MediaManager: A Distributed Multimedia Management System for Content-Based Retrieval, Authoring and Presentation

Shu-Ching Chen[1], Mei-Ling Shyu[2], Na Zhao[1]
[1]Distributed Multimedia Information System Laboratory
School of Computer Science, Florida International University, Miami, FL 33199, USA
[2]Department of Electrical & Computer Engineering
University of Miami, Coral Gables, FL 33124, USA
[1]{chens, nzhao002}@cs.fiu.edu, [2]shyu@miami.edu

ABSTRACT

A distributed multimedia management system named MediaManager is presented in this paper. Unlike many existing multimedia systems, our MediaManager supports content-based retrieval, multimedia authoring, and multimedia presentation together in a single framework. There are also three main characteristics in MediaManager. Firstly, it provides friendly icon-based query interfaces for the users to browse and retrieve multimedia data from the database. Secondly, the multimedia augmented transition network (MATN) model is implemented to effectively and easily model the varieties of multimedia data in terms of their structures, behaviors and functions, and to support multimedia presentations. Thirdly, it provides the capability to allow the users to watch the designed multimedia presentation via both Java Media Framework and the web browser.

1. INTRODUCTION

With the rapid development of technologies and applications for consumer digital media, there has been an explosion in the complexity and amount of digital multimedia data being generated and thus persistently stored. In comparison to traditional text and data, multimedia objects are typically very large in size and may include images, video, audio and some other visualization components. In order to organize and access the large amount of multimedia data, many subsequent research issues arise within the fields of multimedia analysis, storage, retrieval, transmission, and presentation.

First of all, efficient content-based retrieval functionality is necessary for a multimedia database management system (MDBMS). In addition to the retrieval/querying capability, the development of an abstract semantic model is also essential for a robust MDBMS. The model should be powerful enough to support multimedia presentation synchronization and utilize efficient programming data structures for the implementation. One such model is the multimedia augmented transition network (MATN) model, which can model the temporal and/or spatial relations of different media streams in the multimedia presentations, and offer multimedia browsing capabilities [2][3] .

In this paper, we present a multi-threaded client/server architecture that runs on Unix, Windows, and Linux platforms, and is developed using Java and C++. In this distributed multimedia management system, a database engine supports content-based image and key frame video queries, feature extraction and indexing of new images/key frames, storage and retrieval of the MATN models, and file supply for the created multimedia presentations. The client application utilizes several content-based retrieval user interfaces which allow the browsing of the contents of various domains (e.g., sports, hurricane, medical, etc.) from its respective data stored within the database. An MATN-based multimedia presentation authoring environment allows the user to interactively create, edit, and store multimedia presentations from media content within the database, as well as allows the user to select and view any portion of a multimedia presentation obtained from its authoring environment.

In order to support content-based retrieval, the image data management structure and the video data management structure have been developed into the system. Moreover, in addition to the multimedia data, the multimedia metadata such as the features for the images and the key frame information of the video, are also managed from within. Finally, our system allows the users to view the designed multimedia presentation via the web browsers.

The rest of this paper is organized as follows. First, the next section offers brief descriptions of selected related work. Section 3 describes the implemented techniques, as well as demonstrates the interfaces of each module in the client side of MediaManager. This section will also describe the main components: the multimedia data retrieval methods, the MATN-based multimedia authoring environment, and then two presentation approaches. The multi-threaded client-server architecture is introduced in Section 4. The database server and the system integration are also briefly introduced in this section with our objective in showing how these modules can work together efficiently. Finally, the conclusions and the acknowledgments are given.

2. RELATED WORK

Technologies for multimedia data organization and retrieval have been applied successfully within many academic and commercial approaches, for example, PhotoBook [9], Fotofile [7], IBM's QBIC system [6], and Virage's VIR engine [13]. Different from the traditional keyword-based search technologies, the content-based indexing and retrieval approaches automatically extract features such as color, texture, and shape. It provides more powerful search abilities, so that people's perceptual abilities can be utilized, but these systems focus only on content-based image retrieval (CBIR).

Beyond such systems, some projects begin to offer video data solutions, for example, the project of multimedia analysis and retrieval system (MARS) [8], where the video representation is a vital segment of data. To structure a video into a set of scenes, the role of a table of content (ToC) is employed. VisualSEEk [10] combines two query methods so that the user can issue queries using visual features of both images and videos. Another interactive image/video retrieval system is VISMAP [5].

Another significant area of multimedia research is multimedia presentation model design. A multimedia presentation is a delivery medium of a collection of media streams which are constrained by temporal synchronization relationships among each other. There are a lot of conceptual models that have been developed over the past several years. These models can be summarized into different categories such as timeline based models, script based models, graph based models, and structure based models. For example, the multimedia information presentation system (MIPS) [11] allows the end-users to browse multimedia information presented in a user-friendly and consistent manner.

As we have reviewed above, most of the existing multimedia approaches deal primarily with one or a few problems in this field. However, it is more desirable to have a complete and integrated multimedia management system to provide all the possible functions so that the users can process and make use of the various formats of multimedia data in a more efficient and flexible manner. Our MediaManager is designed to meet this goal. First, it supports not only content-based image queries but also key frame video queries. Second, the multimedia augmented transition network (MATN) is implemented to model multimedia presentations and offer intuitive multimedia browsing [2][3]. We also utilize a popular script-based language, SMIL, to generate the on-line presentation. Hence, our MediaManager also supports multimedia authoring and multimedia presentations.

3. SYSTEM FUNCTIONALITIES

The client is broken down into three modules: the query module, the MATN design module, and the multimedia presentation module.

3.1 Query Module

Multimedia data can generally be categorized based on their subject domains. Users can name or select their preferred domain when uploading, managing, or retrieving the data. In addition, MediaManager utilizes different storage strategies to manage the four media types: text, image, audio, and video in the multimedia data repository.

The query module serves as the front-end of the content-based retrieval user interface which supports five types of queries: directory queries, file queries, media duration queries, content-based image queries, and key-frame video queries. From the query interfaces, the users can browse the contents of a database domain or retrieve multimedia data that will be used in the MATN module to design a multimedia presentation.

3.1.1 General Queries

Because the multimedia data are all categorized and stored under a variety of domains with their associated directories, the "*Browse Server*" function allows the users to view the database domains and their contents. The multimedia data can also be downloaded from their corresponding directories. The server will send back not only the selected data, but also some metadata that will be useful in the presentation such as the duration information of the media files (audio or video).

3.1.2 Content-based Image Queries

The objective of content-based image retrieval (CBIR) is to offer the user with an efficient way of finding and retrieving those images that are qualified for the matching criteria of the users' queries from the database. Our system supports this type of queries by providing the "*Query by image*" function to get all the similar images in the multimedia data repository.

There are three kinds of data items that are stored on the server-side, which consist of the original images, metadata for the images, and features of the images. To insert a new image into the database, first the user selects a domain and uploads the image into the database. Each image is assigned an image ID by the server. Then, we use MS Access to store the corresponding metadata such as domain name, image filename, file length, and the start position of the image data for an image. Lastly, the features of the image are obtained and stored in the database.

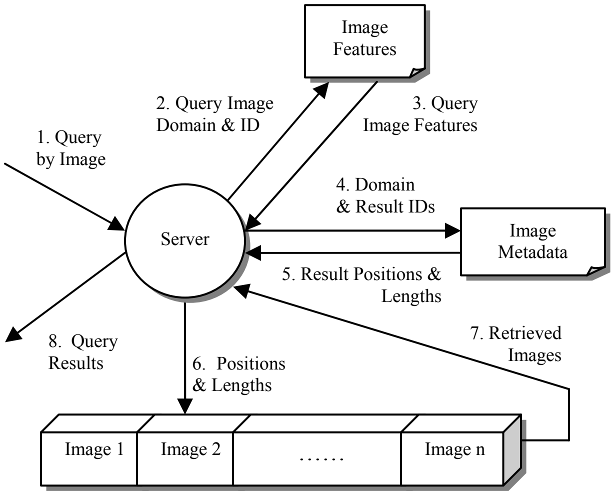

Figure 1. Image data management structure

Figure 1 presents the corresponding image data management structure. After the user clicks on the query image from the user interface, the domain and ID can be obtained and used to retrieve its features. Based on the information, the server runs the feature matching procedure and gets a list of image IDs that are most similar to the query image. The image IDs are then used to search the image metadata to get the start positions and the lengths of the image data. Finally, the corresponding images are retrieved and returned to the client.

(a) The bottom middle image is selected and outlined in red

(b) The nine most similar images are retrieved

Figure 2. Content-based image retrieval interface

The screen shots of two views of the query within the user interface are shown in Figure 2. The user selects a domain and nine images of that domain are displayed simultaneously. To issue a query-by-image query, the user selects the query image with the mouse and then presses the "*Query*" button. The selected query image is outlined in red (as shown in Figure 2(a)). After the retrieval process described in Figure 1, the nine most salient images are returned (with the original query image being on the upper left corner) as shown in Figure 2(b). The upload and download functions are also provided so that the users can manage the image database conveniently.

The preliminary image retrieval module was described in [4]. In this paper, the database module is redesigned and the corresponding functions are updated.

3.1.3 Key-frame Video Queries

A powerful search engine technique named "*Key-frame video query*" is supported. The key frames of the videos are generated and organized by the server. Users can select video data of the format ".avi", ".mpg", or ".mpeg", and upload them to the server. The server can automatically identify both shots and key-frame images, where each key frame corresponds to a shot file. For example, the Key Frame 1 represents Shot 1, which consists of those frames between Key Frame 1 and Key Frame 2 in Figure 3.

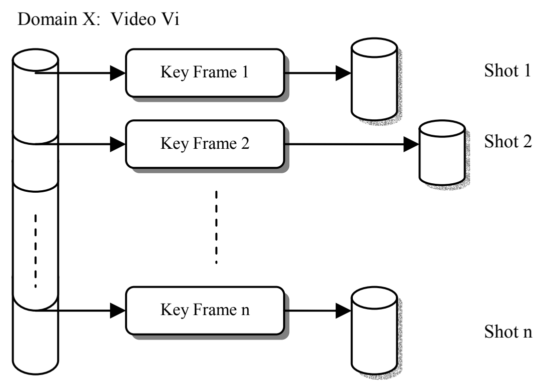

Figure 3. Video data management structure

Figure 4 shows the key frames of a video file with the name "roadkill_pigs.mpeg" [15]. As can be seen from this figure, the user can select his/her domains of interest, and the video files (with their file names) within the selected domain are displayed. Then, the user selects one video file name, and the key frames of this video file will be displayed. From the retrieved key frames, the user can easily know the contents of this video file without the need to preview the whole video. For example, the key frames in Figure 4 tell us that the video is a 3D animation video of a story with lovely pigs. Our system also allows the user to preview each shot by double-clicking the corresponding key frame. By using the video retrieval interface, the user can select and download not only the whole video, but also their favorite shots.

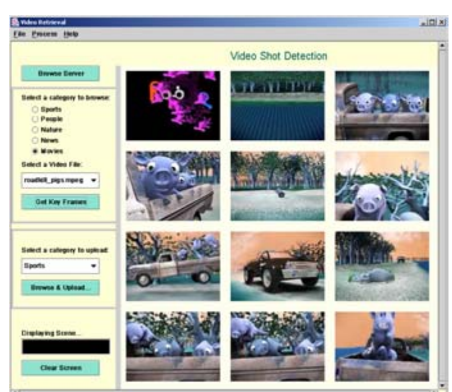

Figure 4. Key-frame video retrieval interface

3.2 MATN Design Module

The MATN model is capable of modeling the spatio-temporal relations among the multimedia objects. The details of the MATN model and multimedia input strings can be found in [2] [3]. An MATN is an 8-tuple: (Σ, Γ, Q, ψ, δ, T, S, F).

- Σ: The media stream and control command alphabet. Its elements include any character between "a"-"z", "A"-"Z", and "0"-"9". These characters are components of the input symbols.
- Γ: The special input symbol alphabet. $\Gamma = \{\&, +, *, -, (,), \$\}$. The alphabets of Γ are components of the multimedia input strings.
- $Q = \{Q_1, Q_2, \ldots, Q_n\}$: A finite set of states. Q_i means the i^{th} state of the MATN.
- $\psi = \{X_1, X_2, \ldots, X_m\}$: A finite set of input symbols. X_j is the input symbol on the j^{th} arc.
- δ: $Q \times \psi \rightarrow Q$. The transitions function from one state to another. An arc connects two states in an MATN. After finishing the execution of the current control commands or media stream displaying, it will move to a new state.
- S: The start state of an MATN, $S \in Q$.
- F: The set of final states of an MATN, $F \subset Q$.
- T: The condition/action table of an MATN, which is used to define conditions with the associated actions.

In the graphical representation of an MATN model, the states Q_is are represented by filled circles. The multimedia input strings X_i are put on the arcs and represented as arc labels. The multimedia input string indicates the combination of different media to be displayed during the presentation's duration. The user can select two states as the start state S and the final state F to generate the presentation. Then these two specific states are highlighted by the system.

In MediaManager, the designed MATN structure can be saved as a ".matn" file and stored in the server-side database. The pre-designed MATN model can be reused when designing the new MATN structure. User can use the "*Add Subnetwork*" function and then select a ".matn" file. The system will recognize it on an arc with label Pi where "i" is an integer number. When displaying this part of the new scenario, the system will open the corresponding ".matn" file, show its structure, and display the corresponding scenario. This function makes it possible to illustrate a complicated model without losing the integrated view. In addition, it definitely offers the great convenience to the user.

As shown in Figure 5, after the multimedia retrieval procedure, the selected data files will be organized and shown in the MATN design interface as a tree hierarchy. The MATN design interface provides a set of tools for the users to design their desired presentation structure more easily, which is a specific feature of the multimedia authoring environment. Models can be created, opened, or edited (by adding or deleting states/media, or editing the duration time of each media). Users can preview the videos or listen to the audios when they edit the MATN model. For example, a multimedia presentation that includes a hurricane might be needed. So, the users can use the "*preview*" function to select the related files. In the left-bottom corner of Figure 5, a selected hurricane video [12] is previewed.

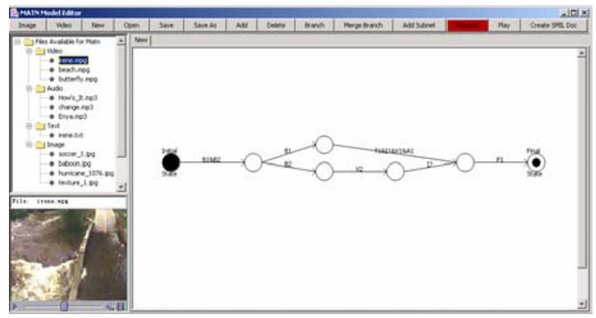

Figure 5. MATN design interface

Moreover, our MediaManager provides the function to convert a simple MATN structure (without user selection) to a SMIL (Synchronized Multimedia Integration Language) [14] file automatically. The technique used here will be introduced later.

3.3 Presentation Module

3.3.1 Presentation via Java Media Framework

By specifying a beginning state and an ending state from an MATN model, the users can select any part of the presentation to be displayed. Our MediaManager provides a convenient way to interact with the system because branches and subnetwork functions are supported.

Figure 6. Displaying the presentation via Java Media Framework

For the MATN model in Figure 5, after the user clicks the "*Play*" button, a window is created to display the presentation. In this MATN model, the branches are designed at the very beginning so that another window pops up to let the users select the branch they want to watch. Suppose the first branch is selected, then the corresponding four multimedia files on branch B1 will be displayed together (as shown in Figure 6). The arc label

P1 that follows the merged branches is a subnetwork. After the player finishes playing the four multimedia files on branch B1, the presentation window continues to display the contents in subnetwork P1. Moreover, the MATN model of subnetwork P1 will be displayed in the MATN design interface so that the users can preview the presentation structure easily.

3.3.2 Presentation via Web Browser

Based on XML, SMIL language [14] is powerful at creating synchronized multimedia presentations over low bandwidth connections. Developers can mix many types of media such as text, video, audio and images together, and synchronize them into a timeline. The MATN model can be used to model the conceptual structure of SMIL, as well as the temporal relations and synchronization control [1]. When the user designs an MATN model, he/she can select the files and the system then transmits the multimedia files to the client-side. Then the system can retrieve the files' duration time as well as their paths and file names, which will be used to fill in an HTML+SMIL template.

As shown in Figure 7, a presentation showing a scene at the FIU campus is displayed via the web browser. This presentation is designed in MediaManager in the following two steps. Firstly, the MATN model is used to build the presentation. Secondly, MediaManager uses the created MATN structure to generate an html file by adopting the SMIL language.

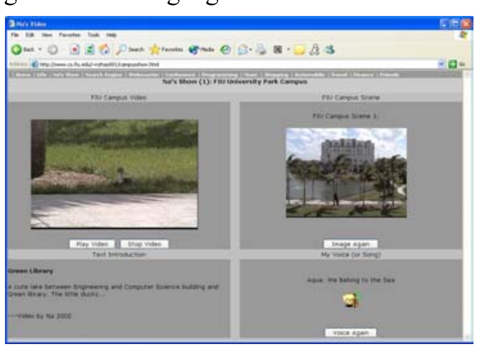

Figure 7. Displaying the presentation via a web browser

4 SYSTEM ARCHITECTURE

MediaManager adopts the multi-threaded client/server architecture. As shown in Figure 8, the query requests are initiated by the clients and sent to the database server. The server runs as a daemon process waiting for properly formatted request messages. When a request is received, a server thread is created to handle that request. The client and the server communicate using the UDP protocol. In the presentation authoring module, the complete contents of the media file are required for playback. In this way, the TCP communication will replace UDP for file transfers, and the RTP protocol will replace UDP for presentation playback.

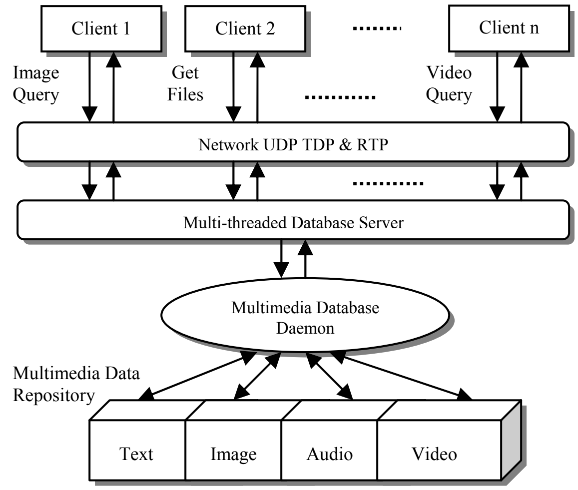

Figure 8. The client/server architecture

In our current system design, the client can issue five types of queries, namely directory queries, image queries, video queries, file queries, and media duration queries. Also, the multimedia data repository includes four media types: text, image, audio, and video, which use different storage strategies. In addition, the multimedia data are also categorized based on their subject domains. Users can name or select their preferred domain when uploading, managing or retrieving the multimedia data.

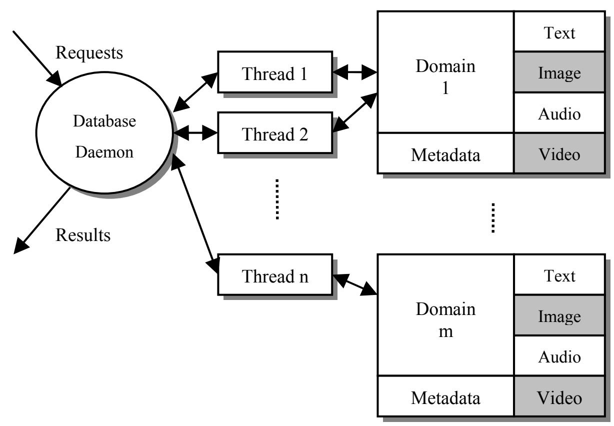

Figure 9. Multithreaded database engine architecture

4.1 Database Server

The task of a multimedia database server is to build up a system that can manage huge collections of multimedia data efficiently. Figure 9 depicts the overall request processing of the database server. The database is organized by the domains specified by the users to support multiple media types (text, video, audio and imagery) and operations upon those media. File retrieval and playback are also supported for all media types.

Currently, the server in our system supports the following operations: analyzing the multimedia data and retrieving the useful features; processing content-based retrieval for images and videos; listing the contents of a media type (text, audio, video, image) for a particular domain; transferring the file from a specified media type of a specified domain to the client; and returning the duration of the file.

4.2 The Client

The pipelining of MediaManager can be divided into three steps based on the diverse functionalities. As shown in Figure 10, multiple data query methods are supported by MediaManager. In the first step, the corresponding data and their features are retrieved as the presentation model construction material. Then the users use the graphical interface to design an MATN presentation model. When using Java Media Framework to display the media, the users can interact with the system to select their desired portion or branch of the whole presentation. In addition, the designed MATN structure can also be used to generate an SMIL document for the Web multimedia presentation.

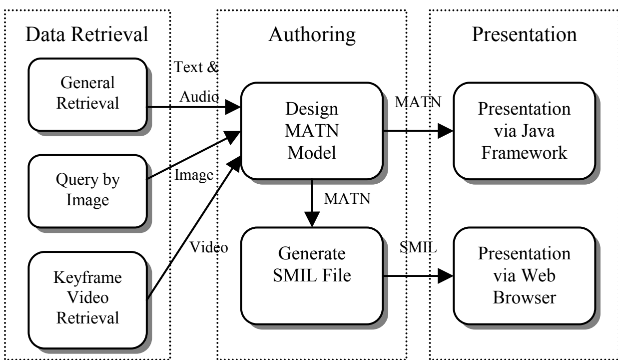

Figure 10. Integration of multimedia data retrieval, authoring, and presentation

5 CONCLUSIONS

The goal of our research is to design and develop an efficient multimedia data management infrastructure. In this paper, an innovative distributed multimedia management system named MediaManager is presented which provides an integrated framework to retrieve multimedia data, construct presentation models and demonstrate the presentation results. In MediaManager, the variety of multimedia data and their metadata are stored, managed, and retrieved efficiently. The MATN model and SMIL language are used in multimedia authoring. Finally, two choices are provided to the users to watch the designed multimedia presentations, either by Java Media Framework or via the web browser.

6 ACKNOWLEDGMENTS

This research was supported in part by NSF CDA-9711582, NSF EIA-0220562, and the office of the Provost/FIU Foundation. We would like to thank the following people for their efforts in the continuing development of this system: Jeff Strickrott, Xia Jin, Qiong Chen, Liping Li, Xiaoming Chen, and Ellen Luo.

7 REFERENCES

[1] S.-C. Chen, S.-T. Li, H.-C. Chen, M.-L. Shyu, and C. Zhang, "Streaming SMIL Presentations via a Multimedia Semantic Model," *Second International Workshop on Intelligent Multimedia Computing and Networking (IMMCN'2002)*, pp. 919-922, Durham, North Carolina, USA, March 8-12, 2002.

[2] S.-C. Chen and R.L. Kashyap, "A Spatio-Temporal Semantic Model for Multimedia Database Systems and Multimedia Information Systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, no. 4, pp. 607-622, July-August 2001.

[3] S.-C. Chen, R.L. Kashyap, and A. Ghafoor, *Semantic Models for Multimedia Database Searching and Browsing*, Kluwer, 2000.

[4] S.-C. Chen, M.-L. Shyu, X. Jin, Q. Chen, C. Zhang, and J. Strickrott, "A Flexible Image Retrieval and Multimedia Presentation Management System for Multimedia Databases," *ACM Multimedia 2001 conference*, pp. 601-602, Ottawa, CANADA, September 30 - October 5, 2001.

[5] W. Chen and S. F. Chang, "VISMAP: An Interactive Image/Video Retrieval System Using Visualization and Concept Maps," *In Proc. of Int. Conf. on Image Processing (ICIP)*, Greece, October 2001.

[6] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by image and video content: The QBIC system," *IEEE Computer*, Vol. 28, No. 9, pp. 23-31, September 1995.

[7] A. Kuchinsky, C. Pering, M. Creech, D. Freeze, B. Serra, and J. Gwizdka, "Fotofile: A Consumer Multimedia Organization and Retrieval System," *ACM CHI 99 Conference Proceedings*, pp. 496-503, May 1999.

[8] S. Mehrotra, K. Chakrabarti, M. Ortega, Y. Rui, and T. S. Huang, "Multimedia analysis and retrieval system," *In Proc. of the 3rd Int. Workshop on Information Retrieval Systems*, 1997.

[9] A. Pentland, R.W. Picard, and S. Sclaroff, "Photobook: Tools for Content-Based Manipulation of Image Databases," *Proc. Storage and Retrieval for Image and Video Databases II*, Vol. 2185, pp. 34-47, SPIE, Bellingham, Washington, 1994.

[10] J. R. Smith and S. F. Chang, "VisualSEEK: A Fully Automated Content-based Image Query System," *In Proceedings ACM Intern. Conf. Multimedia*, pp. 87-98, Boston, November 1996.

[11] M.D. Wilson and D.H. Marwick, "Describing the MIPS multimedia presentation system with the IMMPS reference model," *Computer Standards & Interfaces*, Vol. 18 (6&7), pp. 623-630, North Holland, December 1997.

[12] http://www.nhc.noaa.gov.

[13] http://www.virage.com.

[14] Synchronized Multimedia Integration Language (SMIL), http://www.w3.org/TR/smil20/.

[15] http://www.subres.com.