

Personal Video Manager: Managing and Mining Home Video Collections

Peng Wu¹ and Pere Obrador¹

Hewlett-Packard Laboratories, 1501 Page Mill Road, Palo Alto, CA 94304, U.S.A.
Hewlett-Packard Company

ABSTRACT

Home video collections constitute an important source of content to be experienced within the digital entertainment context. To make such content easy to access and reuse, various video analysis technologies have been researched and developed to extract video assets for management tasks, including video shot/scene detection, keyframe extraction, and video skimming/summarization. However, one less addressed issue is to investigate how useful those assets are in helping consumers managing their video collections and the usage pattern of the assets. In this paper, we present Personal Video Manager, both as a home video management system and an explorative research platform to enable a systematic analysis and understanding of consumers' demand on video assets and video processing technologies. For understanding consumer's interest, PVM adopts database management technologies to model and archive how consumers identify video assets and utilize them for management tasks. The PVM mining engine performs data mining on such archived data to mine useful knowledge of consumer's preference on video assets and behavior on utilizing the assets. As revealed in the experiment, consumer's interaction embeds rich information to be leveraged in developing more effective video analysis technologies.

Keywords: home video management, database, data mining

1 INTRODUCTION

Digital entertainment is largely about prompting consumers' experience on multimedia content. Specific to video content, many technologies have been developed to identify useful assets to organize, access and reuse content, such as video shot/scene detection³, keyframe extraction¹¹, and video skimming/summarization^{1,2,4,7}. On the other hand, given the progress achieved in those areas in the past decade, managing video content, particularly, home video content is still considered to be a time consuming and complex task that is beyond mass consumers' reach. One factor contributing to such gap is the lack of systematic analysis and understanding of what assets can best help consumers managing their video collections and in what ways consumers prefer to utilize the assets for the management tasks.

In this paper, we present Personal Video Manager that serves as both a home video management system and an explorative research platform to explore consumer's interest on video assets. As a home video management system, PVM provides following tools to help consumers to manage their video collections:

- Content ingestion: PVM offers a digitization station that can transform video content stored on various media (Hi-8, mini DV, VHS, etc.) into digital video files of various formats (MPEG-1, MPEG-2 and AVI).
- Content asset collection: PVM provides a web-based asset annotation application to enable consumers easily interact with digital video collections to identify video assets and archive the assets in a systematic way by utilizing server/client and database technologies.
- Content consumption: PVM provides a web-based searching/browsing application to enable consumers efficiently accessing their video collections using the video assets collected from the annotation application.

As a research platform, the issue that PVM is mostly interested to explore is what assets can best help consumers managing their video collections and in what ways they prefer to utilize assets for the management tasks. The cornerstone that makes such exploration possible is a set of carefully designed metadata schemas that represent the video assets. The metadata schemas regulate the scope of semantics that a consumer can use to characterize video content. A consumer utilizes the annotation application to identify video assets. The descriptions of the identified assets are stored according to the

1. {peng.wu, pere.obrador}@hp.com; phone: 1 650 8573683; fax: 1 650 8572951.

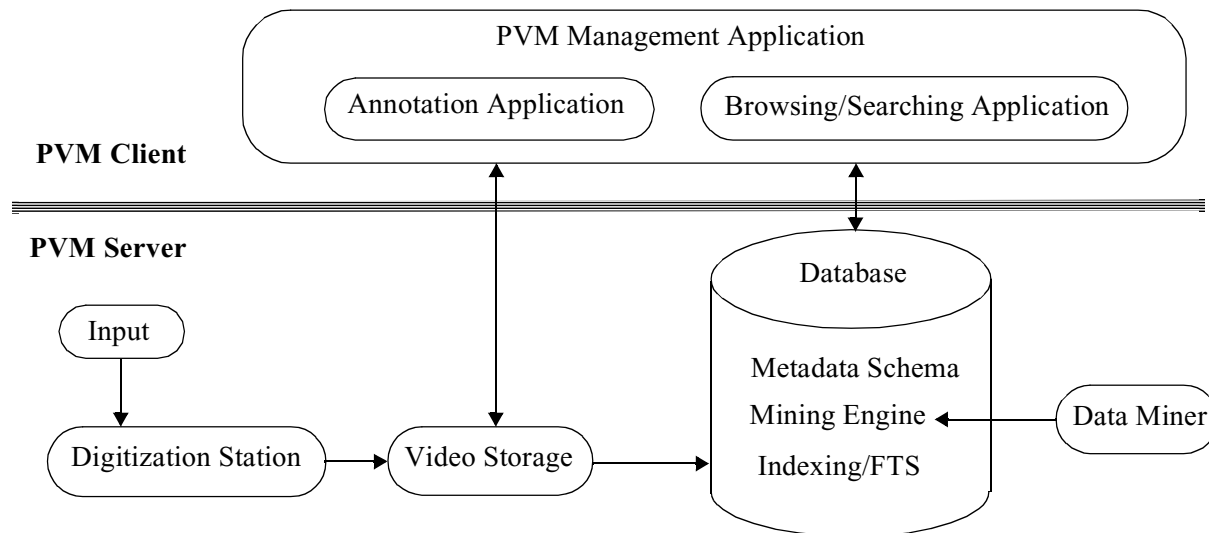


Fig 1. PVM architecture

schemas in the database. In the searching/browsing application, a consumer queries on the stored metadata to find assets of interest. As a result, the stored metadata and the query pattern on the metadata provide a data repository to mine consumers' interest on collecting and utilizing video assets. And the metadata schemas assure that a systematic analysis is feasible.

Given that standards like MPEG-7⁶ and MPV⁵ have been well-known for offering metadata descriptions of multimedia content, it is necessary to clarify that the emphasis of PVM, as a research platform, is to select a set of commonly used video assets, which is only a small subset of the video assets covered in the standards, and put them in a tractable environment to examine how they are utilized by consumers. By having a full control of the implementation of the metadata schemas and PVM management applications, i.e., the asset annotation and browsing/searching applications, we have an opportunity to collect the valuable data from consumers' interaction with video collections on how they like to characterize the content and utilize the characterization. For one example, the concept of "video keyframe" has been widely known from various research literatures and commercial products. However, do consumers really use the concept of "keyframe" to memorize the video content? Under what occasions, does this concept (not) apply? The answers to questions of such nature are fundamental to many multimedia asset management tasks but can only be found through studying the consumer experience and interaction with content. It is from this perspective we justify the research motivation of PVM.

In the following, we first give an overview of PVM on its architecture design and implementation; In Section 3, we describe the metadata schemas proposed and implemented in PVM. This is followed by the introduction of PVM management applications, including the asset annotation application and browsing/searching application in Section 4 and the data mining engine in Section 5. In Section 6, we describe the experiment results from mining the database and interfacing with consumers and end the paper in Section 7 with a short discussion.

2 PVM ARCHITECTURE

The PVM architecture is depicted in Figure 1. As shown in Figure 1, PVM adopts server/client architecture to support flexible and distributed content access and asset management. The captured videos, after being digitized and compressed, are stored on a networked storage server and accessible through PVM web-based management applications. A client uses PVM annotation application to extract video assets, which are represented as metadata. The metadata is stored in a networked database server, on which a set of schemas (tables) are defined to archive the metadata. In the PVM browsing/searching application, a client browses and searches on his/her video collections by querying on the metadata and based on client's query, the server utilizes the metadata to compose a customized view of the video content and return this view to the client.

Regarding the technologies adopted to implement this architecture, we use Microsoft SQL server 2000⁸ to archive the metadata and perform indexing and full text search to answer client's request. The mining engine is also provided as part of the server to be utilized by a data miner for data mining tasks. The PVM server and the client communicate through server pages. The server utilizes .NET techniques, including ADO.NET and ASP.NET to compose the server page delivered to the client. On the client side, JavaScript is used to support the client interaction with the server page, and control the embedded windows media player to view the video. In the next section, we discuss the design and implementation of the metadata schemas in detail.

3 PVM METADATA SCHEMA DESIGN

In PVM, the personal video assets are characterized and represented by metadata. The relational database model is utilized to describe assets and a relational database management system is developed to support the archive and management of the metadata assets. In relational database model⁸, a relation is an abstract term referring to an object or a relationship among objects. A relation consists of a relation schema and a relation instance. A relation instance is a table with columns. Each row in the table is called a record. The format of the record, such as the data type, associated constraints, is defined by the relation schema and described using the SQL syntax. The metadata schema essentially defines the nature of the data stored in a table.

3.1 Metadata schemas

The design of PVM metadata schemas has the following considerations:

- Considering the broad variety of the content of personal video collections, we can only capture a small set of assets in PVM. In PVM, six metadata schemas are designed for capturing personal video assets, which are "VideoAudioEvent", "VideoCameraMotion", "VideoShot", "VideoSegment", "VideoKeyframe" and "VideoVisualChange". Those assets are chosen for two reasons. First, they are frequently utilized in video management applications; Secondly, the investigation of the usefulness and preferred usage model on these categories of metadata can facilitate developing more efficient asset extraction algorithms.
- As personal video captures and reflects personal experience, the understanding and interpretation of the personal video are subjective as well. Bearing this in mind, we separate the metadata into "Editing Metadata" and "Raw Metadata". The "Editing Metadata" includes the six types of assets mentioned above. The "Raw Metadata" is captured by "VideoSource" schema. Given a video, the "Editing Metadata" captures a personalized view of the content, whereas "Raw Metadata" describes the unchangeable knowledge of the raw video material. A video can be associated with multiple sets of "Editing Metadata" but only one set of "Raw Metadata", such as length and format of the video file. The linkage between a personalized instance of "Editing Metadata" and "Raw Metadata" is captured by "VideoInfo" schema.

Figure 2 provides an overview of the schema implementation in PVM and how the "Editing Metadata" schemas are associated with the "Raw Metadata" through the "VideoInfo" schema. Some notes to help better understanding Figure 2 are given below:

- In both "VideoSource" and "VideoInfo" schemas, the attribute "Privacy" provides video content owner and editor control on which level the content can be exposed. More specifically, the value of the "Privacy" can be chosen from "Individual", "Group" and "Public" to enforce content exposure at different levels.
- All "Editing Metadata" schemas have the "Annotation" attribute, which is defined as a text string. The "Annotation" attribute provides a consumer to annotate a video asset in text, besides visual/audio attributes.
- "VideoAudioEvent" schema: for describing a distinguishable audio character, such as singing, laughing, applause, of a video subsequence.
- "VideoCameraMotion" schema: for specifying the type of camera motion (pan or zoom) and the direction of camera motion (in/out for zoom and left/right/up/down for pan) of a video subsequence.
- "VideoSegment" schema: for describing a video subsequence with some particular visual characters, including time gap, highlight, high motion intensity.
- "VideoVisualChange" schema: for describing the visual cues that indicates the transition of stories in the video content, which can happen within a shot or across shots.
- In both "VideoShot" and "VideoKeyframe" schemas, the "HyperlinkType" and "Hyperlink" attributes provide a way to specify a reference material (a file or a website) to better describe a keyframe or a shot.

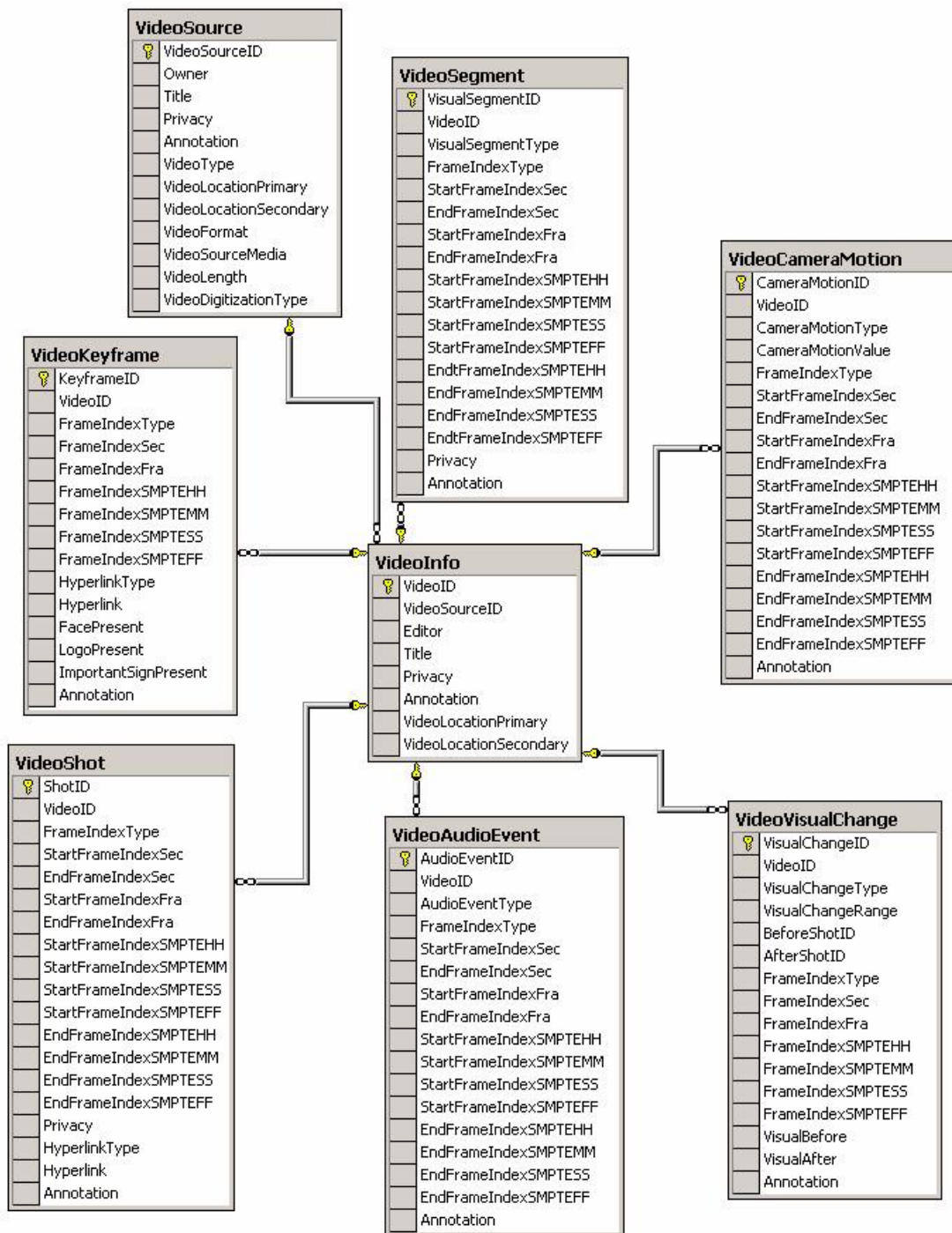


Fig 2. Overview of PVM metadata schemas: Each box represents one metadata schema and the bold text on top left corner of the box indicates the name of the relation that the schema describes. Each row in a box represents an attribute of the relation. The row with the key symbol besides is the attribute used to uniquely identify an instance of the relation. The “foreign key constraint”⁸ between two relations is represented by the link with key symbol on one end.

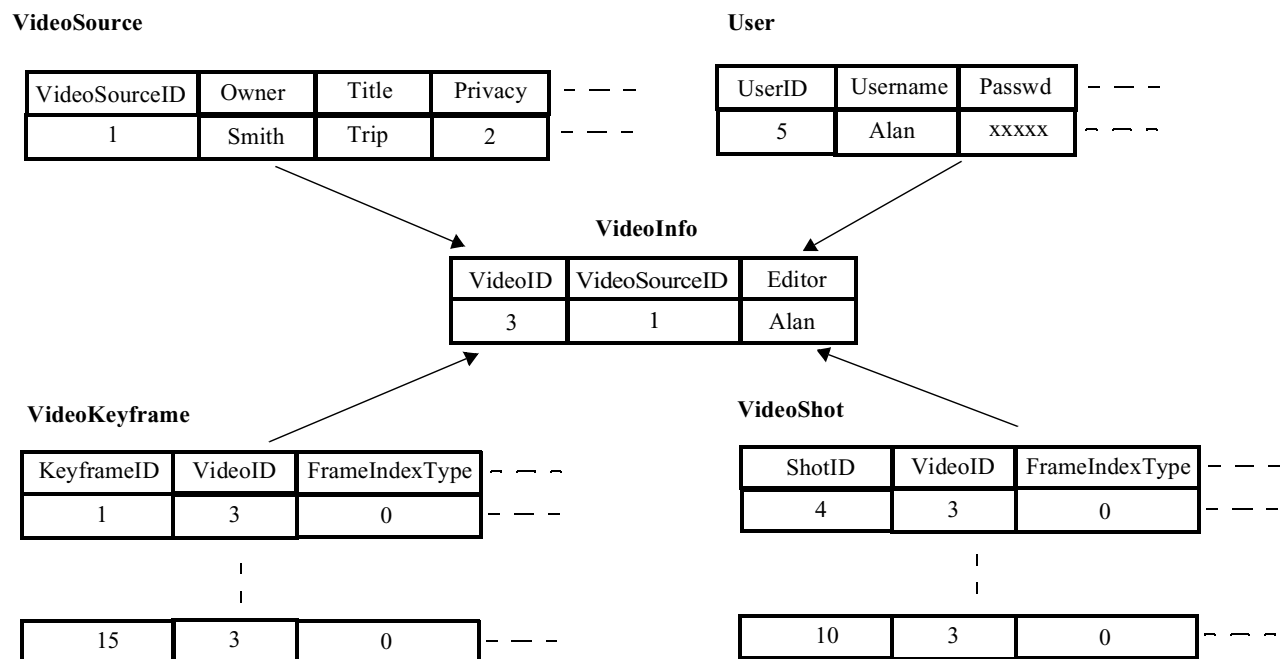


Fig 3. An illustrative example on metadata generation and association in PVM

A more detailed description of the schemas can be found in [10]. Next, we illustrate how database tables are instantiated and populated according to these schemas and how the assets are organized so that a reliable correspondence between the “Editing Metadata” and “Raw Metadata” can be maintained.

3.2 Metadata flow in PVM

Figure 3 shows a simplified example on how the metadata is generated according to the schema and how different categories of metadata are associated together. Once a video input is digitized and registered in database, an entry (row) is added on “VideoSource” table with a unique integer (“VideoSourceID”) automatically generated as the ID and assigned to that video. A registered video is protected by a user validation process. A user can access a video file if one of the following conditions is met: 1) The user is also the owner of the video file; 2) The privacy level is set to “group” or “public” by the content owner and the user is in the same group or in the same network as the owner. The justification of the access level is resolved by network authentication process, such as Windows NT authentication, when a user logs into PVM web application. A valid user is uniquely identified by the “UserID”.

Once a user logs into PVM and selects a video to annotate, an entry (row) is inserted into the “VideoInfo” table. The DBMS enforces a restriction that an entry has to have a unique pair of “VideoSourceID” and “Editor” to be inserted into the table. The “Editor” corresponds to “Username” in “User” table. For each valid entry, a unique integer “VideoID” is generated and assigned to that entry.

Using the PVM web annotation application, a user, after logging in, edits the chosen video to generate all sorts of metadata, such as metadata about “VideoKeyframe”, “VideoShot” and so on and the metadata is stored into different tables according to the metadata schemas. For illustration purpose, we use the keyframe asset as an example to describe the metadata flow. When a user indicates a frame in the chosen video file is a keyframe, a new entry (row) is inserted into the “VideoKeyframe” table. This entry, besides all the keyframe associated information, such as “FrameIndexType” and so on, has the “VideoID” field, which associates the entry in the “VideoKeyframe” table with an entry in the “VideoInfo” table. The schema design adopts the so called “foreign key constraint” in DBMS to assure that an entry in “VideoKeyframe” table has to have the “VideoID” field filled with a valid number pointing to a valid entry in the “VideoInfo” table. This constraint also applies to other “Editing Metadata” schemas.

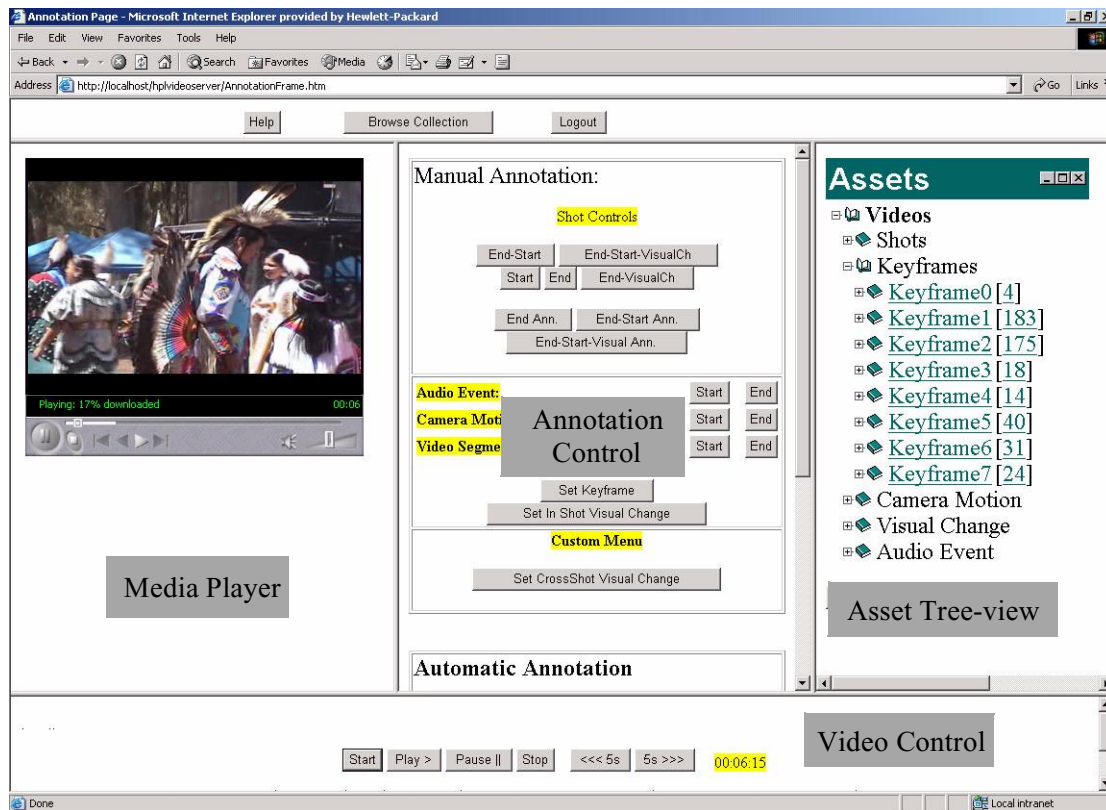


Fig 4. Annotation application

The schemas such designed allow the DBMS to use one table to archive one type of metadata, though the entries of the table are generated from different users annotating different video files. The “VideoID” field in the “Editing Metadata” schemas serves the purpose to identify with which video file that entry is associated and by whom that entry is generated. For one example, in Figure 3, by using the “VideoID” field of first entry of the “VideoKeyframe” table (“VideoID”=3), we can index back in the “VideoInfo” table to find the corresponding entry that has 3 as the value of “VideoID”. Thus, by looking at this entry, we can tell who the editor is (“Alan”, for this case) and which video file is edited (the one with “VideoSourceID”=1). Meanwhile, the unique restriction on “VideoInfo” states that an entry is considered to be a valid one as long as it has a unique pair of values of “VideoSourceID” and “Editor”. This restriction implies that a video file can be annotated by multiple valid users. However, for one video file, a user can only produce one set of “Editing Metadata” as the annotation of that file. In summary, the entries in “VideoInfo” table establish the association between the “Raw Metadata” and “Editing Metadata” and the “foreign key constraint” is adopted in the schema design to ensure the association is valid and accurate.

4 PVM MANAGEMENT APPLICATIONS

As introduced in Section 2, PVM implements two server/client based web applications to help users manage their video collections. The annotation application provides means to enable a user interacting with the video content. It is also responsible for translating the user’s interaction into metadata and archiving the metadata to the DBMS. The browsing/searching application provides an environment in which a user can quickly find the interested video content by using the metadata as the key to query the video collection.

In the following, we will give an overview of the main features and user interactions supported in both applications. Since many possible actions involved in the user’s interaction with video content, we will not go through the full fledged description of the interaction but only provide the description of the main functionalities.

4.1 Annotation application

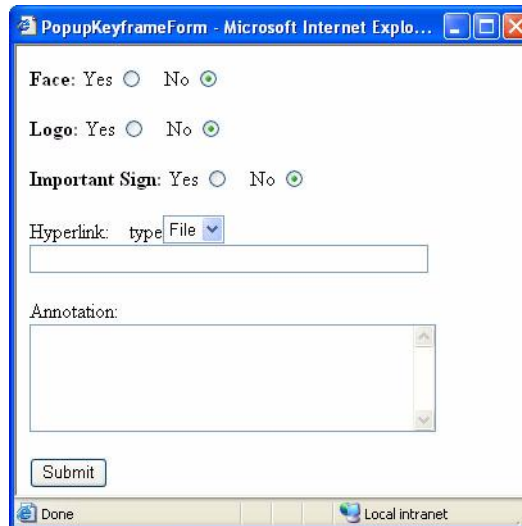


Fig 5. Pop-up frame to specify keyframe attributes

Figure 4 shows the interface of annotation web application. This interface is composed by four parts

- Media Player panel: a media player to display the video to be annotated
- Annotation Control panel: web controls to be triggered to annotate video assets.
- Asset Tree-view panel: a tree-view of the video content constructed using the available assets of that video
- Video Control panel: web controls of video display and simple browsing

To give a flavor of the user interaction, we illustrate how a keyframe is identified and annotated. Using the Media Player and web controls in the Video Control panel, a user can start playing the video and performing random seeking in the video file. Once a frame is spotted as a keyframe, the user can click on the “Set Keyframe” button located in the Annotation Control panel. Such interaction will trigger a pop-up frame, as shown in Figure 5, on which a set of web controls are presented to allow the user further specify the attributes associated with this keyframe. Once the user clicks on the “Submit” button on that frame, all the information, including the frame timestamp and attributes, will be submitted to the server and the DBMS on the server will validate the submitted information and generate a new entry on the “VideoKeyframe” table. Once the new entry is created on server’s database table, the Asset Tree-view panel will be updated to add one leaf under the “Keyframe” node to reflect current assets of the video file. Other types of assets can be annotated in a similar fashion.

It also should be noted that PVM architecture is designed to be open to automatic asset generation. The automatic annotation web controls shown in Figure 4 are presented to demonstrate the possible plug-in of the automatic asset extraction algorithms, such as face detection, keyframe extraction and so on.

4.2 Searching/Browsing application

Figure 6 shows the searching and browsing environment in which a user utilizes the collected metadata to easily find the interesting visual assets in the collection. It consists of the following parts:

- Query panel: web controls to enable query construction.
- Retrieval panel: web controls to list retrieved video assets and support user’s selection on listed items for a detailed asset view.
- Asset-view panel: a detailed view of the list item selected from the Retrieval Panel. The detailed view includes both the video display and the text annotation display.
- Video Control panel: web controls of video display and simple browsing

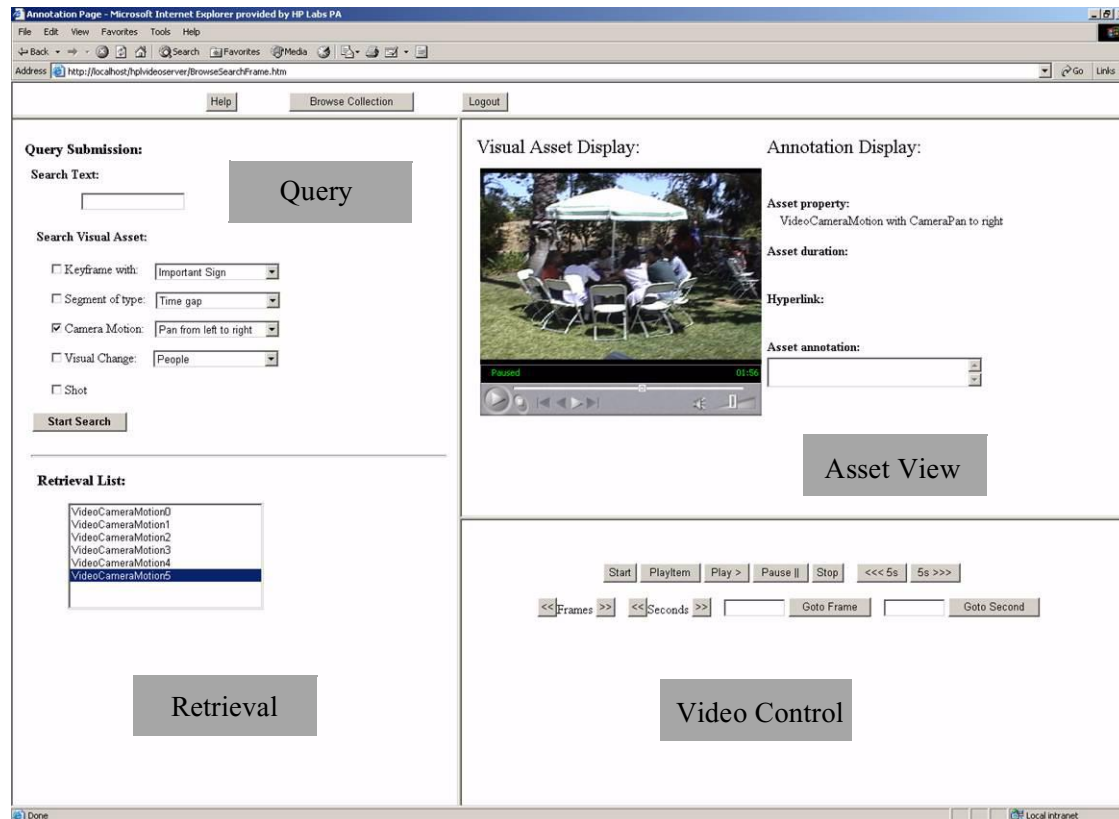


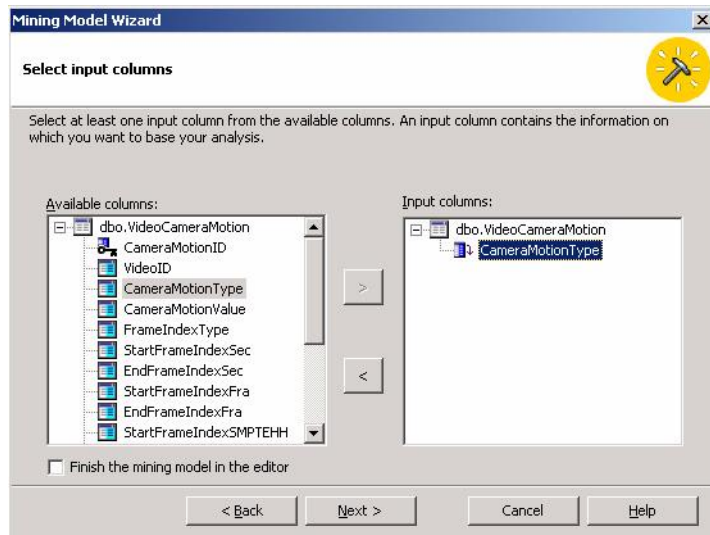
Fig 6. Browsing/Searching application

A user composes the query in the Query Panel. There are two types of queries are supported in current implementation: text query and visual asset query. The text query searches on the “Annotation” property of video assets by using the full-text search capability supported in Microsoft SQL Server 2000 and returns the matched video assets as retrievals. The other type of query is to let the user indicates which type of video assets to search for and what properties of that type of asset should have. The Query panel in Figure 6 shows an example of searching for assets with camera motion that is a pan from left to right. The matched video assets are returned to the user a list of clickable items in the Retrieval panel. The user can select one of them to view the details of the asset. The selected asset is presented in the Asset-view Panel, in which the Media Player displays the video frame or segment and the text for the text annotation associated with the asset.

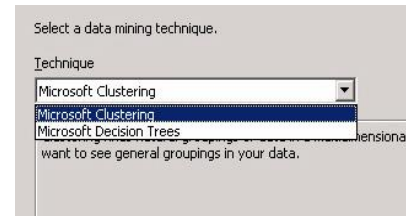
To summarize, with both the annotation and searching/browsing applications, PVM enables consumers efficiently collecting the metadata of videos and easily utilizing the metadata to access and reuse of the content.

5 PVM MINING ENGINE

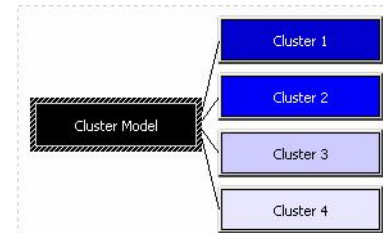
In PVM, two basic data mining techniques, k-means based clustering and decision tree based classification, offered by Microsoft analysis engine⁸ are available to perform analysis on data accumulated through consumer interaction with video content. Both the two techniques are well known ones in machine learning applications. The main contribution of Microsoft analysis engine is to enable an environment in which a data miner can flexibly apply the techniques to the data stored in the database. For example, the analysis engine allows a data miner to easily pick columns from any table on a SQL server as the input to the clustering engine and quickly present the graphical clustering results. This flexibility will be further illustrated in Section 6. It should be noted that while many complex and advanced data mining techniques, such as association rule mining, being researched and developed in data mining community, PVM’s current focus is mainly on how to practically utilize the available data mining tools to explore the accumulated video asset metadata accumulated in PVM.



(a)



(b)



(c)

Fig 7. Illustration of utilizing analysis engine to perform clustering on the “CameraMotionType” attribute of “VideoCameraMotion” table. In (c), there are 4 clustered resulted from clustering 61 instances. There are 25, 21, 8, and 7 instances in Cluster 1, 2, 3 and 4 respectively. The instances in Cluster 1, 2 represent camera zoom in and out respectively. The instances in Cluster 3, 4 represent camera pan from left and right respectively.

6 IMPLEMENTATION AND EXPERIMENTAL RESULTS

The PVM offers complete consumer experience, including content ingestion, asset annotation and content consumption. Currently, PVM hosts about 80 hours of home video content, contributed from 11 consumers. Consumers interact with content through PVM to manage their video collection, with the acknowledgement that PVM is a monitored environment. The design and implementation of PVM itself is also a evolving process that is tailored by consumers’ feedback. In this section, we first summarize the experience from interfacing consumers and then present the results from data mining process on the video asset metadata.

6.1 Notes from consumer feedback

Interfacing consumers directly and understanding their needs on video management are proven to be a very valuable and revealing experience. Some notes taken from this experience are listed below:

- At the early stage of designing metadata schemas, content privacy, from both content owner and content editor’s perspectives, is a concern frequently expressed by consumers. This concern motivates the realization of “Privacy” attribute in both “VideoSource” and “VideoInfo” tables to support the privacy control. However, a database design solution on “Privacy” notion is merely enough. One challenge to video analysis/management community is how to make it easy for consumers to express such notion effectively.
- From the consumer’s feedback, identifying a highlight or representative segment of video is a task that consumers very much like to do to summarize the content. However, it is also commonly commented that manually locating the boundary of a video segment is the most time consuming and tiring task for consumers. Motivated by such feedback, a semi-automatic approach⁹ is developed as an effort to answer this challenge.
- In searching/browsing applications, keyword search from the text annotation is more common than search on some visual cues, such as “camera pan from left to right”.

6.2 Practicing data mining

Systematic data mining is performed on video asset tables. Figure 7 illustrates one example to demonstrate the data mining practice in PVM. Figure 7 (a) shows the first step of data mining process, choosing the data to be analyzed. As shown in Figure 7 (a), a data miner can pick a table from database and chose one or multiple (only one shown in the exam-

ple) attributes as the targeted data to analyze. The second step is to choose which tool to use for the data analysis, as shown in Figure 7 (b). Currently, the available two options are decision tree and k-means clustering. The last step is to apply the tool to the chosen data and present the analysis result, as illustrated in Figure 7 (c). Through some simple clustering based data mining exercise as just described, some rather interesting patterns are revealed, as listed below:

- Camera zoom in and out are more frequently found than camera pan. In Figure 7 (c), the total number of “VideoCameraMotion” instances, identified by the consumers, is 61. Among them, 46 are camera zoom in/out and 15 are camera pan.
- Among the 15 camera pan instances, there about 8 of them are panning from left to right and 7 from right to left. However, by further looking into the time stamps of the instances, it is further revealed that many instances are temporally connected. There are 7 pair of continuous of panning from left to right or vice versa. Among them, only one is a pair panning from right to left. It seems to suggest that panning from left to right happens more naturally and frequently.
- A further study by cross checking the temporal overlap of the “VideoCameraMotion” instances with the “VideoAudioEvent” instances reveals that when camera is in motion, the audio rarely presents as speech, singing, or semantically meaningful audio segment.
- Regarding the “VideoKeyframe” asset, there are total 30 instances identified by the consumers. 15 out of 30 are ones with face(s) and 7 of the rest of them have important signs, such as birthday cake, identified. The fact that the number of keyframes is much fewer than the number of camera motions hints that the “keyframe” notion may not be intuitive to consumers. Also, the frames with non-natural objects, such as face, case, building, are more likely to be chosen as keyframes.

The experiment described above reveals some discovers among the accumulated “VideoCameraMotion” and “VideoKeyframe” metadata. More interestingly, it demonstrates that for other interested video assets, similar analysis can be performed to mine the data and reveal insights, if they exist, embedded within the data. Given such data is resulted from consumer’s interaction with video content, the revealed knowledge reflects consumer’s capture pattern and usage pattern on applying metadata for video management.

7 DISCUSSION

PVM is evolving as both a consumer oriented video management application and a research platform to explore consumer’s interest on video assets and demand on video technologies. Regarding the future development, there are two directions that are of our interest. First, as the experiments revealed, there is rich knowledge to be learned from consumers’ interaction with video. So it is our interest to further explore the metadata accumulated from consumer interaction with video content. This may involve utilizing more advanced data mining techniques, such as association rule learning, to perform mining on multiple metadata asset categories. The analysis may also incorporate the low lever features from video analysis in discovering the association between low level features and the formation of high level assets; The second direction is to investigate how to motivate better media analysis technologies by taking the mined knowledge into account. One such example is the semi-automatic highlight detection work described in [9]. As the mined knowledge is derived directly from consumers’ experience, it can server as a valuable driving force for developing media analysis technologies that address consumers’ needs more effectively.

ACKNOWLEDGEMENT

We thank our colleagues, Dan Tretter, Tong Zhang and Ullas Gargi for their valuable suggestions on the metadata schema design, and Baris Sumengen for his participation on the development of PVM. We are also thankful to the content providers who kindly lent us their video content to make our experiment possible.

REFERENCES

1. Michael G. Christel, Michael A. Smith, C. Roy Taylor, and David B. Winkler, “Evolving video skims into useful multimedia abstractions,” Proc. Human factors in computing systems, Los Angeles, California, United States, pp. 171-178, 1998.
2. Andreas Girgensohn, John Boreczky, Patrick Chiu, John Doherty, Jonathan Foote, Gene Golovchinsky, Shingo Uchihashi, and Lynn Wilcox, “A semi-automatic approach to home video editing,” Proc. the 13th annual ACM symposium on

User interface software and technology, San Diego, California, United States, pp. 81-89, 2000.

3. Ying Li, W. Ming and C.-C. Jay Kuo, "Semantic video content abstraction based on multiple cues," Proc. ICME 2001, Japan, August 2001.
4. Wei-Ying Ma and HongJiang Zhang, "An Indexing And Browsing System For Home Video", Invited paper, EUSIPCO'2000, 10th European Signal Processing Conference, 5-8, Sept. 2000, Tampere, Finland.
5. MusicPhotoVideo metadata standard, <http://www.osta.org/mpv/public/index.htm>.
6. Multimedia Content Description Interface standard (MPEG-7), <http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm>.
7. M.A. Smith, T. Kanade, "Video skimming and characterization through the combination of image and language understanding," Proc. IEEE Workshop on Content-Based Access of Image and Video Database, pp. 61-70, 1998.
8. Robert Vieira, Professional SQL Server 2000 Programming, Birmingham, UK : Wrox Press, c2000.
9. Peng Wu, "A Semi-automatic Approach to Detect Highlights for Home Video Annotation," IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Montreal, Canada, May 2004.
10. Peng Wu and Pere Obrador, Technical Report HPL-2004-162, "Personal Video Manager: toward better accessibility and reusability of personal video content".
11. Tong Zhang, "Intelligent Keyframe Extraction for Video Printing", Proc. of SPIE's Conference on Internet Multimedia Management Systems, vol. 5601, pp.25-35, Philadelphia, Oct.2004.