LYU9904

# Multi Model Digital Video Library

Department of Computer Science and Engineering,
The Chines University of Hong Kong

Supervisor:

Prof. Michael Lyu

Team Members:

Jacky Ma

Joan Chung

# 1. Abstract

Our project, titled "Multi Model Digital Video Library", is targeted to learn issues about digital video libraries and implement a small-scale model. Digital video library becomes more and more important as the Internet technology evolves, and will soon affecting the mode which people consumes media data, and thus the media industry. The recent development of digital video libraries is to employ various new technologies for building the database, indexing the contents, searching and retrieving the video resources in effective and efficient way. The breakthrough is to extract and index the "semantic" of a video data, which create a new "Interactive Video" view against the traditional "Interrupted Video" view. In our work, we built an application that can play synchronized video data with the transcript, where the transcript can be treated as a form of the "semantic" representation of the video.

# 2. Table of Content

# 3. Introduction

The raise of Internet in the 20th Century changes the way people live. Estimation shows that there are 50 million people using the Internet on a regular basis. Entertainment, education, commercial activities, etc. is being merged with the irresistible trend of Internet. Multimedia information is making a revolution in this trend. New technologies realized the concept of digital video library. And this initiates our project.

In this report, we will first introduce the background of digital video library, followed by general issues about DVL and their addressing techniques. After reviewing the general ideas of DVL, we will introduce our work, sectioned by considerations and approach we take, design and implementation details, and discussions about our work. Finally, we will have a conclusion and mention some other works related to this topic.

# 4. Background

Video is not like pure texts or images, it is large in size and contains audio and sequence of images. It will be much more complex to handle video in computer world. There are many questions arise before establish a digital video library. How do you build a vast video database? How do you index the video contents? How can you search and retrieve the video resources efficiently? How can you let users to view the resources conveniently and effectively? New techniques are needed to organize and search these vast data collections and retrieve the most relevant selections.

Emerging techniques for digital video libraries will allow independent, self-motivated access to information for self-learning, exploration and research. The potential impact on training and education delivery is critical. Digital video libraries offer the potential to deliver vicarious field trips to places that are too dangerous or expensive to visit in person. It allows virtual guest speakers and topic experts to deliver talks and be interviewed in the classroom and at home. It can provide virtual access to rare, unique, expensive, dangerous materials in a safe, comfortable educational setting. Exploring Antarctica can be done without the need for a winter coat; the results of combining volatile chemicals can be witnessed without fear of bodily harm. However, in order to make these true, the information embedded within the digital library must become easy to find, manage and use. These are

challenging issues when video is included as a primary component of a digital library.

Video causes unique problems because of the difficulties in representing its contents.

It is well known that if you electronically scan a page of a book into a raster image, the image will use a significantly greater amount of memory space than an ASCII representation of the original text. While page description languages may be more efficient, if the page contains many images, a raster image may be the only choice for representation. Video is not only imagery, but consists about 30 images per second. The adage “a picture is worth a thousand words” was never more appropriate. Details descriptions of video images can be many thousands of words and even a short video clip description can be massive. However, the alternative of no description leaves even the shortest video clip a black box, giving the user no way to know what is within it. The issues on creating a digital video library (gathering video, representing its contents, and segmenting it appropriately) and utilizing and exploring the library (retrieving and browsing effectively) are also challenging parts in this topic.

# 5. Issues about Digital Video Library

In a good digital video library (DVL), it should have large and well-developed digitized video databases, have efficient indexing and retrieving methods of the resources in the library, and provide online access for its users. Through the digital video library, users can find out the most related materials which they want conveniently and rapidly by queries.

## 5.1. Building Video Databases

Digital video takes a tremendous amount of space. Therefore, in order to build a video database, we need to consider the video format for the databases. It is important to choose a video format which can save space but still maintain the quality of video. A single high quality, uncompressed video channel would require a bandwidth of 200 million bits per second. Such bandwidth requirements are not practical today or perhaps ever, so the quality of the video may be reduced and compression schemes used to make possible the inclusion of video into digital libraries. For example, the MPEG algorithm for video compression was designed to deliver good quality at a very high compression ratio and random access to various points within the sequence. It is a scalable algorithm allowing more quality at the expense of requiring greater bandwidth. The MPEG1 SIF resolution will work for standard CD-ROM bandwidth requirements (1.2 Megabits per second), allowing 352 x 240 resolution at 30 frames per second or 352 x 288 resolution at 25 frames per second, thus delivering VHS quality NTSC/PAL video.

Even before the video can be digitized and placed into the library, a number of intellectual property rights issues need to be resolve. As new legal rules will likely be established and evolve as consumers and publishers move fully into the electronic age where copying is simple, accurate, and cheap [Samuelson95, Samuelson93]. In the coming years, the U.S. Library of Congress will play a leading role in the resolution of problems of copyright and intellectual property rights with respect to digital libraries [Becker95]. For now, these problems can be dealt with in the following ways:

- only include public domain resources in the digital library, or resources for which you have proper permissions.

- make arrangements with resource providers for remuneration and proper attribution; then control access to the digital video library so that owners and retailers of information can be paid when their materials are accessed. NetBill developed by CMU is one such electronic commerce mechanism enabling a market economy in information and providing all of the services necessary to account for intellectual property delivered via a network.

Another consideration in the creation of a digital library is enabling access to the resources in the databases. Even with MPEG1 compression, a thousand

hours of video will take approximately a terabyte (1024 gigabytes) of storage. It is so unlikely that user workstations will have the complete library stored locally at their machines. Rather, a key element of on-line digital video libraries will be the communication fabric through which media servers and satellite (user) nodes are interconnected. Traditional modem-based access over voice-grade phone lines is not adequate for this multimedia application, as evidenced by the difficulty in trying to move VHS-quality video between arbitrary sites on the Internet. The ideal fabric has the following characteristics:

- communication should be transparent to the user. Special-purpose hardware and software support should be minimized in both server and slave nodes.

- communication services must be cost effective, implying that link capability bandwidth) be scalable to match the needs of a given node. Server nodes, for example, will require the highest bandwidth because they are shared among a number of satellite nodes.

- the deployment of a custom communication network should be avoided. The most cost effective, and timely, solution will build on communication services already available or in field-test.

## 5.2. Indexing the Video Contents

A library cannot be very effective if it is merely a collection of information without some understanding of what is contained in that collection. Without that understanding it could take hundreds of hours of viewing to determine if an item of interest is in a 1000-hour video library.

Information is found best on the Internet when the providers augment the information with rich keywords and descriptors, provide links to related information, and allow the contents of their pages to be searched and indexed. There is a long history of sophisticated parsing and indexing for text processing in various structured forms, from ASCII to PostScript to SGML and HTML. However, it is not as simple to index video content.

An hour-long motion video segment clearly contains some information suitable for indexing, so that user can find an item of interest within it. The problem is not the lack of information in video, but rather the inaccessibility of that information to our primarily text-based information retrieval mechanisms today. In fact, the video likely contains an overabundance of information, conveyed in both the video signal (camera motion, scene changes, colors) and the audio signal (noises, silence, dialogue). A common practice today is to log or tag the video with keywords and other forms of structured text to identify its contents. Such text descriptors have the following limitations:

- Manual processes are tedious and time consuming.

- Manual processes are seriously incomplete. Even if full transcripts of the audio track are entered, other information about the video will almost surely be left out, such as the identity of persons and objects in each scene.

- Transcripts are inaccurate, with mistypings and incorrect classifications often introduced.

- Text descriptors are biased by whatever predetermined structures are used to classify the video contents. For example, if you have a classification of "inside or outside", how do you tag a scene of people in a cave?

- Cinematic information is complex and difficult to describe, especially for non-experts. For example, in an establishing shot that zooms from a wide angle to a close-up, determining the point when the scene changed is open to interpretation.

- Text descriptors are biased by the ambiguity of natural language. Example again, one indexer may decide to label a particular video segment as occurring in a public vehicle. Another may decide to label the same segment as occurring in a bus. These different tags have implications for later browsing and retrieval of the video.

In order to reduce these limitations, there should be some technology supports to handle the indexing automatically instead of manually.

## 5.3. Breaking the Video into Segments

Anyone who has retrieved video from the Internet may realize that it takes a long time to move a video clip from one location to another because of its size. If a library consists of only 30 minutes clips, when users check one out, it may take them 30 minutes to determine whether the clip met their needs. Returning a full one-half hour video with only one minute is relevant is much worse than returning a complete book with one chapter is needed. With a book, tables of contents allow users to quickly find the material they need. However, since the time to scan a video cannot be dramatically shorter than the real time of the video clips, a digital video library should be efficient at giving users the relevant material. To make a faster retrieval and viewing, the digital video library will need to support:

- partitioning video into small-sized clips
- alternate representations of the video

### 5.3.1. Video Paragraphing

Just as textbooks can be decomposed into paragraphs with different chapters and subtitles, video library can be partitioned into video paragraphs. There are difficulties arise in how to carry out video paragraphing. Analogous structure is contained in video through scenes, shots and camera motions. It is difficult and tremendous to describe this structure manually.

The boundaries of paragraph could be done by parsing and indexing on the video segment. Some videos, such as news broadcasts, have a well-defined

structure which could be parsed into short video paragraphs for different news stories, sports and weather. Techniques monitoring the video signal can break the video into sequences sharing the same spatial location. These scenes could be used as paragraphs.

However, physically decomposing a video library into fixed number of small video files will not meet the future needs of the library user. What we need are "representations which make clips, not representations of clips"[Davis94]. A more flexible alternative is to logically segment the library by adding sets of video paragraph markers and indices, but still keeping the video data intact in its original context. This improvement allows later enrichment of the description of the video content. The original material can be retrieved easily without redundancy for the user if desired. Moreover, the video clip to return to the user can be based dynamically on user and query characteristics, with more annotations allowing more possible segmentations of the video data. In order for a digital video library to be logically segmented as such, the system must be capable of delivering a subset of a movie (rather than having that subset stored as its own movie) quickly and efficiently to the user. Video compression schemes will have to be chosen carefully for the library to retain the necessary random access within a video to allow it to be logically segmented.

### 5.3.2. Alternate Representations for Video Clips

In addition to trying to size the video clips appropriately, the digital video library can provide the users alternate representations or layers of information for the video[Stevens94, Zhang95, Rao95]. Users could then review a given layer of information before deciding whether to incur the cost of richer layers of information or the complete video clip. For example, a given half hour video may have a text title, a text abstract, a full text transcript, a representative single image, and a representative one minute "skim" video, all in addition to the full video itself. The user could quickly review the title and perhaps the representative image, decide on whether to view the abstract and maybe the full transcript, and finally the user may decide whether to retrieve and view the full video.

## 5.4. Retrieving Video

An important part of the function of a digital video library is that it should provide the utility for users to get the information they need from it easily and efficiently. There are two standard to measure the performance in information retrieval: recall and precision. Recall is the proportion of relevant documents that are actually retrieved. Precision is the proportion of retrieved documents that are actually relevant. These two measures may be traded off one for the other. Returning one document that is a known match to a query guarantees 100% precision, but fails at recall if a number of other documents were relevant as well. On the other hand, returning all of the library's contents for a query guarantees 100% recall, but fail at precision and filtering the

information. The goal of information retrieval is to maximize both recall and precision.

In many information systems, precision is maximized by narrowing the domain considerably, extensively indexing the data according to the parameters of the domain, and allowing queries only via those parameters. This approach is taken by many CD-ROM data sets. For example, a CD-ROM on animals might fully index the data by genus, species, habitat, diet, growth rate, estimated population, and other biological and environmental factors. The data becomes very useful for its given purpose (e.g. studies on animals), but this approach has a few limitations:

- Data could really only be added if it falls within the domain established by the predefined indices. For example, if we want to add the information about the countries animals coming from to the CD-ROM, new indices would have to be added as well.
- Access to the data is limited by the predefined indices. A user may not be able to find all birds that are blue if colour is not one of the attributes which were indexed.

Researchers of multimedia information systems have raised concerns over the difficulties in adequately indexing a video database so that it can be used as a general purpose library, rather than a more narrow domain [Davis94, Zhang95]. For general purpose use, there may not be enough domain knowledge to apply to the user's query and to the library index to return only a very small subset of the library to the user matching just the given query. For example, in a soccer-only library, a query about goal can be interpreted to mean a score, and just those appropriate materials can be retrieved accordingly. In a more open context, goal could mean a score in hockey or a general aim or objective. A larger set of results will need to be returned to the user, given a less domain knowledge from which to leverage.

In attempting to create a general purpose digital video library, precision may have to be sacrificed in order to ensure that the material the user is interested in will be recalled in the result set. The result set may then become quite large, so the user may need to filter the set and decide what is important. Three principle issues with respect to searching for information are:

- how to let the user quickly skim the video objects to locate sections of interest
- how to let the user adjust the size of the video objects returned
- how to aid users in the identification of desired video when multiple objects are returned.

### 5.4.1. Returning Small Pieces

There are about 150 spoken words per minute of "talking head" video. One-hour video will contain 9000 words, which is about 15 pages of text. If a user issues a query and receives ten half-hour video clips, it could take him/she hours to review the results to determine the relevance. If the results set were instead ten two-minute clips, then the review time is reduced considerably. In

order to return small and relevant clips, the video contents need to be indexed well and sized appropriately.

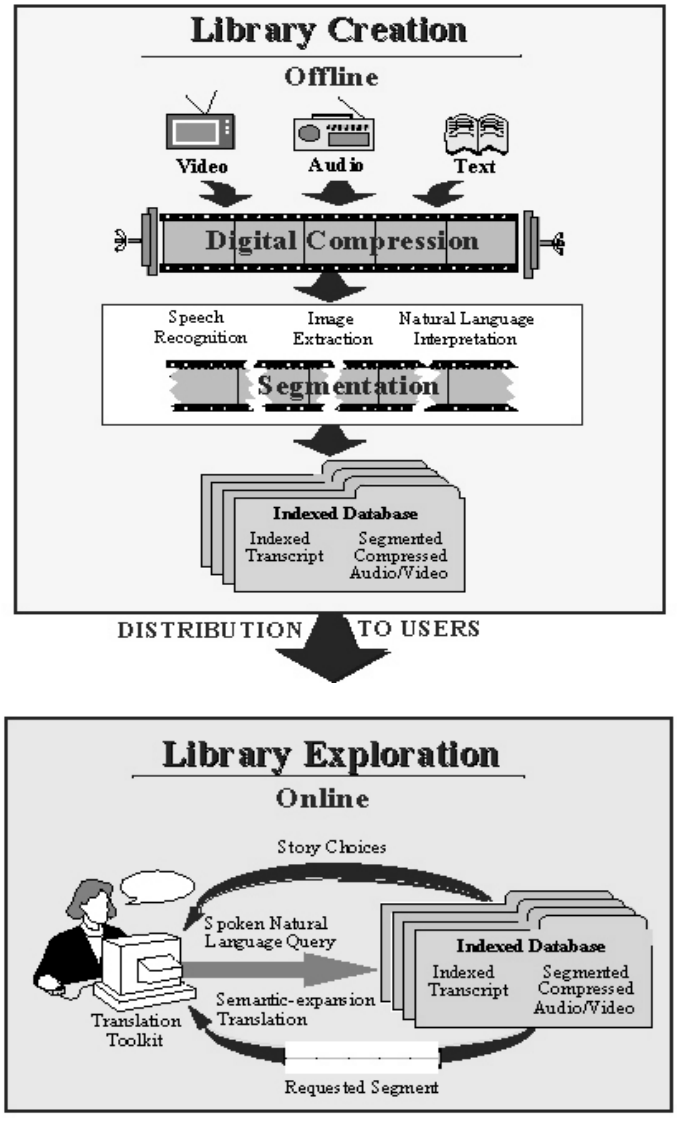
### 5.4.2. Information Visualization

Users often wish to peruse video much as they flip through the pages of a book. Unfortunately, today's mechanisms for this are inadequate. Tools have been created to facilitate sound browsing which present graphical representations of the audio waveform to the user to aid identification of locations of interest. However, this has been shown to be useful only for audio segments under three minutes [Degen92]. When searching for a specific piece of information in hours of audio or video, other mechanisms will be required. The results from a query may be too large to be effectively handled with conventional presentations such as a scrollable list. To enable better filtering and browsing, the features deemed important by the user should be emphasized and made visible. What are these features, though, and how can they be made visible, especially if the digital video library is general purpose rather than specialized to a particular domain? These questions return us back to the problem of identifying the content within the video data and representing it in forms that facilitate browsing, visualization, and retrieval.

# 6. Techniques Addressing DVL Issues

After discussing some issues on a digital video library, in this section, we will examine the techniques on those issues in more detail. There are some techniques on the aspects of text descriptions of video, speech analysis, image analysis and natural language processing.

Here is the overview of digital video library system:




Overview of the Informedia Digital Video Library System [Informedia CMU]

In Figure 1, it shows the procedures to establish a digital video library. Initially, there are raw materials of videotapes with audio and video part. By using speech recognition and natural language processing technologies, generates a corresponding text transcript of each of the video file automatically. In addition to the generated text scripts, there may be some

other information given. Composing these sub-products, the part of text indexing is completed. With the combination use of audio and image analysis techniques, the segmentation and "paragraphing" of compressed video clips can be done. The whole indexed video database is then built. The creation part of the database is offline. It is just like the printing and binding procedure of book publishing. On the other hand, exploration and retrieval of library resources is in real-time. User makes a textual query or spoken query. The speech recognition is used in the user interface. The natural language analysis technique is used for the searching part. User can online either watch the returned video segment he/she wants or store it.

More details for the techniques in digital video library will be mentioned below.

## 6.1. Text description of Video

Text description of video is important for retrieval and user's searching. It can be the title of a video, or a brief outline of the video. Close-caption recorders and OCR technology can be used to convert this information into an electronic text representation, suitable for processing and augmentation by the other techniques described in this section. Even if no other automation techniques are used, a human indexer would produce more accurate and complete text indices, or tags, for the video if given this supplemental information rather than just a title or nothing at all.

Given a large body of text, it can be accumulated to describe the video contents, it needs to be structured in at least three ways:

- the text needs to be associated with the video it describes; a title will describe a large chunk of video, while text from a close-caption will describe only a few seconds' worth. By associating the text annotations closely with the video, they can be used to retrieve more precise, shorter duration video clips, as well as being used to build clips matching the user's needs more closely.

- it needs to be kept in separate fields, i.e., its semantics of origin must be preserved, so that a user interested in filtering out production notes information and looking only at titles can do so, or a visualization technique that lets the user browse according to information in the close-caption track, production notes, or other attribute can do so. Preserving as much semantic information as possible also allows a query asking for Kevin Costner as director to distinguish videos where he is an actor, director, narrator, or host.

- it also needs to be layered, to support browsing and the user's needs. If users wish to quickly determine whether a result is likely to have promise, perhaps they only need to browse through the titles, or a text abstract. Perhaps a movie preview will suffice in letting the user decide whether he or she wants to retrieve the complete movie from the library. The Scatter/Gather paradigm [Rao95] uses titles and terms of importance in a cluster of documents as a high level interface the user can browse with before focusing in on particular documents.

## 6.2. Speech Analysis

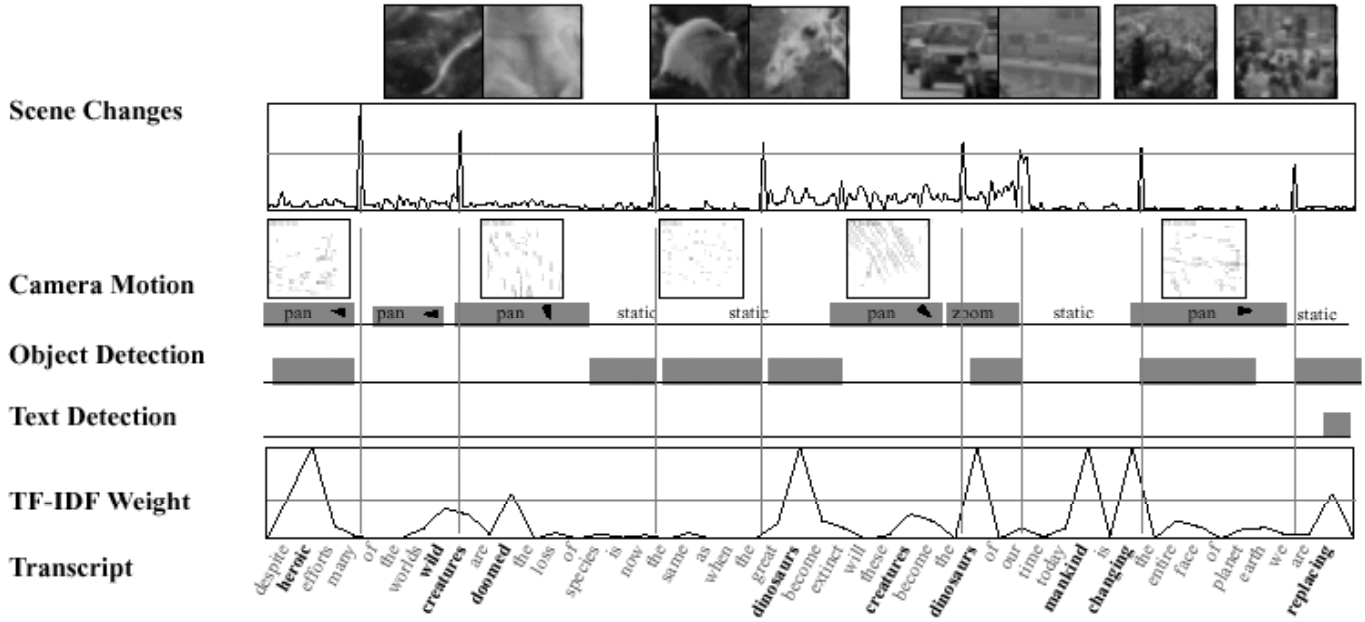
Much of the information conveyed in the audio for a given movie is captured in its close-caption text. Even though much of broadcast television is close-captioned, many other video and film assets are not. More importantly, typical video production generates 50 to 100 more data that is not broadcast and therefore not captioned. Clearly, effective use and reuse of all these video information assets within digital video libraries will require automatic generation of transcripts in order to make the information in the audio more accessible. Speech recognition technology can be applied to automatic transcript generation, but the accuracy of speech recognition is low. It still needs more time to improve the accuracy.

The audio conveys other information besides just dialog. Researchers have made progress in identifying pauses and silence [Arons93], as well as specialized audio parsers for music, laughter, and other highly distinct acoustic phenomena [Hawley93]. This information can supplement the other structured descriptors, and some such as pauses may be especially useful to identify natural start and stop times for video paragraphing as well as allowing for a degree of compression in presenting a “skim” video.

## 6.3. Image Analysis

Image analysis is primarily used for the identification of breaks between scenes and the identification of a single static frame icon that is representative of a scene. Image analysis can be mainly use in the segmentation of video. Identifying camera pans and zooms, edit effects like fades, cuts, and dissolves, can be useful for segmenting, or “paragraphing”, the video into a group of frames when video library is formed. Each group can be reasonably abstracted by a representative frame.

Video is segmented into scenes through the use of comparative difference measures [Zhang93]. Images with small histogram disparity are considered to be relatively equivalent. By detecting significant changes in the weighted colour histogram of each successive frame, image sequences can be separated into individual scenes. With image analysis, it can interpret camera motion which is one important method for video segmentation and description [Akutsu94].




**Video Characterization: keywords, scene breaks, camera motion, significant objects (face and text)**

However, a more efficient digital video library needs content-based video paragraphing methods, and image analysis by itself cannot determine all of the information. Some information system developers parse video in a particular domain, such as news footage, to supplement the image analysis with more structure and semantics, while others use human indexers to document video content, including space, time, weather, characters, objects, character actions, object actions, relative position, screen position, and cinematography. The digital video library user is interested in subject or content retrieval, not just "image" retrieval. The subject consists of both image content and textual content (from audio and other sources); the combination specifies the content. Any textual information attached is useful to quickly filter video segments locating potential items of interest. But subsequent query is usually visual, referring to image content. For example, "Find video with similar scenery," "Find the same scene with different camera motion," "Find video with the same person," and so on. Although part of the capability can be realized by content-free methods, such as histogram comparison, it is a long-term challenge to this field of computer vision research on the real solutions in content-based image search.

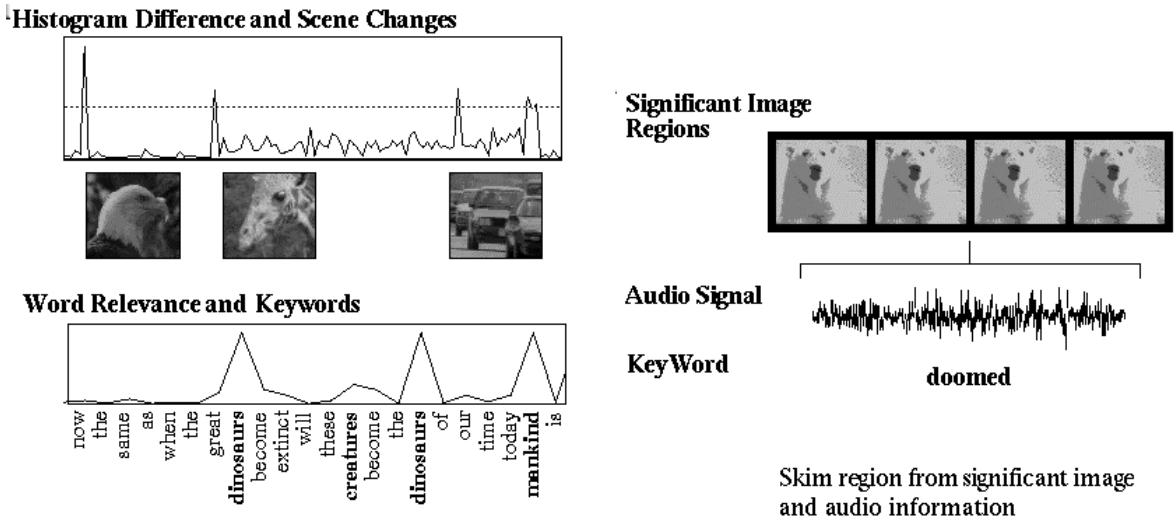
## 6.4. Natural Language Processing

Natural language queries allow straightforward description of the subject matter of the material desired. An initial query may be textual, entered either through the keyboard, mouse, or spoken words entered via microphone and recognized by the system. Subsequent refinements of the query, or new, related queries may relate to visual attributes such as: "find me scenes with similar visual backgrounds." Current retrieval technology works well on textual material from newspapers, electronic archives and other sources of

grammatically correct and properly spelled written content. However, the video retrieval task, based upon searching errorful transcripts of spoken language, challenges the state of the art. Even understanding a perfect transcription of the audio would be too complicated for current natural language technology.

There are several ways for natural language processing to improve the utility of a digital video library:

- Query processing: the user must be able to specify a subject or content area for search without having to resort to specialized syntax or complicated command forms.
- Retrieval: once the system has digested a user query, the corresponding text objects must be located, scored, and ranked according to user interest.
- Display: the video segments associated with each relevant text object must be located, and appropriate scene boundaries identified for each video object (visual sentence, paragraph or page) used to generate a menu of visual segments for user selection. [Informedia CMU]




Combined technology to segment video library data

It is a figure showing the integration of technologies used in digital video library [Informedia CMU]. There is a raw digitized video tape. The audio portion is fed through the speech analysis routines, which produces a transcript of the spoken text, such as the word “doomed” in the above figure. The audio signal is also analyzed for low energy sections that indicate acoustic “paragraphs” through silence. This is the first pass at segmentation. If a close-caption transcript is available, it is better to use it instead of the errorful speech recognition output. The transcript is processed by the natural language system and important keywords are identified. Using the results returned from the image analysis, we can then match the acoustic paragraph to the nearest scene break. This gives us an appropriate video paragraph clip in response to a user’s

request. First, search for keywords from the user query in the recognition transcript. When a match is found, the surrounding video paragraph is returned. For the static icon representative of a video clip, it place most emphasis on the image data. The paragraph is determined by the transcript and keywords. Within the paragraph, the most prominent keywords (found by natural language processing) identify the most prominent scene. The scene boundaries are determined by the image analysis of colour histogram differences and optical flow analysis.

# 7. Considerations and Approach

## 7.1. Focus and Target

To build a product-quality Digital Video Library that contains all the features we mentioned so far will need years of work and a large working team. In order to define targets which is suitable for a Final Year Project, we have to divide the project into modules that can be implemented in stages. And we also have to focus on building some components that will be reusable in the future stages.

We define some essential parts for a Digital Video Library that we can focus on:

- Build a component for video playback
- Build a component for user query
- Build a component to serve the query

For video playback, there are various video file formats that we can choose from. In the current stage, we are using the MPEG1 file format. MPEG1 format is not the only choice, Apple QuickTime format and Real Video format are possible alternatives. Reasons for we to choose MPEG1 format are its reasonable compassion rate and video quality, and also the free license for encoding.

For the query part, both the user side and the server side, we are targeted to implement a full-text search for the transcripts of the videos. Although there are various searching methods, but full-text searching is the fundamental kind and useful in many applications, and is relatively easy to implement, while other kinds of queries can be added in later.

## 7.2. Equipment

Our Digital Video Library Server (DVLS) required relatively a large amount of space for video storage. We use a PC with static IP to build our DVLS, which makes the network programming easier. The PC is equipped with about 16Gb of harddisk, and more than 10Gb of the space is used to install a sample digital video library 'Informedia' from Carnegie Mellon University. When the amount of video in our library grow, the need for more space may be necessary.

To prepare the video files, we use the facilities in Multimedia Lab of CSE Department to convert VHS tape video to MPEG1 files. With the help of

hardware encoding card, the encoding can be done in real time, which is quite satisfactory.

## 7.3. Programming Environment

### 7.3.1. Platform

We are now developing our project using Microsoft Windows 98/NT platform. It is because up to now the Windows platform is still the most popular end-user OS systems. And, besides of that, Windows has better support for entertainment applications, so we got more convenience in developing the multimedia application.

But, instead of sticking to the Microsoft Windows, cross-platform operability also affecting our decision. There are a lot of advantages for the capability to move the application to another platform, especially when we consider the stability of Unix system and the raise of Linux system.

### 7.3.2. Programming Language

We'd take Java™ as the programming language tools for this project.

There are certain aspects of Java that makes it favorite for the project.

**Platform independence:**

Platform independence means that the software written in Java can be run on any machines which has a Java Virtual Machine, no matter that's a PC, Unix, Linux or a Macintosh. This is obviously an advantage that the possible user group becomes larger.

**Network ready**

As in the very beginning, Java is considered as a language to work over the network, therefore its support in networking is good. Various kind of network model can be implemented with the network classes provided in Java, and is relatively easier to develop the same thing for other programming languages. Using Java may benefit when we implement the client/server model of the DVL. In case, we can even implement the Client as a Java Applet, which can be accessed by the web browsers on the WWW.

**International appeal**

The Unicode character set is an integral part of Java, allowing students to learn about the issues of developing software for the international market.

**Classes for GUI and video playback**

There are classes in Java which support GUI building and, most important, the video playback function. The simple but useful API free the programmers

from low level programming details and can play more effort on the higher level system architecture etc.

Nothing is ever perfect, there are drawbacks of using Java as programming language also.

**Interpreted language**

Java is an interpreted language, programs written in Java won't be as fast as those compiled languages such as C/C++

**Immature**

Java is a young language, it's API is likely to be changed in the successive versions, which may lead to difficulties in maintain the program.

**Using of extended API**

The classes provide video playback function, Java Media Framework API, is not included in the standard Java Runtime Environment, nor a standard install of Java Development Kit (JDK). The extra package has to be downloaded and installed before running the client program.

### 7.3.3. Network API

We use classes in java.net package and java.rmi package to implement the network functionality of our DVL.

Classes in java.net is used for implementing networking applications. There are the socket classes for communication on the Internet or implement a Internet server. Some other classes are provided to make it convenient to use Universal Resource Locators (URLs) to retrieve data on the Internet.

Package java.rmi provide methods for Remote Method Invocation (RMI). It is a mechanism that enables an object on one Java virtual machine to invoke methods on an object in another Java virtual machine. When such an object is invoked, its arguments are ``marshalled'' and sent from the local virtual machine to the remote one, where the arguments are ``unmarshalled.'' When the method terminates, the results are marshalled from the remote machine and sent to the caller's virtual machine. If the method invocation results in an exception being thrown, the exception is indicated to caller.

In our DVL, these two packages are used to handle the client requests for queries and the server responds.

## 7.4. Presenting Time-Base Media

### 7.4.1. Java Media Framework API

For the client program of a Digital Video Library, the core part is to implement the playback the media data. The Java Media Framework (JMF) is an Application Programming Interface (API) for incorporating media data types into Java applications and applets. It is specifically designed to take advantage of Java platform features.

The version of JMF API we are using is JMF 2.0 API. It provides a platform-neutral multimedia solution that runs on Java platforms that support JDK 1.1.5 or later. Here are some of its features:

- Present time-based media in Java programs
- Support for capturing and storing media data
- Control the type of processing that is performed during playback
- Perform custom processing on media data streams

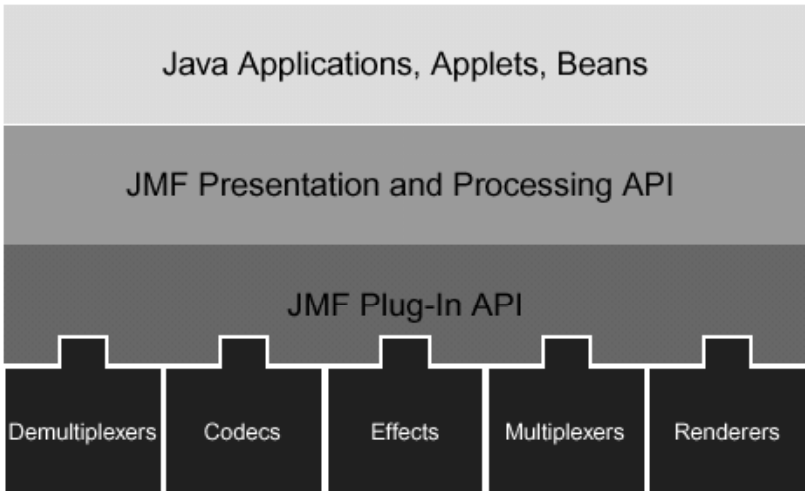
The package javax.media and javax.media.beam.playerbean is most useful for playback of video files. The classes in javax.media.rtp, javax.media.rtp.event, and javax.media.rtp.rtcp provide support for RTP (Real-Time Transport Protocol). RTP enables the transmission and reception of real-time media streams across the network. RTP can be used for media-on-demand applications which may be quite useful in the future versions of the project.

### 7.4.2. High-Level Architecture

JMF uses the traditional model for recording, processing, and presenting time-base media. It is quite the same as playing a movie using a VCR:

you provide the media stream to the VCR by inserting a video tape. The VCR reads and interprets the data on the tape and sends appropriate signals to your television and speakers.

In JMF, a *data source* encapsulates the media stream much like a video tape and a *player* provides processing and control mechanisms similar to a VCR. *Data sources* and *players* are integral parts of JMF's high-level API for managing the capture, presentation, and processing of time-based media.




High level JMF Architecture

### 7.4.3. Data Sources

JMF media players usually use DataSources to manage the transfer of media-content. A DataSource encapsulates both the location of media and the protocol and software used to deliver the media. As media data can be obtained from a variety of sources, such as local or network files and live broadcasts. JMF data sources can be categorized according to how data transfer is initiated:

- Pull Data-Source - the client initiates the data transfer and controls the flow of data from pull data-sources. Established protocols for this type of data include Hypertext Transfer Protocol (HTTP) and FILE.

- Push Data-Source - the server initiates the data transfer and controls the flow of data from a push data-source. Push data-sources include broadcast media, multicast media, and video-on-demand (VOD). For broadcast data, one protocol is the Real-time Transport Protocol (RTP), under development by the Internet Engineering Task Force (IETF). The MediaBase protocol developed by SGI is one protocol used for VOD.

The degree of control that a client program can extend to the user depends on the type of data source being presented. For example, an MPEG file can be repositioned and a client program could allow the user to replay the video clip or seek to a new position in the video. In contrast, broadcast media is under server control and cannot be repositioned. Some VOD protocols might support limited user control, for example, a client program might be able to allow the user to seek to a new position, but not fast for ward or rewind.

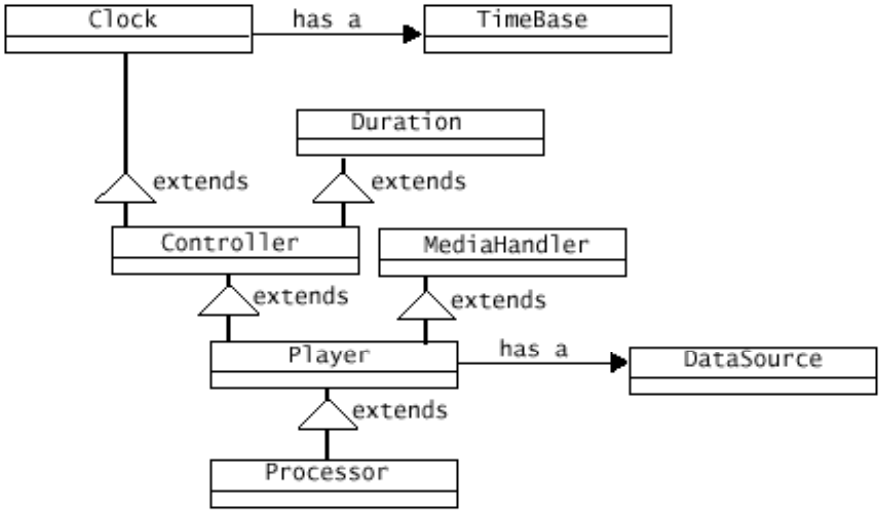
### 7.4.4. Data Formats

The exact media format of an object is represented by a Format object. The format itself carries no encoding-specific parameters or global timing information, it describes the format's encoding name and the type of data the format requires.

An AudioFormat describes the attributes specific to an audio format, such as sample rate, bits per sample, and number of channels. A VideoFormat encapsulates information relevant to video data. Several formats derived from VideoFormat to describe the attributes of common video formats are:

- IndexedColorFormat
- RGBFormat
- YUVFormat
- JPEGFormat
- H261Format
- H263Format

### 7.4.5. Presentation




In JMF, the presentation process is modeled by the Controller interface. Controller defines the basic state and control mechanism for controls, presents, or captures time-based media. It defines the phases that a media controller goes through and provides a mechanism for controlling the transitions between those phases. A number of the operations that must be performed before media data can be presented can be time consuming, so JMF allows programmatic control over when they occur.

To present time-based media such as audio or video with JMF, we can use a Player which implements the Controller. Playback can be controlled programmatically, or by display a control-panel component that enables the user to control playback interactively.

A Player generally has two types of user interface components, a visual component and a control-panel component. A visual component is where a Player presents the visual representation of its media, if it has one. Even an audio Player might have a visual component, such as a waveform display or animated character; A control panel component allows the user to control the media presentation. For example, a Player might be associated with a set of buttons to start, stop, and pause the media stream, and with a slider control to

adjust the volume. Some Player implementations can display additional components, such as volume controls and download-progress bars.

When several media streams are to be play, a separate Player for each one will be used, to play them in sync, we can use one of the Player objects to control the operation of the others.

# 8. Design and Implementation

## 8.1. Working Schedule

| | |
|---|---|
| June,1999 | Study papers related to issues about DVL |
| July,1999 | Continue to study different aspects about DVL in the papers |
| | Attend meetings among the cooperated work groups |
| | Install a sample DVL 'Informedia' provided by CMU |
| August,1999 | Get familiar and try different functions in the 'Informedia' |
| | Do some investigation for the programming language use for implementation |
| September,1999 | Define our focus and target for the FYP |
| | Study how to prepare digital video files from analog source |
| | Study Java as the programming tools for the project |
| October,1999 | Prepare digital video clips from VHS video |
| | Implement a program to demo how to play a video file by Java |
| November,1999 | Prepare more video clips |
| | Prepare transcript files with synchronized timestamps |
| | Program improvement:<br>Display video with synchronized transcript<br>Accept and able to play video over internet protocol |

## 8.2. System Overview

Our Digital Video Library consist of two major components, the client and the server. From a users perspective, the client program is all that he will have interaction with, while the server should be transparent to the user. Therefore,

the client program will accept queries/commands from user, and play the video clips wanted; what the server have to handle is to process the queries and return the corresponding video clip to the clients.

## 8.3. System Consideration

### 8.3.1. User Perspective

All of the user's action is done on the client program, and other parts of the system are transparent to the user, so that the user would not need to take care of other system's details. The client program will be built on the Windows platform in the beginning, and we'll try to transport it to unix/linux platform later.

There are some qualities we want the client program to possess, which may guide us to better software product during the design phase:

- Easy to use
- Simple and clear interface
- Functions should be reachable with a few clicks
- Easy to configure
- Stable performance

### 8.3.2. Server Program

As the server does not have a GUI, and what's important for it is the stable performance, so we will be built on the Unix platform in this phase. And transport to Windows platform in later phase to test the interoperability.

Here is the qualities we want for the server program:

- Able to handle requests from multiple clients
- Robustness
- Stable performance

### 8.3.3. Network Issue

Transmitting MPEG1 video signals requires a large bandwidth (approximate 350k bit/s). Therefore we test the program mainly in the LAN environment. For playback of media files at remote sites, the video will be buffered before enough data have been received for continuous playback.

### 8.3.4. Video Collections

In the current stage, we use the VHS tape from ATV as the video source, which recorded a few hours of news. But when the video collection has to scale up, the copyright of video source has to be considered.

## 8.4. System Design

The system may be divided into three main parts, they are the client, server and the library database.

### 8.4.1. Modules

**Client**

The client program is where the end users can interact with the video library, including input the query and watch the video playback.

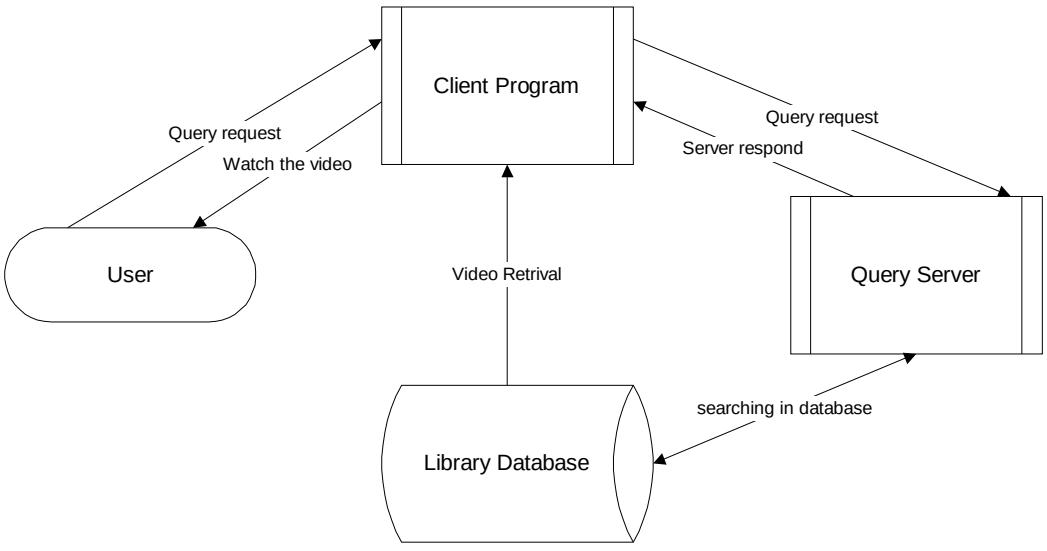
**Server**

The server is used to process the query and return the result to the client program. To increase the flexibility of the program, the server actually returns the URL to access the media files and the transcript files to the client.

**Library database**

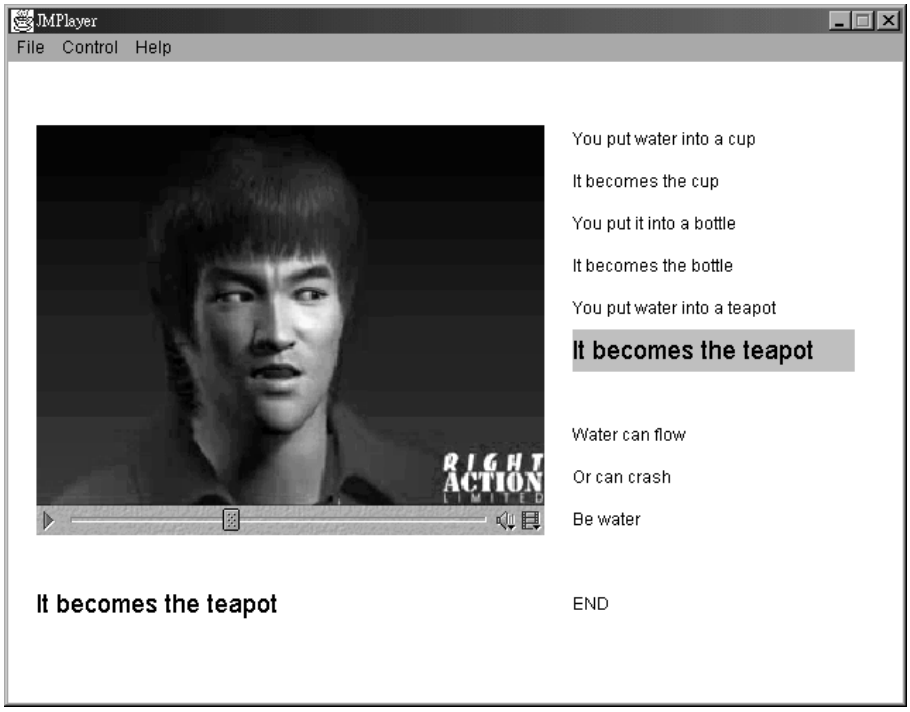
The library database will store the media files and the transcript files. Now they are stored as ordinary files under a web server, and can be accessed using the HTTP protocol.

### 8.4.2. Program Flow



## 8.5. System Implementation

### 8.5.1. Client Program




The client program's user interface is shown in the screen shot. The menu bar provides access to various functions, while the main window panel is use for playback. There is a video player on the left, and a rolling transcript on the right.

The core part of the client program is the video player. It is implemented using the Java Media Framework API. User may use the "File" menu to open a video file, or input a query in the dialog. And user may control the playback of video using commands under the "Control" menu.

### 8.5.2. Server Program

We use classes in the standard Java package java.rmi to establish the network connection. The main function of the server is to process the user's query and return the appropriate results. In the first step, the server will only process keyword/phrase for full-text searching, and then return URLs to access the videos which the keyword/phrase is present in the transcript.

### 8.5.3. Library Preparation

To build up the library, we have to prepare the video clips and transcript files. For the video clips, we use the equipment in Multimedia Lab, which includes VHS video recorder, and a PC with MPEG encoding card. The encoding takes place in real time, but preparing 10 hours of video still needs 10 hour of time, while for large scale libraries, more manpower and machine should be used for parallel production.

For the transcript files, now we prepare the transcript and also the timestamps inside, which is used for synchronized display with the video, by manual. This is definitely a time consuming job. Therefore we will build a tools for putting the timestamp into the text-transcript in a semi-auto way. This does not solve the problem if the library is large, but highly automated transcript preparing and indexing needs more advanced techniques, which should be added in successive stages later. Here are some techniques that maybe used for the automated transcript preparing and indexing:

- Speech recognition
- Scene detection
- Image processing
- Character recognition
- Video segmentation
- Nature language understanding

Up to now, there are nearly one hour video clips with transcripts, and both are in Chinese. In this one hour video, totally 38 video segments in the collections. All of them are news reports from ATV of Hong Kong.

News report is well structure. We can easily segment the whole daily news report into short clips with different topics of stories. Moreover, news report has less background noise compared with movie, TV programs, etc. There is mainly the reporter's speech with clear presentation. It is easily to transcribe the video manually.

# 9. Discussion

## 9.1. Problems Encountered

### 9.1.1. Hard To Define a Suitable Scope of Our Work

During the summer holiday, we studied the papers about different aspect of digital video library. As there are many different technologies involved in DVL, and many of them needs advance knowledge and deep understanding of the specialized field, so it is impossible to implement the full feature DVL by ourselves within one year.

After the semester began, we finally decided to develop a small scale DVL project, which started by implement a program that can play video files, and subsequently adding other functions, which is the program we now have.

### 9.1.2. Choosing Programming Language

Right after we had defined our working direction, we felt into trouble of the techniques needed for implementation. For playback of media data, MPEG1 files in specific, we only knew that we might write our own MPEG-decoder or use the DirectX API in the beginning. But after some investigation, we found that writing a MPEG-decoder by ourselves will involve too much work, which will be too low level and deviated from the high-level system design and so our project target. For DirectX API, it is powerful, but from the experience of other FYP teams, it takes a long time to study; also, it is platform dependent, which is not desired for the interoperability of the software. After more investigation, we found the Java Media Framework API and realized it's advantage over DirectX, including high-level of abstraction and ready to work over the network, which is very favorable for our project. And therefore we finally choose Java as the programming language for our project.

### 9.1.3. Preparing Video and Synchronized Transcripts

In this stage, we prepared the videos and synchronous transcripts of our database manually, which was a tedious and time-consuming process. Not only digitizing the videotape consume time, but also segmenting the video files into small pieces. But spending most of the process time was the preparation of text scripts. Since it is a full-content text script of video, we had to listen all the words spoken in the video carefully and typed them into the script files. Apart from that, timestamps will be added into the transcripts for synchronization. To finish all these pre-process on a two-minute video segment will spend more than half an hour.

## 9.2. Things Learned

We have learned the following things from our project:

- Different Issues of a Digital Video Library
- Building GUI with Java
- Building Applications with JMF API
- Digitizing Video

## 9.3. Future Plans

### 9.3.1. Semi-auto Timestamp Editing Tools

We decide to develop a tool to edit timestamps into text transcripts semi-automatically. Hope that it can reduce the manual work on building our video database.

The basic idea is that the tool has an interface showing a video on one side, and the corresponding transcripts on other side. During the playback of the video, the user can click a button to insert the current media timestamp into the current line in the transcript.

### 9.3.2. Searching and Indexing Capability

We have developed an interface for client side to show a video and the corresponding transcript. We will expand the work with the full text searching capability between client/server.

User can input keywords or phrases, all the videos with these keywords or phrases exist in the full-text transcript will be returned to the client. The client interface will show all the retrieved videos with their own representative images. User can click on an image, and view the video.

## 9.4. Possible Extensions

### 9.4.1. Automatically Derived Transcripts

Manually prepare the transcript of video is a time consuming job, which is not suitable for large scale DVL. Auto-transcribing should be implemented to reduce the manual work. Techniques of continuous speech recognizer, probably aided by the existence of auxiliary vocabularies from closed-captioning or available scripts, is needed to implement this feature,.

### 9.4.2. Language Processing of Queries and Transcripts

To further improve the accuracy of query results, linguistic indexing can be used to the derived transcript, generate lexicons, equivalence classes, and search indices. Techniques of natural language processing will be needed.

### 9.4.3. Content-based Image Manipulation

Image understanding plays a critical role for organizing, searching, and reusing digital video. Yet, the traditional database search by keywords, where images are only referenced, not directly searched for, is not appropriate or useful enough. Instead, digital video images themselves must be segmented, searched for, manipulated, and presented for similarity matching, parallel presentation, context sizing, and skimming, while preserving image content. For this extension, techniques including comprehensive image statistics, camera motion , object presence, object and scene understanding in 3D may be employed.

# 10. Conclusion

In this project, we have studied the user perspective, technology perspective, and system engineering perspective of implementation of digital video libraries. We also learn the JMF API as a tool to built multimedia applications. For our work, a small-scale digital video library is built, which can acted as a framework and test bed for later works. Our client program is able to play MPEG1 videos together synchronous transcripts in network environment, while over 30 digital video clips is prepared in our library. By the way, we do evaluate the performance of running multiple copies of client program. In our plan, we will continue to scale up the library, implement the processing of queries, as well as improve the method for indexing.

# 11. Related Works

## 11.1.Informedia™


The Informedia Digital Video Library project is a research initiative at Carnegie Mellon University funded by the NSF, DARPA, NASA and others that studies how multimedia digital libraries can be established and used. Informedia is building a multimedia library that will contain over one thousand hours of digital video, audio, images, and text. Informedia's digital video library is populated automatically encoding, segmenting, and indexing data. Research in the areas of speech recognition, image understanding, and natural language processing supports the automatic preparation of diverse media for full-content and knowledge-based search and retrieval.

## 11.2.The VISION Digital Video Library System

This is a research project at The University of Kansas. They have developed a prototypical digital video library system called VISION (Video Indexing for SearchIng Over Networks. VISION can digitize, compress, store, index, search, and retrieve video, audio, and closed-caption information. A client and server has been developed which can provide video information over the Internet or the evolving National Information Infrastructure

# 12. References


[Akutsu94] Akutsu, A. and Tonomura, Y. "Video Tomography: An efficient method for Camerawork Extraction and Motion Analysis," *Proc of ACM Multimedia '94* Oct. 15-20, 1994, San Francisco, CA, pp.349-356.

[Arons93] Arons, B. "SpeechSkimmer: Interactively Skimming Recorded Speech," *Proc. of ACM Symposium on User Interface Software and Technology (UIST)'9*3, Nov. 3-5, 1993, Atlanta, GA, pp. 187-196.

[Davis94] Davis, M. "Knowledge Representation for Video," *Proc. of AAAI '94*, 1994, Seattle, WA, pp. 120-127.

[Degen92] Degen, L., Mander, R., and Salomon, G. "Working with Audio: Integrating Personal Tape Recorders and Desktop Computers," *Proc. CHI '9*2, May 1992, Monterey, CA, pp. 413-418.

[Hawley93] Hawley, M. *Structure out of Sound*. Ph.D. Thesis, Massachusetts Institute of Technology, 1993.

[Informedia CMU] http://www.informedia.cs.cmu.edu/
Informedia Digital Video Project, Carnegie Mellon University

[Rao95] Rao, R., Pedersen, J., Hearst, M., Mackinlay, J., Card, S., Masinter, L., Hal-vorsen, P.-K., and Robertson, G., "Rich Interaction in the Digital Video Library," *Communications of the ACM* **3**8, April 1995, pp. 29-39.

[Samuelson93] Samuelson, P. "Copyright and Digital Libraries," Communications of the ACM 38, April 1995, pp.15-21, 110.

[Samuelson95] Samuelson, P. and Glushko, R.J. "Intellectual property rights in digital library and hypertext publishing systems. Harvard Journal of Law & Technology, 6, 237 (1993).

[Stevens94] Stevens, S., Christel, M., & Wactlar, H. Informedia: Improving Access to Digital Video. *interactions* **1** (October 1994), pp. 67-71.

[Zhang93] Zhang, H., Kankanhalli, A., and Smoliar, S. "Automatic partitioning of full-motion video," *Multimedia Systems* (1993) 1, pp. 10-28.

[Zhang95] Zhang, H., Tan, S., Smoliar, S., and Yihong, G. "Automatic parsing and indexing of news video," *Multimedia Systems* (1995) 2, pp. 256-266.

# 13. Acknowledgement

We would like to thank our supervisor, Prof. Michael Lyu, to give us valuable advice in our work.

We would also like to thank the following people who did give us a hand: Mr. Tim, Technical Staff, CSE Department, CUHK; Mr. Tony, Technical Staff, CSE Department, CUHK; Vincent Cheung, M.Phil Student, CSE Department, CUHK; Anson Lee, M.Phil Student, CSE Department, CUHK; Mole, Fellow Classmate, CSE Department, CUHK.

# 14. Appendix

## 14.1.JMF Example - PlayerApplet.

```
import java.applet.*;
import java.awt.*;
import java.net.*;
import javax.media.*;

public class PlayerApplet extends Applet implements
ControllerListener {

      Player player = null;

      public void init() {
            setLayout(new BorderLayout());
            String mediaFile = getParameter(“FILE”);
            try {
                  URL mediaURL =
                        new URL(getDocumentBase(), mediaFile);
                  player = Manager.createPlayer(mediaURL);
                  player.addControllerListener(this);
            }catch (Exception e) {
                  System.err.println("Got exception "+e);
            }
      }

      public void start() {
            player.start();
      }

      public void stop() {
            player.stop();
            player.deallocate();
      }

      public void destroy() {
            player.close();
      }

      public synchronized void controllerUpdate(ControllerEvent
event) {
            if (event instanceof RealizeCompleteEvent) {
                  Component comp;
                  if ((comp = player.getVisualComponent()) != null)
                        add ("Center", comp);
                  if ((comp = player.getControlPanelComponent())
                        != null)
                        add ("South", comp);
                  validate();
            }
      }
}
```