimedia accesses. Networking and computational performances are key concerns when considering the use of Java to develop performance-sensitive distributed multimedia search engines. This paper makes three contributions to the study of such performance-sensitive distributed multimedia search engines. First, we describe the innovative architecture of MEVISE, the MediaSys VIdeo Search Engine over a large scale network. Second, we introduce the content-based representation of the video data inside the MediaSys server. Third, we present the search capabilities of MEVISE.

Keywords: Video Search Engine, Information Retrieval, Multimedia Database

1. INTRODUCTION

Lot of works in the field of multimedia management systems have followed generic approaches. Furthermore, Visualisation tools have been developed such as video search engines by content (Ram 1999), and

32 VISUAL DATABASE SYSTEMS

especially for the manipulation and the storage of video features (Flickner *et al.* 1995) such as the colours, the edges, or the shape.

Also, the design of data structure has been studied for modelling (Frakes & Baeza-Yates 1992, Ghandeharizadeh 1996) representing and storing multimedia video content [Ozden 1996] as well as for accessing efficiently to video data [Guttman 1984, Sellis *et al.* 1987, Beckmann *et al.* 1990, Lin *et al.* 1994, Ahanger & Little 1996, Bertino 1997).

However, problems of large scale distributed multimedia information systems such as the quality of the search, the performance, the flexibility, and the customisability have been mostly ignored. MediaSys provides such an extensible infrastructure for multimedia management. For instance, plugins for video search operations can be dynamically associated and integrated. This eases its upgrades according to the user's requirements and according to the evolving technology.

The remainder of this paper is organised as follows: Section 2 motivates and describes the architecture of our multimedia distributed system. Then, Section 3 describes more particularly the MEVISE component, i. e., the client part of the architecture. Next, Section 4 introduces MediaSys, the server and the content-based representation for video storage management. Finally, we give some concluding remarks.

2. ARCHITECTURE OF THE DISTRIBUTED MULTIMEDIA SYSTEM

2.1 MOTIVATIONS AND GENERAL ARCHITECTURE

Video retrieval plays a key role in cultural, medical, or environmental applications. The demand for systems that support video searching, visualisation, analysis and processing has increased significantly, as reported by the Gardner Group. This increase is due to the advent of the digital world and the generalisation of direct production of video. Also, due to Internet, the distribution aspect through portal is more and more important.

Figure 1 shows the general architecture of a large-scale distributed multimedia system. In this type of environment, various devices produce video data that are transferred to the multimedia storage systems, i. e., MediaSys servers. Users with various roles and profiles can search for and accesses to such video data using MEVISE clients. They can visualise them and/or analyse them according to some semantics.

The need for video search engines over distributed environment is also driven by economic factors. Actors and holders of video data such as museums, news broadcasting companies require search engines to unify the search over their multiple and distinct video repositories. Furthermore, they are required to provide a quality of service (QoS) in terms of high performance and functionality. High-speed networks, such as ATM or Fast Ethernet, allow the transfer of video efficiently, reliably, and economically.

Motivations and General Architecture Motivations and General Architecture



Figure 1 Architecture of the Multimedia Distributed System

2.2 REQUIREMENTS OF A VIDEO SEARCH ENGINE OVER A LARGE SCALE DISTRIBUTED ENVIRONMENT

A typical Video search engine over a distributed environment (Mungee 1998, Coulson 1999) must be:

- *Extensible* to manipulate new video format and to adapt to various data models;

- *Efficient* to process video data in order to get their semantic and to deliver it to the users;

- *Scalable* to support the growing demands of video searching over large-scale systems;

- *Flexible* to dynamically reconfigure video processing features to cope with changing requirements;

- *Reliable* to ensure QoS in terms of correctness and availability of video data when requested by users;

34 VISUAL DATABASE SYSTEMS

- *Cost-effective* to minimize the overhead of accessing video across large scale distributed systems;

- Secure to ensure that data confidentiality is preserved according to user's role.

Developing a distributed video search engine that meets all of these requirements is challenging, particularly because some features conflict with others. For instance, efficiency often requires high-performance computers and high-speed networks, thereby raising costs as the number of users increases.

2.3 ADOPTING WEB AND JAVA TECHNOLOGIES

All The strong interest in the Java language has followed the ubiquity of inexpensive Web browsers. These have brought the Web technology to the desktop of any computer. Over the past two years, the Web technology enables the Java programming language to spark considerable interest among software developers. Its popularity stems from its flexibility, portability, and relative simplicity compared to other object-oriented programming languages (Jain & Schmidt 1997).

The *applet* feature supported by Java is particularly relevant to distributed multimedia system. An applet is a Java class that can be downloaded from a Web server and run in a context application such as a Web browser or an applet viewer. The ability to download Java classes across a network can simplify the development and configuration of efficient and reliable distributed multimedia search engines (Schmidt 1997).

The MEVISE prototype has been developed as a Java applet. Therefore, it can run on any Java-enabled browser that supports the standard Java windowing toolkit. MEVISE leverages the convenience of Java to manipulate video and provides video-processing capabilities to users connected via the Web. In our experience, developing a video search engine over distributed environment in Java is relatively cost effective.Performance is a key requirement in large-scale distributed multimedia systems, regardless of any particular application domain. Meeting the performance demands requires the following support from MEVISE. First, its video searching and processing capabilities must be precise and efficient. Secondly, its networking mechanisms must download, and upload large video rapidly. Assuming that efficient video processing algorithms are used to extract semantic characteristics, the performance of a MEVISE applet depends largely on the efficiency of the hardware and of the JVM (*Java Virtual Machine*) implementation on which the applet is running.

2.4 DESIGN OF THE MEVISE PLATFORM

Figure 2 shows the detailed architecture of the MEVISE platform. The two major components of the initial version are the MEVISE client and the MediaSys server. MEVISE is an extension of the Name-It prototype (Satoh 1997). The MediaSys server is based on the AHYDS platform (*Active HYpermedia Delivery System*) (Andrès & Ono 1998), which is an active hypermedia delivery system developed at NACSIS since 1995. Furthermore, the MEVISE tool uses the resource manager JACE [URL1] developed at Washington University, Saint Louis, Missouri. Finally, it also follows the Active Object approach (Lavender & Schmidt 1996).

The MEVISE Client and The MediaSys Server are discussed in the following two sections.



Figure 2 The MEVISE platform

3. THE CLIENT MEVISE APPLET

In this section, we overview the major features of MEVISE and its Search Component. Furthermore, we sketch the use of fuzzy technology for advanced video retrieval by content.

3.1 THE KEY FEATURES OF MEVISE

MEVISE allows users to search for names or faces inside video shot information system and also allows to access to any video stored in a MediaSys server. In addition, the MEVISE applet provides a hierarchical browser that allows users to traverse sets of video on remote MediaSys servers. This makes it straightforward to find and select video across the network, making MEVISE quite usable, as well as easy to learn.

As shown in Figure 2, there are three main components in a MEVISE client applet:

- *Graphical User Interface:* which provides a front-end to the video search engine. It enables the users to search video according to some criteria, and to receive them. Figure 3 illustrates the MEVISE Graphical User Interface (GUI.)

- Server Locator: which locates an URL address associated to the user role that can reference a video-server or an image server (MediaSys-typed server). If the URL points to a MediaSys server, its content is listed so any user can browse them to choose one video to retrieve. The following video Downloader module uses this component.

- *Video Downloader:* which downloads a video or a set of video frames located by the Server Locator and displays the video or the set of video frames in the applet. The Video Downloader also ensures that all frames are retrieved and displayed properly.

3.2 The MEVISE SEARCH COMPONENT

The input to the Search Component consists of a set of sample captured images. First, the Search Component enables the user to retrieve similar faces from a set of video frames according to a given name: it is the Face Similarity Retrieval Function. The result is an ordered list of captured images of decreasing similarity with respect to a ranking weight. Secondly, it also enables to retrieve associated names according to a selected frame: it is the Associated Name Retrieval Function. The result is an ordered list of names of decreasing ranking of relevance. Third, the Search Component can also retrieve video shots related to a selected face in order to play it.

Search component for Names according to a given Face

The Search Process is based on three steps:

(1) Find a similar face given a input face

(2) Find coincident Names according to the score

(3) Evaluate the co-occurrence taking face similarities & name scores into account.



Search Component for Faces according to a given Name

The Search Process is based on 4 steps:

- (1) Input a name
- (2) Navigate in the links between Names and score and Locate the occurrence of the input name in the set of frames
- (3) Find coincident faces
- (4) Calculate co-occurrence using face similarities and name scores based on the SR-tree indexes.

4. THE MEDIASYS SERVER

4.1 GLOBAL DESCRIPTION

At the other end of the network, the MediaSys server is a highperformance, multi-threaded, HTTP application-oriented information engine based on the AHYDS (*Active Hypermedia Delivery System*) platform. A detailed description has been done previously (Andrès & Ono 1998). It stores the MEVISE applet, video data and captured images. Figure 3

38 VISUAL DATABASE SYSTEMS

illustrates the interaction between the MediaSys server and the MEVISE applet concerning video/captured image transfers. The MEVISE applet uses the HTTP GET method to interact with the MediaSys server to request video, names or captured images (1) and (2) in Figure 4. The MEVISE applet also accesses to the MediaSys server to download specific video or image retrieval operations. Each image or video filter is a Java class that can be downloaded by MEVISE. This design allows MEVISE applets to be dynamically configured with video retrieval plug-ins. In addition, the MediaSys server supports captured image uploading using the HTTP PUT method. This allows the MISE applet to save processed images persistently at the server.



Figure 3 Interaction between the MEVISE Client and the MediaSys Server.

Java applets provide an exception to these security restrictions. In particular, the Java applet class provides a method that allows an applet to download images from any server reachable via an URL. Since the method

Toward The MEdiaSys VIdeo Search Engine (MEVISE) 39

is defined in the Java applet class, it allows Java to ensure that there is no security violation. MediaSys uses this method to download video and captured images across the network. Therefore, video and captured images to be processed can reside in the MediaSys server from where the MEVISE client applet was downloaded. They also can reside on some other MediaSys servers in the network. However, videos/captured images data flow can only be used to upload images to the server from where the MEVISE applet was downloaded.



Figure 4 Video/Captured Image Data Flow Management

4.2 Video Storage

4.2.1 Video Abstraction Processing based on Name-It.

Video abstraction done by Name-It (Satoh 1997) is based on the extraction of content information by combining image understanding processing and natural language processing. Name-It automatically extracts face and name associations as content information from video news or video movies as input source. To accomplish this task, the system takes a multi-modal video analysis approach: (1) face sequence extraction/identification from videos, (2) name extraction from transcripts, and (3) video caption recognition. Each method includes several advanced image and natural language processing techniques: face tracking, face identification, intelligent name extraction using dictionary, thesaurus, and parser, text region detection,

image enhancement, character recognition, and the integration of these techniques.

4.2.2 Content-based representation Storage

The content-based representation includes three sets of data: the face information (FaceMinPosition, FaceMaxPosition, and Frame), the Face name information (Name) and the relationship with Video Shots. All those informations are stored inside the Extended Binary Graph (EBG) data structure (Andres and Ono 97) to provide high performance access for the retrieval process. The Major benefits of the graph based approach are: (1) a pointer-based structure enabling both navigation and selection; (2) Compact Relationships between Ids and Values (e.g. Name, Video shots); and (3) Access from IDs to values and reverse access from values to ID.

Figure 5 summarises the content-based representation of the video semantic used by MEVISE. FaceMinPosition and FaceMaxPosition are used to locate



Figure 5 EBG support for Video Abstraction

faces inside a specific frame. The Name-FrameId relationship and videoShots-FrameId relationship enable to traverse the data structure graph according to the user's interests. Each information (Name, Frame,..) can be clustered according the application's requirement. Furthermore, the resource

Toward The MEdiaSys VIdeo Search Engine (MEVISE) 41

supervisor of the MediaSys platform can process each attribute of the video abstraction accordingly to the characteristics of the application's access.

The MediaSys server also provides a nearest neighbor search capabilities. MediaSys server supports a main memory oriented SR-Tree structure (Katayama 1997) stored as EBG in order to index the large number of face positions inside frames where disk accesses dominate. In this case, the MediaSys server uses a clustering operation where the SR-Tree is reordered such as each memory page contains one subtree. Figure 6a and 6b overviews the SRTree structure and the clustering approach at the node level. Each face location is mapped into a multi-dimensional space where the distance between two faces corresponds to the similarity between the two faces.



Figure 6a EBG support for Video Abstraction



Figure 6b EBG support for Video Abstraction

Each sub-region in memory is determined by the intersection between the bounding sphere and the bounding rectangle as it is described at the node level in Figure 6b

5. CONCLUSION

This paper described the design, and the architecture of the MEdiaSys VIdeo Search Engine (MEVISE), an advanced video search system over Internet and/or high-speed network environments. This system enables the users to search, to browse, and to retrieve video and captured images according to the combination of visual and name features with meta-data related to the videos. The MediaSys servers store the meta-data, visual and textual features, and both video and captured images over a large-scale distributed and heterogeneous system.

Java has still limitations that required to be fixed to improve future versions of MEVISE:

- 1- Memory limitation (conflict between large images and browser).
- 2- An improvement in terms of portability of Java Swing in the navigators.
- 3- A modification of the security inside Java to configure dynamically the accessible servers.

As conclusion, MEVISE, the client video search engine of the MediaSys project, gains its efficiency by the use of Java. The Java approach enables to build valuable tool, as MEVISE is a simple, portable, and distributed system.

In the on-going research work, the MediaSys VIdeo Search Engine will be extended with flexible fuzzy search algorithms with new video features such as key word, and multi-resolution. Performance evaluation will assess Toward The MEdiaSys VIdeo Search Engine (MEVISE) 43

the quality of our approach and the architecture of MEVISE search algorithm.

REFERENCES

- Ahanger, G., Little, T. D. C. (1996) A Survey of Technologies for Parsing and Indexing Digital Video; Journal of Visual Communication and Image Representation, (Special Issue on Digital Libraries), March 1996, Vol. 7, No. 1, pp. 28-43
- Andrès, F., Ono, K. (1997) Phasme: Un Système Parallele de Gestion de Bases de Données Oriente Application, Calculateurs Paralleles, Special Issue on Parallel and Distributed Database Systems, 1997.
- Andrès, F., Ono, K. (1998) Active Hypermedia Delivery System; Proc. of the Int'l Conf. On Data Engineering (ICDE'98), Florida, February 1998, pp. 600.
- Bertino E. et al. (1997) Indexing Techniques for Advanced Database Systems; Kluwer Academic Publishers, Inc. San Francisco, California, 1997, 250p
- Beckmann, N., Kriegel, H. P., Schneider, R., Seeger, B. (1990) The R*-Tree: an Efficient and Robust Access Method for Points and Rectangles; Proc. of ACM SIG Int'l Conf. on Management of Data (SIGMOD'90), Atlantic City, NJ, May 1990, pp. 322-331
- Coulson G. (1999) A Configurable Multimedia Middleware Platform, IEEE Multimedia, January-March 1999, pp. 62-76.
- Flickner, M., Sawhnery, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D., Yanker, P. (1995) *Query by Image and Video Content*: The QBIC System; IEEE Computer, September 1995, pp. 23-32
- Frakes, W. B., Baeza-Yates, R. (1992) Information Retrieval: Data Structures & Algorithms; Prentice-Hall, 1992, 504 p.
- Ghandeharizadeh S. (1996) Stream-based Versus Structured Video Objects: Issues, Solutions, and Challenges; in Multimedia Database Systems, Springer-verlag, 1996, p 215-236
- Guttman, A. (1984) *R-Trees: A Dynamic Index for Geometric Data*; Proc. of ACM SIG Int'l Conf. on Management of Data (SIGMOD'84), 1984, pp. 47-57
- Katayama N., Satoh S., (1997) The SR-tree: An Index Structure for High-Dimensional Nearest Neighbor Queries; Proc. of the 1997 ACM SIGMOD International Conference on Management of Data (May 1997) pp. 369-380
- Jain, P., Schmidt, D. C. (1997) Experiences Converting a C Communication Software Framework to Java; C Report, Vol. 9, January 1997
- Lavender, R. G., Schmidt, D. C. (1996) Active Object: an Object Behavioral Pattern for Concurrent Programming; in Pattern Languages of Program Design, Reading, MA: Addison-Wesley, 1996.
- Lin, K. I., Jagadish H. V., Faloutsos, C. (1994) The TV-Tree An Index Structure for High Dimentional Data; The VLDB Journal, 3(4), October 1994, pp. 517-542
- Mungee S., Surendran N., and Schmidt D. (1998) The Design and Implementation of a Corba Audio/Video Streaming Service, Washington Univ. Tech. Report, WUCS-98-15, Dept of Computer Science, Washington University at St Louis, Mo., May 1998.
- Ozden B., Rastogi R., and Silberschatz, A. (1996) *The Storage and Retrieval of Continuous Media Data;* in Multimedia Database Systems, Springer Verlag, 1996, p 236-261
- Satoh S., Kanade T. (1997) Name-It: Association of Face and Name in Video, Proc. of CVPR'97, 1997, pp. 368-373
- Satoh S., Nakamura Y., and Kanade T. (1999) *Name-it: Naming and Detecting Faces in News Videos*, IEEE Multimedia, January-March 1999, pp. 22-48.
- Sellis, T., Roussopoulos, N., Faloutsos, C. (1987) The R+-Tree: A Dynamic Index of Multidimensional Objects; Proc. of the 18th Int'l Conf. on Very Large Data Bases (VLDB'87), 1987

BIOGRAPHIES

Frederic Andres is a visiting researcher, AHYDS project leader at National Center for Science Information Systems (NACSIS), Japan since 1996 under the COE framework of the Japanese ministry of Education. He received his PhD and HDR degree in 1993 and 2000 respectively from the University of Paris VI and University of Nantes. His research interests include distributed and heterogeneous multimedia information systems, and multimedia retrieval. He was a scientist at Bull from 1989 to 1993, and system architect at Ifatec/Euriware from 1993 to 1996.

Kinjo Ono received a B.Sc. in physics from the University of Tokyo in 1962, an M.S. in Electrical Engineering from Stanford University in 1972, and a Dr. of Eng. from the University of Tokyo in 1983. He joined KDD in 1962 where he was engaged in the research on satellite communications and computer communications. After serving as the Director of Research Laboratories he was assigned as Professor at NACSIS in 1993. His current research interests include high-speed networking and distributed multimedia. He was awarded the Prize of the Minister of the STA, the Achievement Prize of IEICE. He is a Technical Program Chairman of ICCC'99, member of IEICE and Fellow of IEEE.

Shin'ichi Satoh is an associate professor at National Center for Science Information Systems (NACSIS), Japan. He received his BE degree in 1987, his ME and PhD degrees in 1989 and 1992 from the University of Tokyo. His research interests include video analysis and multimedia database. He was a visiting scientist at the Robotics Institute, Carnegie Mellon University, from 1995 to 1997.

Nicolas Dessaigne was graduate from Ecole Superieure d'Informatique-Electronique-Automatique (ESIEA, France) at the beginning of 2000. He is currently a research engineer in the NAtional Center for Science Information Systems (NACSIS, Tokyo). He has worked since July 1999 on the MediaSys project.