# Intelligent Streaming Video Data over the Web

J. R. Wang and N. Parameswaran
*School of Computer Science and Engineering, University of NSW*
*{jennyw,paramesh}@cse.unsw.edu.au*

## Abstract

*Delivering video over the Web has posted a big problem. Video files are extremely big and need hours or even tens of hours to download. The available bandwidth varies dynamically. The conventional proactive buffering approach used by current video players cannot fully satisfy user needs. This paper presents a system designed to allow efficient retrieving, browsing and real-time playing of videos through the Internet using a web browser. The scheme is domain-specific.*

## 1. Introduction

With the development of hardware and large capacity communication facilities, video has been more and more widely used in every field. The availability of digital videos has led to a new set of communication means with applications such as video-on-demand, video-conferencing and video-aided distance learning. Video files are extremely big compared with text and image files. Much research has been done on delivering video over the Internet. The techniques range from video server design [1], deliver agent [2], adaptive coding of video [3] and modification of low-level networks protocols [4, 5]. Current Microsoft wmplayer and RealNetworks RealPlayer use a proactive buffer which decodes a video file while it is in transmission. This will greatly reduce the waiting time but will freeze the video stream whenever the transmission is lagged behind play. Feng et al. [6] propose a proactive buffer management for deliver a video stream using the a priori information stored in the video stream. The scheme also needs to monitor the available bandwidth.

There are fatal problems with the above-mentioned approaches. Developing a new server, an agent or a network protocol is not scalable. We cannot predict and restrict the number of accesses at one time. No matter how efficient a server, an agent or a protocol is, it will be stuck when the number of accesses increases. This situation could happen in video-on-demand systems. Using a proactive buffer or adaptive coding video streams is still impractical for real-time applications. We do not know what is the proper size of the buffer. If the buffer is sufficiently large, it may be equivalent to download the whole video. Adding the buffer management will also add extra overhead to the video server and reduce the efficiency.

In this paper, techniques used in the effective retrieving and viewing of videos through the Web are explored and the design and implement of an online browsing site is presented. We discuss our multi-access hierarchical structure for organizing video streams in Section 2. An implementation including encoding, playing and structural browsing is presented in Section 3. Section 4 analyzes the system in terms of memory usage, and the usability of the system. Finally, there is a conclusion.

## 2. Multi-access hierarchical structure

Current research in video representation and modeling [7; 8; 9, 10] provide ways of partitioning video clips into smaller shots and indexing them based on visual features, such as colour, texture and motion, using the key frame of each shot. These low-level visual features have provided success in searching for similar video scenes given an example video shot (query by example). However, this kind of operation has limited capability and does not properly address the richness of video data and the versatile use of such data. Consequently, this approach has limited applications.

Based on our understanding of movie industry and film semiotics pioneered by the film theorist Christian Metz [11], we propose a general metadata to represent video in multi-layer strata with general information at the top level of the strata and details at the bottom of the strata. The metadata includes a multi-layer physical stratum and a five-level semantic paradigm. The physical stratum includes objects, I-frames, frames, takes, shots, plays, scenes and plots. An object is a region in a frame. It can be a semantic meaningful object such as people, car, building, beach, sky, etc, or a visually sensible region such as a region of the same colour, similar texture, etc. It can also be an interactively grouped region. We call both semantic objects and visual objects those are perceptual objects. A frame is one complete unit in presentation. A stack of consecutive frames forms a video sequence. An I-frame is an identification frame among a group of frames. It is consistent with the definition of I-frames in the MPEG compression standard. Many literatures use the term key-frame. We use I-frame instead of key-frame for many reasons. First, the key-frame is not a well-defined

Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI'04)
0-7695-2100-2/04 $ 20.00 IEEE

term. There is also no standard criterion to extract key-frames. In order to fully represent the video sequence, some researchers even suggest using a complicated frame, such as a storyboard sketch, which may not be any one inside the sequence, as the key-frame. Besides, an I-frame gives enough representation of the video sequence in our model. A take is a sequence of frames which contain one action of a perceptual object. A shot is a sequence of frames, which give a clear description of certain perceptual objects. A video sequence containing multiple perceptual objects performing many actions at the same location forms a play. The same location can appear many times in a video. These appearances can be from different angles in terms of the camera. Putting these multi-angle views into a mosaic forms a scene. Multiple plays developed under the same story form a plot. Note that the stratum allows overlap between takes and shots, and plots and scenes.

We identify five levels of cinematic codification for the semantic video hierarchy:

- the perceptual level: the level at which visual phenomena become perceptually meaningful, the level at which distinctions are perceived by a viewer within the perceptual object. This level includes visual objects, visual characteristics (colour, texture, motion, etc) and photographic characteristics (short take, long take, close-up, etc).
- the diegetic level: the four-dimensional spatio-temporal world posited by a video image or sequence of video images, including the spatio-temporal descriptions of agents, objects, actions, and events that take place within that world.
- the connotative level: metaphorical, analogical, and associative meaning that the denoted (i.e. diegetic) objects and events of a video may have. The connotative level captures the codes that define the culture of a social group and are considered "natural" within the group.
- the subtextual level: more specialized meanings of symbols and signifiers. It can be customized using the terms of film industry or databases.
- the cinematic level: the specifics of formal film and video techniques incorporated in the production of expressive artefacts. This level includes camera movement, camera operations, multi-camera views, lighting schemes, and optical effects.

Annotation is done semi-automatically and an XML description is generated automatically. The details of implementation is beyond the scope of this paper.

## 3. Implementation

The implementation includes encoding, playing and structural browsing.

### 3.1. Video encoding

To achieve real-time viewing of the video clips through the Internet, the original video clip needs to be encoded into streams that match users' connection speeds. It was decided that in this application, every clip would be encoded into streams in the most commonly used Internet connection speeds.

**3.1.1. Encoder SDK.** Various media encoder SDKs were studied and tested. It was found that Microsoft Media Encoder was the most suitable for this application because of the features it offers. These are described below.

- Environment and Format: Microsoft Media Encoder 7 SDK works in windows 98/2000/NT. It offers encoding methods for most of the commonly used video and audio formats including ".bmp", ".wmv", ".wma" etc. It outputs encoded streams in windows media file format ".wmv" or ".wma" which can be played with Microsoft Media Player.
- Encoding Profiles: The Encoder SDK provides functions to encode input video files according to thirty-three different profiles. They cover almost all common communication bandwidths including ISDN, XSDL, LAN, Web-Server, broadband, etc
- Other Functionalities: Windows Media Encoder 7 SDK also provides other useful functionalities suitable for this development.

Furthermore, the encoding functions provided in the SDK allow encoding for either video only, audio only or both streams. This also offers much freedom for future development.

**3.1.2. Streaming and storage.** In this system, all clip segments in a hierarchical structure need to be playable through the web browser in real time for all the three connection speeds described before. Several options in streaming and storing video clips were examined, which include encoding at browsing time and encoding and storing every clip in the hierarchy. We resort the last approach because none of the other methods suits the currently available tools.

### 3.2. Playing and synchronization

**3.2.1. Playing video in IE5.** Microsoft Media Player 6 and 7 are capable of playing video/audio clips in almost all common formats. And the player control is a standard ActiveX control that uses Microsoft Component Object Model (COM). Using the SDK, the players can be programmed on a standard HTML Web page using JScript or VBScript. Furthermore, Media player 6 is installed by default with Windows 95/98/2000/NT.

Therefore it is the most suitable player SDK to be used for this project.

**3.2.2. Closing captioning.** Media Player 6 and 7 SDKs also support captioning text to be played with videos through the use of Synchronized Accessible Media Interchange (SAMI)--(.smi) files containing text strings associated with specified times within the video. This feature provides our video browsing system the capability to display subtitles for foreign language films or for assisting hearing impaired users.

Below is a sample SAMI file for the clip "Meeting the Children" in "the Sound of music" of Figure 1.

```
<SAMI>
<HEAD><Title>Close Captioning Sample</Title>
</HEAD>
<BODY>
<SYNC Start=1000>
<P Class=ENUSCC>Maria Meeting the Children
<SYNC Start=4000>
<P Class=ENUSCC>Captain introducing the children to Maria.
<SYNC Start=12000>
<P Class=ENUSCC>Liesl
<SYNC Start=14000>
<P Class=ENUSCC>Friedrich
…
</BODY>
</SAMI>
```

**Figure 1**. Closed Captioning Example

### 3.3. Browsing and navigation

Browsing and navigation can be illustrated in a diagram shown in Figure 2.

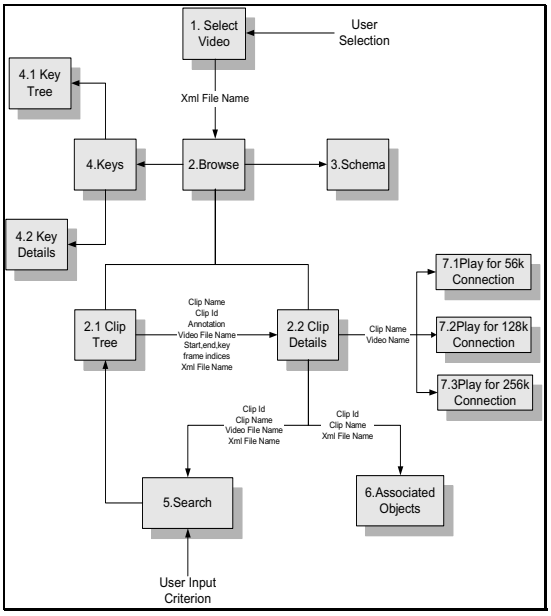



**Figure 2**. Data Flow Diagram

**3.3.1. Video structure tree.** Video structure is implemented as an interactive tree using JavaScript. The tree can be expanded or collapsed at any intermediate node (a node with children). The format of the tree is very similar to windows help files and thus is familiar to most windows users. IE5 and most other commonly used browsers support JavaScript by default and so no extra software needs to be installed for browsing which further increases usability of the system.

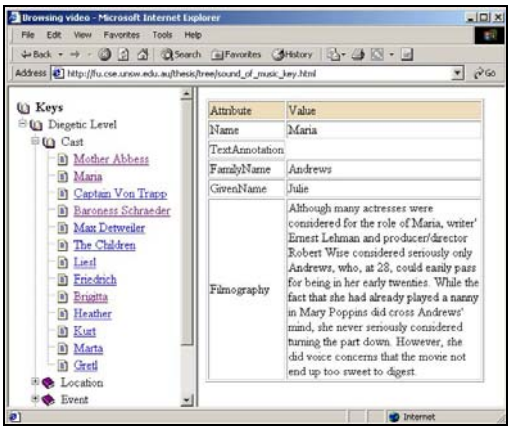



**Figure 3**. Object Details

**3.3.2. Object (key) details.** In the metadata structure used, objects (keys) can be associated with video clips. For every video structure, a hierarchical schema is defined for object types. Each type defines a different set of attributes, which can be used in keyword based searching. The system allows users to view the object schema tree and details of each object in the video structure. Users can also view objects associated with a specific clip at the clip details page, as shown in Figure 3.

## 4. System evaluation

The system encodes all clip segments in the video structure for three different connection profiles and stores them on hard drives on the server side. Significant amount of processing and storage resources are therefore consumed in the running of the system.

### 4.1. Memory resource

The encoded video clips, which need to be stored at server side take up significant amount of memory resource.

A major summary on memory usage is done for "the sound of music":

- Original "sound of music" video clip size = 149MB
- Number of segments in video structure = 25
- Memory used in storing encoded clips for 56k connection = 9.26MB
- Memory used in storing encoded clips for 128k connection = 31.4MB
- Memory used in storing encoded clips for 256k connection = 68.6MB

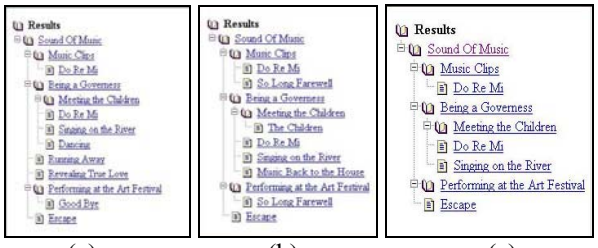
- Total memory used in storing encoded clips = 109.26MB

Memory consumed in storing encoded clips can be effectively reduced if option 3 described in Section 3.1.2 can be implemented. Currently since every clip in the structure is separately encoded and stored, there is overlapping between encoded segments and the level of overlapping depends on the video structure. However if the option of connecting multiple clips into one at play time becomes available, then only bottom level clips need to be encoded and stored and overlapping can be completely avoided. In the case of "sound of music", memory used in storing encoded clips could be reduced by more than 50 percent.

### 4.2. System interface and operations

Browsing: The main browsing page consists of video structure as well as clip information. By clicking a clip in the tree structure, users can view the name, key frame and annotation of the clip and also choose to search or play it.

Search: Search can be done at each clip in the video structure and results are displayed in the same format as video structure tree. Thus users can browse searching result exactly the same way as browsing the original video tree and they can further search under any clip in the searching result tree. A searching criterion can be formed by "and" or "or" simple conditions. As can be seen in the figures below that every clip in the result tree in Figure 5 satisfies both of the two components in its composite condition, ie. it contains both "Maria" and "The Children". This result tree is in fact the intersection of the results of searching for "Name = Maria" (Figure 5a) and "Name=The Children" (Figure 5b) as shown.




(a) (b) (c)

**Figure 5**. Search Result for "Name=Maria" (a), "Name=The Children" (b) and both (c)

## 5. Conclusion

In this paper, an online video browsing system is presented and its design issues are discussed. The system hosts a website which allows the browsing of video metadata structures and real-time playing of video segments through the Internet using IE5. Furthermore, a set of tools are developed to automate the generation of files required to run the website.

Tests on both resources usage and system usability have been conducted. The system is shown to be user friendly, efficient and compatible with most web browsers in the Windows environment. Possible applications of such an online browsing system include distant education, news dissemination, ad tracking and movie archives. Web casting with the aid of well-designed metadata structure can make video communication through the Internet a truly viable method.

## 6. Acknowledgement

This project is supported by Australia Research Council Linkage Grant (LP0347156).

## 7. References

[1] McCanne, S & Jacobson, V (1995). VIC: a flexible framework for packet video. Proc of ACM Multimedia.

[2] Floyd, S; Jacobson, V; Liu, C G; McCanne, S & Zhang, L (1997). A reliable multicast frame-work for light weight sessions and application level framing. IEEE/ACM Transactions on Networking December.

[3] Rowe, L A; Patel, K; Smith, B C & Liu, K (1994). MPEG video in software: representation, transmission and playback. Proc. SPIE: High-Speed Networking and Multimedia Computing, vol 2188, pp.134-144.

[4] McManus, J & Ross, K W (1996). Video on demand over ATM: constant-rate transmission and transport. Proc IEEE INFOCOM, pp.1357-1362.

[5] Chen, Z; Tan, S M; Campbell, R & Li, Y (1998). Real-time video and audio in the Wold Wide Web. World Wide Web Journal Vol 1.

[6] Feng, W, Krishnaswami, B & Prabhudev, A (1998). Proactive buffer management for the streamed delivery of stored video. Proc of ACM Multimedia'98, pp.285-290.

[7] Arman, F; Hsu, A & Chiu, M (1993). Image processing on compressed data for large video databases. Proc. ACM Multimedia, pp.267-272.

[8] Iyengar, G & Lippman, A B (1996). Videobook: an experiment in characterization of video. Proc. of IEEE Intl. Conf. on Image Processing, vol. 3, pp.855-858.

[9] Hampapur, A; Gupta, A; Horowitz, B; Shu, C; Fuller, C; Bach, J & Jain, R (1997). Virage video engine. Proc. of SPIE, vol. 3022, pp.188-200.

[10] Chang, S F; Chen, W; Meng, H; Sundaram, H & Zhong, D (1997). VideoQ: an automated content based video search system using visual cues. Proc. ACM Multimedia.

[11] Metz, C (1974). Film Language: A Semiotics of the Cinema, trans. by M. Taylor, The University of Chicago Press.