THEMATIC INDEXING IN VIDEO DATABASES

By

Shakeel Ahmed Khoja

B.Eng (Hons)

A thesis submitted for the degree of

Doctor of Philosophy

Department of Electronics and Computer Science,

University of Southampton,

United Kingdom

January 2001

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING

ELECTRONICS AND COMPUTER SCIENCE DEPARTMENT

Doctor of Philosophy

THEMATIC INDEXING IN VIDEO DATABASES

By Shakeel Ahmed Khoja

This thesis presents a novel video database system, called tv-DbMS, which caters for complex and long videos, such as documentaries and educational videos. As compared to relatively structured format videos like news or commercial advertisements, this database system has the capacity to work with long and relatively unstructured videos, by using the thematic indexing model.

Thematic indexing is a novel way to track a story in a video. A video contains many themes, which are implicitly related to each other. In order to resolve queries about particular scenes in a video, the scenes are stored hierarchically, which provides "is-a" or "have-part" relations between them.

In tv-DbMS video metadata is stored in a database. This metadata is used for indexing and querying the video. In this model, metadata contains information about segments (combinations of frames) and scenes (collections of segments that represent the same content). The metadata is organized in such a way that it can be used to navigate a theme (concept or idea). Annotations describing scenes are linked in a hierarchical manner to create a story line in the video, and thematic indexing is used to develop a video catalogue. Thematic indexing support the processing of complex queries in a video database and tracking story lines. Integrating open hypermedia capabilities, providing facilities for navigation, further enhances the tv-DbMS model.

The model has been tested on documentaries made for television by the Earl Mountbatten of Burma. The results show that a user can easily query for natural scenes or events. The model is also tested on an educational video namely the inaugural lecture of Prof. W. Hall. The results show how queries on abstract topics discussed in the lecture, can be handled by using thematic indexing.

The database system has been designed to ensure MPEG-7, and RDF compatibility. The metadata and other information are kept in such a format that it could be easily converted as captions to Audio Video Objects (AVOs).

Table of Contents

List of Figures

List of Tables

Acknowledgements

Firstly, I'd like to thank my supervisor, Professor Wendy Hall, for her expert guidance, from the very beginning to the end of my research work and keeping me focused. Gratitude is also due to Dr. Paul Lewis for helping me out at several occasions.

Secondly, I'd like to thank Stephen Chan for encouraging me at every step and being a digital image processing guru and a long lasting friend; Rui Marinheiro, always helping me out in configuring my system and providing me any sort of software, Samhaa and Cora, for encouraging me. Azair Alam, and Naeem Khan for supporting me, during my stay at Southampton.

I would also like to acknowledge the support of Association of Commonwealth Universities, for providing me an opportunity to come to United Kingdom and to pursue higher education.

Finally, not but the least, I would love to thank my mother, my sisters and my fiancée Kanwal, always there for me, for making everything possible.

Chapter 1

Introduction

> *"A film is too intelligible, which is what makes it difficult to analyse. A film is difficult to explain because it is easy to understand"* (Christian Metz 1977)

Advancement in the computing industry is a global phenomenon and its effects are not confined to its economic impact alone, but have also opened the doors of infinite information for the general population. This phenomenon can be described as an information revolution, where knowledge is regarded as an asset, much like conventional material assets. With the arrival of new and rich information mediums like digital images and digital videos, carriers like satellites and fibre optics, protocols like internet and intranets, a vast amount of unstructured information has been made available. This information may be rich in its contents, and may be understandable by the human brain, but becomes useless, in the context of digital processing.

Films and videos are one of the examples of the above-mentioned unstructured information. Until now, videos were considered to be a medium of entertainment with some artistic values. Everyone knows the richness and intensity of the information contained in videos, but because of its unstructured contents it was impracticable to extract any information by using a computer. Recent advancements in digital image processing is making it possible to retrieve some information, but still the organisation of data and searching in videos is at an embryonic stage.

Video production is considered to be an art, but it is also, a very complex technological undertaking. This thesis is focused on the technological issues related to digital video its content based description, intelligently extracting the required knowledge, attaching metadata to describe contents providing indexing techniques to get the information efficiently, and above all, allowing the user to interact with video data.

A video is a continuous medium having both temporal and spatial properties. In the temporal aspect, it is divided into different shots, scenes, segments and frames. A shot is a video sequence recorded by a camera's uninterrupted operation. A scene is a continuous block of story telling either set in a single location or following a particular character. A segment is a sequence of frames regarded as a logical unit, whereas a frame is the complete still image of a video media.

The main idea of this thesis is to incorporate object oriented methodologies to design a video database, providing a user-friendly query module and novel indexing styles to support rapid access. For this reason a thematic indexing module has been developed which considers the semantics and hierarchies of the video data and returns useful data, in response to queries.

1.1 Motivation

The motivation behind this work is the inspiration of technological advancement in the field of multimedia computing technology, which now provides opportunities of fusion between traditional textual databases and modern large digital video files. Great advances have been made in the database field. Relational and object oriented databases, distributed and client-server databases, and large-scale data warehousing are among the most notable. Another inspiration comes from the usage of object-oriented technology, which enables multimedia objects to be handed very efficiently and effectively. It is difficult to give a good definition of a video database management system, due to the rapid change in the technology. Based on Grosky (1994), and Chang and Hsu (1992), a video database can be defined as a system that has repositories of video information

(alphanumeric, feature and video object), and that provides modules for interactive querying, video processing, inserting new video objects, and composing existing video objects. Elmagarmid and Jiang (1997) define a video database management system as a software system that manages a collection of video data and provides content-based access to users. Carrer, Ligresti and Little (1997) provided another feature of video database system, i.e. a video database system is an entity, which provides fast and efficient storage and retrieval of digital video across multiple application domains. By comparing all these definitions we can say that a video database management system should specifically focus on the volume, storage and retrieval capabilities, and tackle other features like HCI issues by supporting hypermedia applications and authoring systems.

1.2 History of Visual Information Systems

The history of visual information is much older than any written form of communication and has been considered more influential than any written script. The popularity of text is attributed to Johannes Guttenberg, who invented the printing press and made text a mass medium. Again in the late fifties, computers that were supposed to be number crunching machines started being applied to text processing. Over the next fifty years, the development of technology made it easier to store and process images and videos. However, even now in the world of cyber technology, dealing with text is much easier than dealing with video or even images. But the popularity of video as an information source is increasing, and many scientists and researchers are developing new and innovative techniques to work with digital video. One of the earliest multimedia information systems that could support video is Palenque (Wilson, 1988), which used optical disc based technology. In the early 1990s systems like Elastic Charles (Brondomo & Davenport, 1990), Learning Constellations (Segall, 1990), IMPACT (Ueda, 1991), VBTools (Otsuji, 1991) and Stratification system (Smith & Pincever, 1991) started supporting videos. Elastic Charles was a hypermedia electronic journal that mainly contained videos and allowed users to go back to the exact point that they stopped at in the original video segment after examining a particular video clip. In other words,

Elastic Charles was one of the first systems that provided a hypermedia access to videos. It also used the idea of micons (motion icons) e.g. a short sample of video sequence that plays in a loop and that appears and disappears temporally during the existence of the link it corresponded to. Commercial formats like Microsoft's AVI, Apple's QuickTime, MPEG, M-JPEG, DVI etc, are the standard formats for digital videos. Major database management systems like Oracle and Informix have started using video objects as BLOBS (binary large object blocks) and provide tools for incorporating them into their databases. These BLOBS are the raw and uncompressed binary format of videos, so users are only able to store very small videos with very low resolution in their databases. For example a 10 second video with a resolution of 320 x 240, with 256 colours and having 25 frames per second will be over 6 Mega bytes in size. For standard length movies which are usually 1.5 hours in length, the video file size can go up to 3240 Mega bytes. Due to this enormous volume compression methods such as DC (direct cosine) became common. MPEG is an example of DC compression's examples. A DC image is a spatially reduced version of a given image. The compression process first divides the original image into blocks of $n \times n$ pixels each and then computes the average value of pixels in each block, which corresponds to one pixel in the DC image. For the compressed video data, a sequence of DC images can be constructed directly from the compressed video sequence, which is called a DC sequence.

With the advent of the internet and the World Wide Web (WWW) (Berners-Lee, et. al, 1990; 1994; Powel, 1998), people started working on projects such as video conferencing (Acharya, et. al, 2000; Galvez & Newman, 1999), video on demand (Liu, et. al, 1997; Kalva, et. al, 1999), video streaming (Xingtech, 1998; Real, 1998; 1999) and online video libraries (Hauptmann & Witbrock, 1998; Kozuch, et. al, 1996; Picard, 1996), and the use of video data is growing immensely.

1.3 Thesis Outline

This thesis is about the integration of metadata and annotations with digital video, rather then image processing, content based processing or object detection from a video clip. The major part of this work is to develop a database model, which stores video in

small segments, deals with video annotation effectively and then allows the user to perform hierarchical and concept based queries with the aid of a novel style of indexing, which we call as thematic indexing. Thematic indexing allows the user to query themes or concepts rather then simple keyword search. The main aim of the thesis is to present a model for a system, known as tv-DbMS, that tries to fulfil the requirements of a complete video database management system that supports thematic indexing

Chapter 2 deals with an overview of current information systems and their associated data models. This chapter is divided into two parts. The first part deals with the different types of (video) database models such as into relational, object oriented, algebraic and hybrid models. The second part of the chapter deals with the query, retrieval and browsing options in current video systems. The chapter also tries to classify existing video information systems based on their particular applications i.e. information retrieval, video authoring and presentation, and video production.

Chapter 3 discusses in detail the problems related to visual semantics and various solutions. How to develop a video document and how to segment the video is also discussed. This chapter also discusses the different levels of semantics in video. It concludes with a comprehensive study about the different types of video indexing.

The chapter finishes with a discussion about metadata and annotations. It also discusses the importance and advantages of metadata in video database management systems. The new and evolving MPEG-4 and MPEG-7 standards are also discussed in this chapter, along with a description of Resource Description Framework, a metadata standard.

Chapter 4 describes in detail the model of tv-DbMS. The chapter starts with the development of the relational model tv-DbMS. Then it defines how object oriented entities like event, are inserted in the main relational model to make this model a hybrid. The final part of the chapter discusses the query management module of tv-DbMS.

Video indexing techniques, like thematic indexing are discussed in chapter 5 as well as how to trace an object with the help of video object tree. The chapter also discusses how to use a particular object in a query with the help of thematic indexing and retrieve effective results. The chapter concludes with a discussion of how to track the story line or story theme in a video by using thematic indexing.

Chapter 6 deals with experimentation. This chapter describes the results that can be achieved using tv-DbMS. For experimentation we have used the TV documentaries of Earl Mountbatten, the last viceroy of united India. This chapter also discuss the incorporation of hyperlinks in digital videos. Hypermedia approaches to the video content and their integration with tv-DbMS are also discussed.

Chapter 7 deals with the evaluation of tv-DbMS as applied to the video lecture. This is an effort to apply the tv-DbMS video model to educational videos, as compared to its application on documentaries, discussed in the previous chapter. The second part of the chapter describes the integration of metadata standards, such as RDF and Dublin Core, with the tv-DbMS video model

Chapter 8 presents future research directions and concludes the thesis.

Chapter 2

Video Data Modelling

In Existing Systems

Data modelling is required to present video data based on its characteristics, its content and the applications it is intended for. It plays a very important role in the system since all the functions, procedures and classes are dependent on it. Again all the features, outputs, storage spaces and processes are described at this stage.

Looking at the complexity of a video stream, the contents, and semantic structure of the video needs to be described in a data model. A generic video model should also support the following features:

- Multi-level video structure abstraction
- Spatial and temporal relationship support
- Metadata support

Initially video data models were based on the conventional relational database model suggested by Codd (1970) in the early 70's. Soon researchers started working on object oriented models and algebraic models. Some researchers also tried to combine these

concepts and came up with hybrid models. So, in this literature review, we first divide these data models into relational, object-based, algebraic and hybrid models:

2.1 Relational Models

These models have provided the foundation for video information systems and can be justifiably called the pioneers of databases. They are designed to support video information management for varied types of video applications and to facilitate video information sharing among applications. In these models, video data are seen as having three major features: structure, detail descriptions and composition information.

Another interesting feature which is commonly used in these database models is that the structure of the video document is usually represented by a hierarchical structure divided into units, sequences and scenes, and shots, and these hierarchies are usually heavily annotated i.e. in these type of models, some textual data (also known as signatures or metadata) about the contents of the video are embedded into the video data. For example the annotations for a typical television news clip will be date and time, headline text, news captions, etc. Usually an annotator is required to write the captions. But in some models dynamic text is also generated depending on scene detection algorithms and other semantics, generated from the spatial characteristics of the video. Examples include histogram colours, basic shapes, textures and other information generated from audio contents. The following section describe some examples of this type of model.

2.1.1 VidIO

In the Multimedia Research Group (MMRG), at the University of Southampton, Salam and Hall (1996) developed the video database system, VidIO (Video Information

Organiser) that takes account of the importance of different perspectives of different users in video retrieval.

The main feature of this system was the provision of support for the creation and maintenance of personalised video materials i.e. personal collections, video segments and video documents. Other features support the user for easy access to personalised video material. The database is maintained by storing both the original and the personalised video.

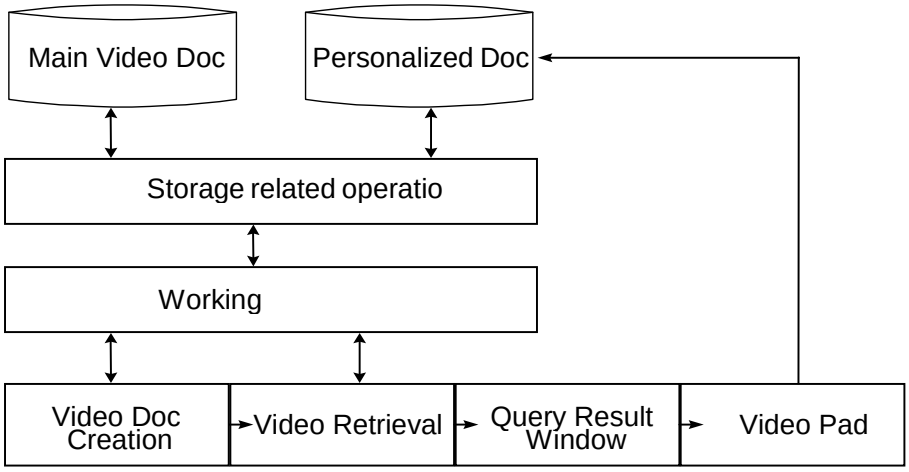


Figure 2.1: Block diagram of VidIO interface and storage

To provide easier user access, tools such as Query Result Window (for displaying user query results and other related information), Video Pad (used in creation, maintenance and re-indexing of personalised video segment, also supports segment browsing, file append and segment addition / deletion) and Document Creation (supports construction and maintenance of the original video database, and aids in organising personalised video segments) were also added to the system, as shown in figure 2.1. (Salam & Hall , 1996).

VidIO's data model is hierarchical and revolves around three data elements, e.g., structural components, contents in text form and the data catalogue. A video segment is divided into four main components: frames, shots, scenes and document. The hierarchy for a particular video is also resolved into components (Salam, 1997). The software's front end was developed in Visual C++(16 bit) and the back end database is Microsoft Access version 2.0. The videos are stored in AVI format *(frame size =240x180, frame rate/sec = 15.0, kilobits/sec data rate =190, compression mode = Microsoft Indeo32)*. The data used for testing this model were recorded news clips of CNN International.

During the creation of personalised documents, users are required to add their own descriptions, which are stored in a hierarchical manner, as shown in figure 2.1 above, maintaining the linear story structures of the original video materials, as they have contextual information that should be preserved to help users to interpret the data during video retrieval. This can be seen for a part of news clip data, stored in hierarchical form, in figure 2.2. At level 1 is the main review section. Then at level 2 are the sub-sections like Arts & Culture, Political, Crime, etc. Further in level 3 are the subsections of Arts & Culture column like WW2 paintings, Ballet Dispute, Movies of the week, etc. In this way each frame of the news video is captioned with its inheritance and can be very easily retrieved from the main video file. Hence according to figure 2.2, the link for the *councillor's speech* part in the main video document will be:

news_reviews . arts&culture . ww2_booty_paintings . councillor's_speech

The major problem with VidIO is the lack of video description input. No tools were developed or incorporated for auto indexing of contents. Again, no provision was provided for the recognition of a particular object in a video scene or frame. Thirdly, major attention was given only to the video part of the data and no consideration was given to the audio support, which was emphasised by Merlino (1997), who suggests that audio information can play a major part in content indexing especially in videos such as news bulletins, archives or documentaries, where there is always an anchor person giving important information in the background. Merlino further suggests that some

textual footings and the presenter's voice can become excellent keys for content indexing.

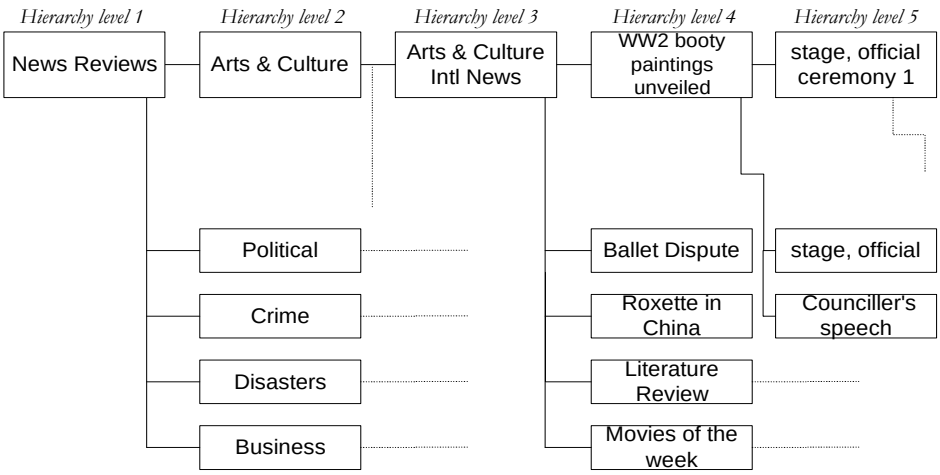


Figure 2.2 VidIO hierarchical description structure for an original video document

Along with this project, the MMRG at Southampton is actively involved in the MAVIS project (Lewis et. al, 1996). This is a programme of research to develop Multimedia Architectures for Video, Image and Sound. A modular approach is used and different modules are responsible for all the processing associated with a particular media-based feature type. As new feature types are introduced, associated matching techniques are developed and added to the main engine. For example, to make use of the added richness which digital video presents, modules are being developed which understand the temporal nature of the video and which can extract combined spatial and temporal features. Another feature of MAVIS is the development of a multimedia thesaurus (MMT) and intelligent agent support for content-based retrieval and navigation. It extended the Microcosm architecture (Fountain et. al, 1990; Hall et. al, 1996) to support the MMT, in which representations of objects in different media are maintained together with their inter-relationships. Intelligent agents have been developed to classify and cluster information from the MMT, as well as additional knowledge extracted during

information authoring and indexing, to seek out discriminators between object classes and also naturally-occurring groupings of media-based features and to accelerate media-based navigation (Lewis, et al, 1996).

2.1.2 Vane

The Multimedia Group at University of Boston has developed the video database Vane (Carrer et. al, 1997) with the property to encompass multiple applications or video document domains. Using SGML[1] and TCL/TK[2], they have developed a semi-automatic annotation process which extracts the content and semantic value of a raw video stream. However, this system requires a human annotator who would expedite the canonical information as metadata within the application domain.

The data model for Vane supports three types of indexing: structural, content based, and bibliographic. The structural metadata is again divided into media specific data and cinematography data. The media specific data indexes information about recording rate, compression format and frame resolution. The cinematic index is about the creation of video, specific information, title, date and camera motion. On the other hand, the content based information is again divided into two main categories information about tangible (physical shapes appearing in video) objects and information about conceptual (events, concepts, actions etc.) entities.

The developers of Vane support the view that three levels of hierarchy e.g. sequences, scenes and shots are enough for most straightforward generic decomposition. Additional layers would yield excessive fragmentation and excessive computation but do not provide significant knowledge (Carrer et. al, 1997; Ahanger et al, 1997).

In the model, a user can have multiple annotations of the same video segment by going a step ahead in the hierarchy but the implementation of this idea makes the model very

[1] Standard General Mark up Language
[2] Tool Command Language / Toolkit for X-windows

complex and it is not clearly stated that how these multiple annotations are stored in the data model.

In order to rectify this problem they have provided additional dynamic metadata definitions, as attributes to the unique identifier DTD (Data Type Definition). These are category definitions, attribute order and attribute definitions.

Overall, this tool was designed to construct large and useful video documents and was tested on some news archives with success. The tool is in its evolution stage and its usability on complex videos like documentaries or commercial programmes is not known. Secondly an annotator with prior domain knowledge is also required to identify the start and end of a segment, which is quite time consuming.

2.1.3 Informedia Digital Video Library

The CMU (Carnegie Mellon University) Speech group started the Informedia Digital Video Library project in 1994, with the aim to create a digital library of text, images, videos and audio data available for full content retrieval. By using technologies from the field of natural language understanding, image processing, speech recognition and video compression, they planned to provide users to explore multimedia data through interactive queries. The Informedia project allows hours long digital video to be segmented into logical pieces and to index them according to their raw contents (dialog, images, narration). Then the users can actively explore the information by finding pieces of interest to their search, rather than following someone else's path through the material (video on demand). This dynamic exploration, supported by a deep, rich library and the indexing / retrieval capabilities, provides opportunities for users to learn more from the system.

The architecture of the Informedia library is divided into search dialogue, text result, text retrieval, audio paragraph, audio playback, video filmstrip, video retrieval, video playback and video transcript modules (Christel et. al, 1994). In the first step, the videos

(which in their case were CNN News clips) are digitised, then a time-aligned transcript from the closed captions or through speech recognition is generated. After archiving the captions (i.e. annotations), segment story boundaries are created, then by segmentation (semi-automatic), scene breaks are also stored. In the final step indexing is done on these story segments and scene breaks. This whole data is stored in a relational database. In the search dialogue, users can initiate a textual search query on closed captions and also on the annotations, generated during the story segmentation and scene break process.

With regard to this thesis, the video retrieval module of Informedia is related to this research. In this module a user can retrieve the desired portion of a video clip, by querying the whole database for the CNN news video clips. The user can ask a query like "*Mr. Wu in China*" and the system will start looking for *Mr Wu* and *China* in the whole database. Then the system will show the starting frame of all the scenes that have matched this query e.g. scene clips of *Mr Wu* or, C*hina* or both (Steven et. al, 1994). Then the user can click on any frame to start viewing that scene. The user can also view the annotations and closed captions by clicking on the options.

2.1.3.1 Informedia II

By the late nineties, the same research group started working on Informedia II, with improved speed and accuracy of the underlying information exchange. The Informedia II database also included the interpretation of name, place, date and time references and features like dynamic story segmentation, speaker voice and face identification, video event characterisation and similarity matching (Christel, 1999). Funded by DARPA (Defence Advanced Research Project, USA), NSF (National Science Foundation, USA) and NASA (National Aeronautics and Science Administration, USA), this project aimed to digitise all the news clips and documentaries produced in the USA as part of a daily routine (i.e. one terabyte of daily news produced on American public television, educational documentaries from Open University UK, QED communications, NASA, the Discovery channel, etc) (Christel & Martin, 1998). Multilingual video libraries, cross-

language search, indexing of continuously unstructured, unedited field video, auto summarisation and visualisation over multiple documents and libraries were among the new modules, which the researchers at CMU planned to insert in Informedia II.

Another interesting feature of Informedia II is the multilingual information module. In the multilingual information module, users are allowed to search video documents in multiple languages. This new system of speech recognition on foreign language (non-English) news broadcasts, segments it into stories and indexes the foreign data together with English news data from English language sources. This process is done in three steps. In the first step a foreign speech recogniser converts the closed captions into small phrases. In the second step, these foreign language phrases are translated into English (and vice-versa), and in the final step, English labels (keywords) are provided to foreign language stories to allow a user to perform queries in English on the foreign language footage.

Another important module in Informedia II is known as Experience-on-Demand (EOD), which allow users to capture a record of their activities discreetly and share them in collaborative settings spanning both time and space. For example, a user can record Global Positioning System (GPS) spatial information, and other sensory data into the system. The EOD environment synthesizes data from many EOD units into a "collective experience" – a global perspective of ongoing and archived personal experiences. Each constituent EOD unit captures and manages information from its unique point of view. This information is transferred to a central site where the integration of multiple points of view provides greater detail for decision-making and event reporting. The aim of this system is to use the EOD unit not only as a data collector, but also a data access device, interoperating with the other EOD units and allowing audio and video search and retrieval (Wactlar, 1999).

One of the applications of the Informedia II project is the "Visual Digests for News Video Libraries". This digest tackles the problem of retrieving too many video clips for a particular query. Once a large amount of video clips are retrieved, this digest lets users

browse the whole result space not in the traditional style of a segment list, but rather results are generated dynamically based on context, providing a synopsis of the data within a collection of segments. For example a query of "Clinton + Andrew + Johnson + impeachment" will produce a 2-D graph having four words Clinton, Andrew, Johnson and Impeachment. The results (video clips) retrieved are shown as small boxes between these words. A result matching two words would be plotted on a line between those two words, placed according to the relevance of each word for that result. A result matching all the four words will be plotted exactly in the centre of these words. (Christel, 1999). This sort of visual query can also be in the form of a 2-D graph, a timeline sheet, or in a map digest (Christel & Martin, 1998).

2.1.4 The VISION Digital Video Library

The VISION (Video Indexing for Searching over Networks) digital video library prototype was developed at the Telecommunications and Information Sciences Laboratory of the University of Kansas as a test bed for evaluating automatic and comprehensive mechanisms for library creation and content-based search and retrieval of video across networks with a wide range of bandwidths.

Gauch and Li (Guach et. al, 1996; Pua et. al, 1993) came up with the idea of an on-line digital video library, which supports content-based search. The major feature of this project is an integrated application, which supports video processing, information retrieval, speech extraction and word-spotting[3] features in real time.

First the full-motion video is captured in AVI format with JPEG compression, then segmented by using two-step algorithm based on video and audio contents. Then a closed caption decoder is used to extract textual information to index video clips. Finally, all the information is stored textually for content-based exploration of the video library over the network.

[3] Key word spotting through speech recognition

A novel algorithm is used to segment the video. First the video is broken into different shots by the traditional image-based segmentation methods. These shots are then post-processed to merge some contiguous segments back together. Analysing audio features extracted from speech signals such as endpoint detection and speaker identification does this.

The endpoint detection algorithm is based on the measurement of the audio signal short-time energy and zero-crossing rate and end of an utterance. The short-time energy function of speech is computed by splitting the speech signal into 'N' samples and computing the total squared value of the signal in each sample. Further the energy of the speech is generally greater than that of silence or background noise. A speech threshold is then determined by taking into account the silence energy and the peak energy. The zero-crossing rate is the measure of the number of times in a given time interval that the speech signal amplitude passes through a value of zero (Guoch, et. al, 1996).

CNN news clips (headline news footage) were used to test this prototype and a successful result was achieved, but some major problems were also incurred. The major problem was that the contiguous clips that were separated by video techniques such as zoom-in/out, fade-in/out, or pan-in/out were all separately segmented. This system also failed where there was no sound between two successive clips, as there is no other way to detect a change on the short-time energy of the audio signal. This system is still at the prototype stage and work is continuing to develop the system.

2.2 Object Oriented Data Models

Since video objects possess both temporal and spatial features, OODBMS (Object Oriented Database Management System) are considered to be the best option for defining multimedia models (Klas et al, 1990; Yeo & Yeung, 1997).

Gibbs defines an AV (audio-video) object as :

> "An AV value, v, is a finite sequence, v_i, of digital audio or digital video data elements" (Gibbs, S., et. al, 1993)

Further he suggests that:

> "Each AV value has a media data type governing the encoding and interpretation of its elements. The type of v determines r_v, the data rate of v."

These definitions clearly state that the media objects define themselves, their own data rate and other attributes, which is a likely behaviour of objects defined in object-oriented environments. Kanda and Tanaka (Kanda et. al, 1998) have also worked on this idea. They developed the Object Oriented Video Information Database (OVID) (Kanda et. al, 1998). Traditional databases are unable to support video data models using multimedia objects for the following three reasons. First, AV (Audio & Video) values often have a very long duration i.e. of minutes or hours. Secondly, it may not be possible to allow concurrent use of AV objects (as is the case of relational models). Finally system resources (buffers, processor cycles, bus bandwidth etc) can only be allotted in a limited quantity to a process to handle a tuple.

Another major problem with conventional databases is the concurrent access to AV data, as it requires explicit scheduling of different processes in the system, by different clients. Since AV sequences are also temporal in nature, the database system should also be responsible for co-ordinating the presentation of these sources to different clients at different times. Traditional databases usually solve this problem by providing a temporary copy of the particular tuple to each client and then saving the last modified copy to the master database. As the AV object is of a very large size (in our case, a single object can be greater than 500 Mega bytes!), the idea of providing individual copy to all the clients becomes impossible.

The above points convinced researchers of the usefulness of object-oriented data models, particularly for AV data models. A lot of work has been done in this regard, which is presented in the following sections.

2.2.1 Modelling language for video

Arthur Caetano and Nuno Guimaraes (Caetano et. al, 1998) define shot as a sequence of one or more frames that are contiguously recorded and if sequenced represent a continuous action in space and time.

They have presented a modelling language for video, which enables the automatic segmentation, labelling and clustering of video data. These clusters of data generate a hierarchy of groups that can represent scenes as well as other types of semantic or logic groups to facilitate browsing and navigating the contents of the video. N-dimensional vectors represent low level features of the video shots. These vectors contain data such as frame rate, frame size, frame priority (for frames), colour, texture, shape (for image / shot) and other related camera operations like tilting, zooming and panning. These vectors (data sets) are stored in R-tree[4] and its variants (R*-tree, -tree) for adequate indexing.

The scenes, frames and shots are stored as Multimedia Objects (MO) which are in fact the basic units of information in their data model. The Multimedia Object is defined as

$$MO= (O_{id}, O_{type} ,F, R)$$

where O_{id} is the unique object identifier,
O_{type} shows the type of object and
F is the feature set associated with the ojbect i.e $F= \{f_i\}$ and $f_i = \{r_{i1}, \ldots\ldots, r_{in}\} \supset R$.

[4] A dynamic structure for spatial indexing

Each feature 'f*i*' is composed of one or more representations, for example, the colors of an object can be representated by histogram, moments, color vectors etc. These features aid in creating hierarchies of data at different levels. The query is then processed by retrieving the k-most similar objects to a set of feature vectors that designate the query's content. Work is also in progress on developing tools for creating algorithms on MPEG and MJPEG compressed video clips.

2.2.2 Gibbs & Breiteneder (1995)

Gibbs and Breiteneder (Gibbs et. al, 1992) have defined several abstractions that provide higher-level descriptions of AV values. The first is *Quality factor,* which takes care of compression ratio and frame size. The second abstraction is of *Temporal Entity Type,* which is specified in the form of **N[**T, {Q}, {D}**]** , where N is the name of the entity type, T is the underlying data type, Q is the quality factor (additional) and D is its discrete time co-ordinate system. For example in the expression

Video Entity[JPEG video Type, 480*640*8, D_{30}]

JPEG shows the compression format, 480*640 is the frame rate, 8 means 8 bits per colour band, and D_{30} shows the data rate of 30 frames per second.

Further, a number of relationships can be derived and instantiated with the above video entity, solving the problem of concurrency. Useful derivation relations can be of the type translation (the start times of a value are uniformly displaced), concatenation (a value is added to the end of another value), selection (a sub-value is extracted) and conversion (a value is re-sampled and used at some other location).

The object-oriented approach is also useful in request scheduling. For instance, read and write requests are scheduled by the concurrency control subsystem and disk accesses are

scheduled by the storage subsystem. With videos, these areas are particularly important and should be visible at the database interface and under the application control of the user. For example, it should be possible to request "play video X now" or "play video X after Y seconds" (Gibbs & Tsichritzis, 1995). This facility will not only aid in attractive features but will also help in scheduling resources and the database itself, thus providing concurrent access to media data.

Gibbs defines a class for categorising video objects as:

```
class VideoValue subclass-of Media Value
{       int width
        int height
        int depth
        int numFrame
        Image Value frame[numFrame]
}
class AudioValue subclass-of Media Value
{       int numChannel
        int depth
        int numSample
        int sample[numChannel][numSample]
}
```

The above classes provide very little information about media objects, i.e. an image is considered as a binary object, and the other features of images are ignored. Furthermore, annotation and signatures for images, scenes and sequences are also missing. These classes provide the basic structure of a video and were produced in the early 90's. Other researchers have made notable improvements in these classes in later years (Srinivasan et. al, 1997; Grosky, 1994).

2.2.3 Liusheng & Xiong (1995)

Xiong and Huang (Liusheng et. al, 1996; Xiong et. al, 1995) divide AV databases into two categories, depending on their functionality. The first category consists of databases that

concentrate on retrieving an image/frame according to one or more specific features. The second category of databases works mainly on retrieving requested information from a single image. In some systems, these two functionalities are combined together.

They further stated that the main problem with conventional OODB modelling is that a class has to be defined statically and objects of a class must be homogeneous in nature, which cannot be applied to AV objects which are inter-related, adhoc, tentative and evolving (Xiong, et. al, 1995).

A Conceptual Clustering Mechanism (CCM) is defined as C_t = <*A,M,X*>, where A is a set of cluster attributes, M is a set of cluster methods and X is the set of role-player associates. X is also the set of roles and objects that play the role within the cluster. Coming to the properties of CCM, they are dynamically created and deleted and can manage adhoc dynamics, but cannot create or delete objects (Xiong, et al, 1995).

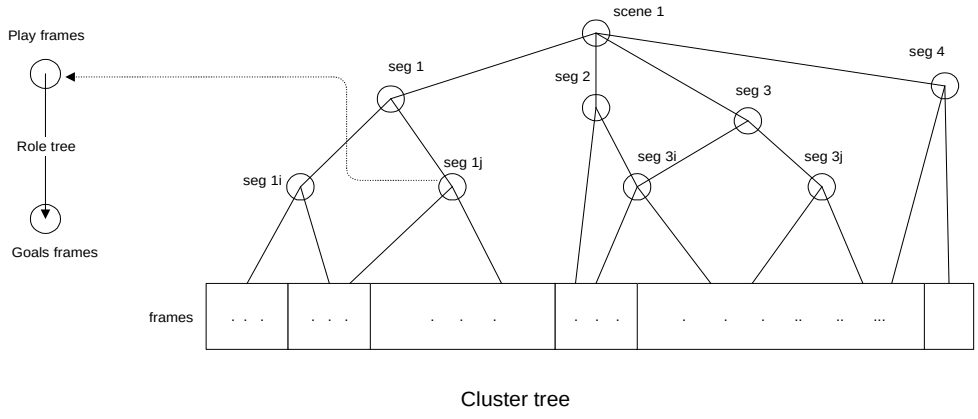


Figure 2.3 Example of a video programme, as illustrated by Liusheng and Xiong

In the indexing mode, video classification decomposes the data into semantically meaningful segments and CCMs, as shown in figure 2.3. It should be noted that it is not necessary for *segXi* to inherit all the properties of *segX,* but it does inherit some of the

basic properties from the parent node. In this way a cluster tree is formed, which is infact is a binary tree (at segment level) and can be easily used for indexing any video scene.

There are many issues not addressed in this model like storage. Also parsing of arbitrary arguments (messages) from one object to another is not very flexible. This model is still in its infancy and has not been tested on real applications.

2.2.4 Jain & Hampapur (MPIL)

Jain and Hampapur (1994) came up with their model "Multiple Perspective Interactive Video (MPIL)" model which can generate three dimensional *voxels* (information vectors) data sets from multiple streams of video data, which is manipulated to be refined and configured. The schema designer then chooses a set of primitives to model this data, which in fact is now a specific class definition. Subsequent objects similar to this data will become the instances of this class. A set of additional features is also assigned to these objects, which aid in indexing the class and its instances to use later in query operations.

The special attributes of AV objects introduced in this model are name (a string composed of two vectors, the first is the identification and the second vector defines the geometric properties of the primitive i.e. it can be of the format: *square.mesh.cylinder* or *simplex.mesh.cylinder*), parameters (define the external values set by the user at the time of data definition), create, delete and display. Other attributes like cut, copy, paste, deform, compare are also added for user facilities.

Common operations for a query in any AV model are union, difference and set-membership of point sets in feature space. More specific operations applicable in MPIL are Find Boundary (returning a hyper-polyhedral boundary of the points in image), Select by Spatial Constraint, Select by Distance and K-nearest neighbour.

Jain(1993) further states that all these classes, functions and operations merge to form a Video Data Modelling Language (VDML) which is destined to support description and manipulation for AV data. In addition they are also developing an implementation of a set of shape representations, stepping beyond the current paradigm of query-by-visual-example, describing the new research prototype of image-space visualisation environment (Kelly et. al., 1996).

2.3 Video Algebra

Weiss and Duda proposed an algebraic data model that defines a video stream by recursively applying a set of algebraic operations on the raw video segment (Weiss, et. al, 1995). The basic object of a video algebra model is presentation. A presentation is a multi-window, spatial, temporal and content combination of video segments. Presentations are described by video expressions, which are constructed from raw video segments using video algebraic operations. These operations are creation(create, delay), composition (concatenation, union, intersection, etc.), output (window) and description (content attributes that can be attached to video expressions).

Segments are specified using the name of the raw video and a range within the raw video. Creating compound video expressions from simpler ones using video algebra operations supports the recursive nature of the video structure. The model also allows nested stratification, i.e. overlapping logical video segments are used to provide multiple coexisting views and annotations for the same raw video data. Users can search video collections with queries by specifying the desired attributes of video expressions. The result of the query is a set of video expressions that can be played back, reused, or manipulated by a user. In addition to the content-based access, algebraic video also allows video browsing.

This model was tested on a collection of a pre-indexed TV broadcast news, commercials and movie trailers. Along with indexing, closed-captioned text (signatures) were also

provided to the Video Algebra System. The video and file server was run on Sun Sparc Station, the model was run on a Silicon Graphics machine and a third different machine with an HTTP server was also connected to them via Ethernet (Weiss et. al, 1994). Users were then logged on through the network and edited and composed the algebraic video nodes. The performance was found to be quite good. At the time of writing the system was still under construction and substantially commercialised tools like database support, auto indexing are in the process of development.

2.4 Hybrid Models

Hjesvold (1994a) introduced an architecture to support video information sharing and reuse, especially when running video presentations, annotating video materials and video information extraction. He incorporated two models to identify video shots, audio recordings, presentations and annotations. Firstly a content model for basic components in a video and secondly a structure model for composite components.

The content model is responsible for scene composition. It also defines attributes, operations and content descriptions for the basic video components. The structure model is used for scene editing. It also defines attributes, operations and semantics for the composite components. The attributes include object-identity, type, object architecture and a textual signature. Video Cassette Recorder (VCR) type operations such as playing, pausing, stopping, enhanced operations like creating and modifying components, and administrative operations like reading and changing the values of the attributes symbolic name, type, coding rate, quality factor, synopsis, etc. are also provided (Hjesvold et. al., 1995). Semantics for modelling a narrative presentation are supported by five story entities: scene, sequence, descriptive story, interlaced story and concurrent story.

This model is heavily dependent on metadata i.e. data about data at every semantic node and an experienced annotator is required to provide this data, but is a very good

example of how to share and re-use the same video among different video applications and users.

Smoliar and Zhang (1994) proposed a frame based (knowledge base) model that represents a hierarchical organisation of video nodes. A frame contains textual information for classes and instances. A class frame holds important information for maintaining the hierarchy such as super class, sub class and instance, while the instance frame holds a pointer to its upper level frame and the actual content representation, description, video and other related data.

Bryan-Kinns (2000) suggested a framework known as VCMF (A Framework for Video Content Modelling). This model depends on the semantic relations of segments in a digital video, and these segments can be re-used in different schemes. He has done the plot analysis of a classic Hollywood movie 'Citizen Kane' and has divided the story into major scenes and sub-scenes. Once the table of these scenes has been constructed, then a particular shot of video can be cross-referenced from any part of the video, providing an efficient way to reuse a particular clip of video.

Brayn-Kinns demonstrated his model by creating a video map so that a user can have a virtual journey through a digital video. In order to create the video map, the author records all the possible turns at the junctions. The video map is then constructed with the footage of the journey. Then the system checks the path used by the user and provides links for the next junction from that particular turn (Brayn-Kinns, 1999).

Yeung et. al. (1995) modelled video data with a hierarchical scene transition graph. The graph is a collection of scenes (or nodes) where each scene has a number of similar shots. Temporal information between scenes is used to construct the temporal structure for the video in the graph.

Simonnot (1995) developed a model, which consists of a hierarchical structure (i.e. documents, sequences, shots and frames) is used to model the structure of a video document. The document model takes account of different specialists' points of view to describe a video document or any part of the document structure. A specialist's point of view is called a facet. The facets that have been taken account of are general characteristics in a document, specific information from application domain specialists, and the contents of a document from a neutral observer's point of view, a user's point of view, an author's point of view, and an archivist's point of view. A document can have many facets and a facet might have a semantic network to help to keep accurate descriptions. Every description in a facet holds a weight that indicates its importance.

2.5 Summary

This chapter has presented and discussed the existing video database models and applications and their uses in different domains. The next chapter discusses techniques for video indexing, querying, filtering and retrieval. Some existing systems for supporting users with tools, such as video editing, authoring, presentation and metadata support are also discussed.

Chapter 3

Visual Information Retrieval

Video database techniques are not the only key factors for providing solutions to video indexing. Other key factors include video compression schemes, which allow efficient methods to store and transmit large archives of video data, video server system design and file system support, an issue that is critical to real-time, video on-demand server applications. Visual information retrieval, such as video segmentation, image processing, pattern recognition and media object detection are also important. These techniques analyse the pixel distribution and then extract the content descriptors automatically from the raw sensory data. Image content descriptors are commonly represented as feature vectors that are usually in multi-dimension feature space. Elmagarmid and Jiang (Elmagarmid & Jiang, 1997) describe four basic problems that need to be addressed in a video database management system. These are video data modelling, data storage and organisation, video data query and video data retrieval. The literature about the first two problems has been described in the previous chapter. This chapter deals with the third problem.

Visual information extraction is now entering a new era. First generation systems allowed access to videos through textual data i.e. keyword search only. Current generation retrieval systems support content-based retrieval by using visual contents and semantics. Access to video information is not only performed at a conceptual level, but also at a perceptual level, using visual contents.

3.1 Video Segmentation

The first step of any video processing system is to segment long chunks of video data into small segments (or streams). These segments are temporal in nature and are usually generated by template matching or histogram matching. Once segments are obtained, they are usually assigned some related information, known as annotations or metadata. These annotations can be manually edited or automatically generated. Manual annotations are more specific and depict higher level of intellect, as compared to automatically generated annotations, which are usually vision driven and provide information about texture, colour and region splitting in a particular frame (Ahanger & Little, 1997a). A segment can also have multiple descriptions, depending upon the semantics used in the segment. Section 3.1.1 describes how to use segments while presenting video data, Section 3.1.2 describes semantics used in segments and section 3.1.3 discusses video streaming, a new tool to transfer video data over the network.

3.1.1 Video Data Presentation using Segments

There are three types of techniques used to present video data while using segmentation techniques. First, video data can be presented as discrete segments with no established relationship among the segments. Second, video data can be pre-assembled e.g., video segments assembled for a particular video shot. Lastly, enough information about video segments are made available to the system to assemble data on the fly for delivery. These presentation are summarised below:

Independent Segment Presentation

This is a trivial presentation of information, in which information is presented in the form of discrete video segments, as shown in figure 3.1. Discrete segments are retrieved from computer's memory (e.g., "retrieve all the video clips that have Earl Mountbatten meeting with Gandhi"), and presented to the user. The relationships between various segments are not evaluated. In figure 3.1, a query has been performed about retrieving video clips of Mountbatten and Gandhi. The results of this query are the video segments, with the entities of Mountbatten and Gandhi, extracted from the video database.

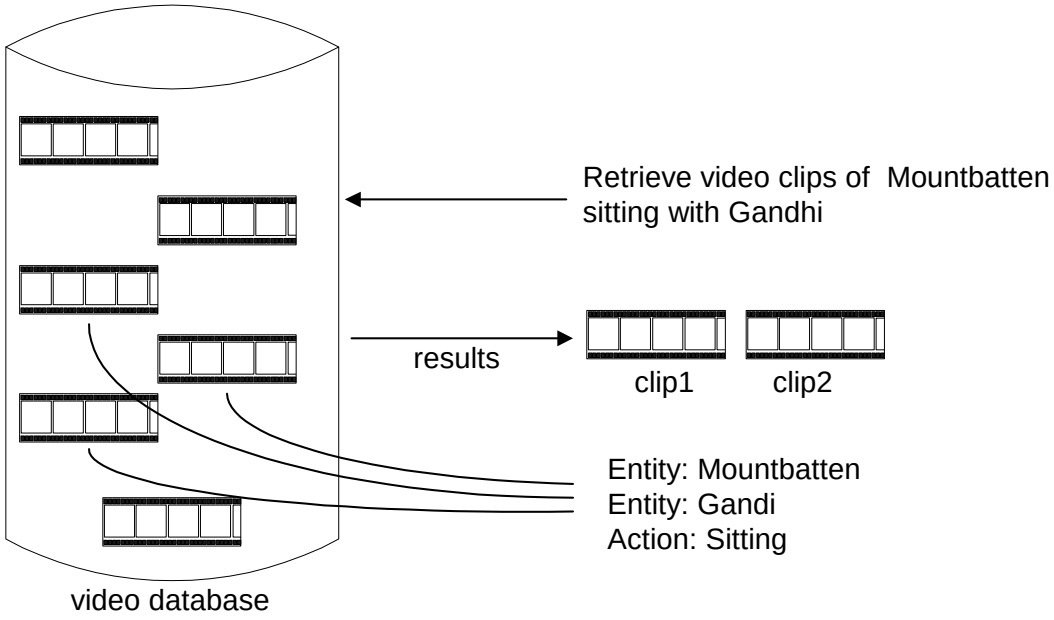


Figure 3.1: An example of independent segment based retrieval

Pre-assembled Presentation

In this presentation technique, the segments are pre-orchestrated (Ahanger & Little, 1997). In other words, the information about segments composing a presentation and

their order are stored as metadata. Figure 3.2 shows that the information about topics and the order they need to be presented is stored as metadata. Depending on the type of query, respective paths are traversed to retrieve information. For pre-assembled video there is little freedom for customisation or reorganisation.

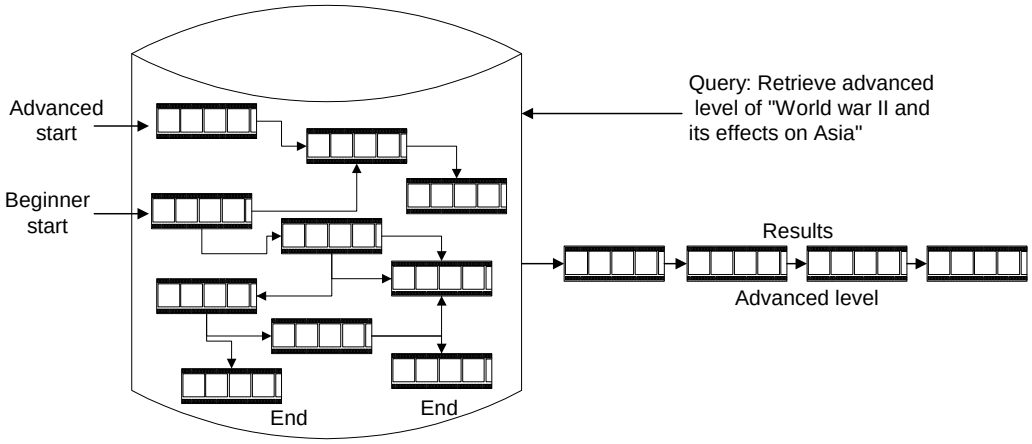


Figure 3.2: An example of a pre-assembled retrieval technique of World War II video clips

Dynamically Assembled Presentation

The sequencing of video segments in a presentation or a narrative is achieved in real time, as shown in figure 3.3. In a dynamic assembly, instead of having information about a particular description, information about the content in the narrative is stored. The information within individual video segments is used to compose and customise a narrative.

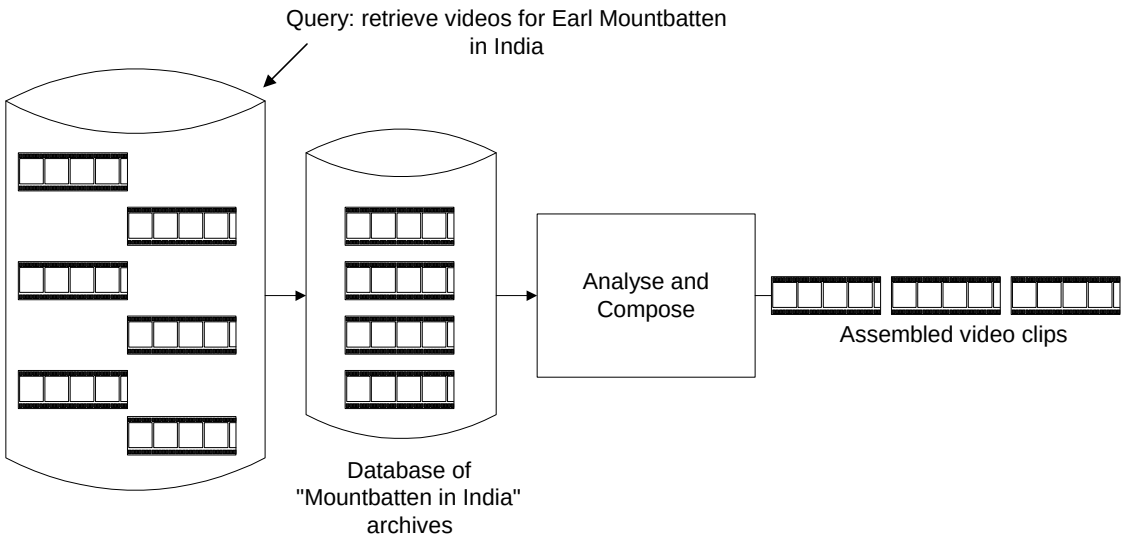


Figure 3.3: An example of a dynamically assembled retrieval for the documentaries of Earl Mountbatten (Chapter 6)

Once content is selected for assembly and a chain is formed, the content is mapped to a timeline called spatio-temporal mapping. The spatio-temporal mapping of the structures can belong to one of the following scenarios described below:

Scenario 1: Structures in the creation time reference are mapped to a timeline, as shown in figure 3.4. Video clips containing particular concepts or information are extracted from a recorded storage medium (e.g., tape, digital file) and are ordered in a sequential timeline. The clips are arranged in the order they were created.

Scenario 2: Structures across multiple references (tapes) of the creation timeline are mapped to a timeline. Multiple references can overlap in time, that is, more than one reference can have information from the same period on the creation timeline. In Figure 3.5, we show the references in two media overlapping.

Scenario 3: Structures in creation reference can be shuffled and mapped to the timeline. Once the structures are selected from a single or multiple time reference the structures can be shuffled in presentation time to satisfy a query.

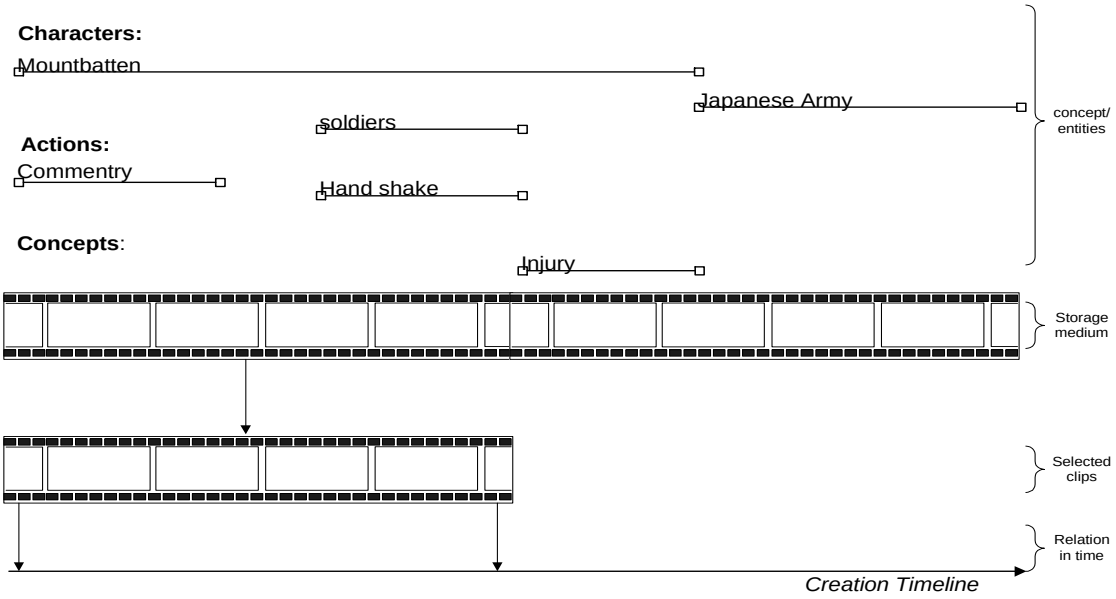


Figure 3.4: Spatio–temporal mapping in one time reference

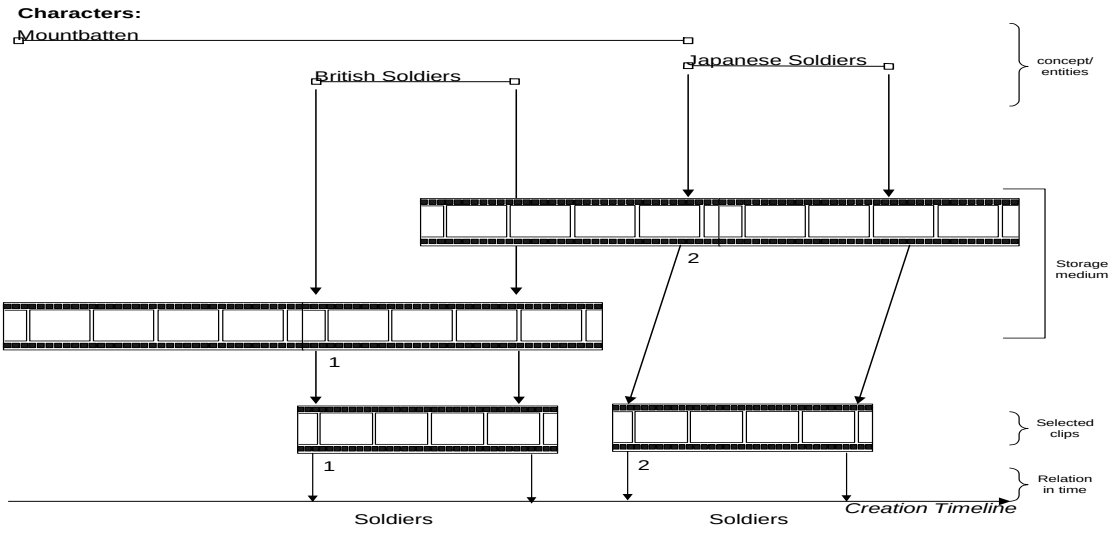


Figure 3.5: Spatio–temporal mapping achieved by structures from multiple references

Some techniques to achieve dynamic composition of video data have been accomplished. ConText (Davenport & Murtaugh, 1997) is a system for automatic temporal composition of a collection of video shots. It lets users navigate semi-randomly

through a collection of documentary scenes associated with a limited range of content metadata describing *character, time, location* and *theme.* The next scene shown to the user is determined by a scoring of all the available scenes. This scoring aims to obtain the preferred continuity and progression of detail in the presentation. This is made possible by establishing a present context consisting of metadata found in already played shots or shots chosen by the user. Each metadata entry is associated with a relevance score. The theme, or story line, is maintained by human intervention and is not completely automated.

ConText demonstrates how cognitive annotations of video material can be used to individualise a viewing session by creating an entirely new version through context-driven concatenation. This dynamic reconstruction can include video material made in a totally different context, thus performing a re-purposing of the material. The temporal ordering in a composition is maintained by scoring the weights given to the keywords representing different types of information.

AUTEUR (Nack & Parkes, 1997) is an application that is used to automatically generate humorous video sequences from arbitrary video material. The composition is based on the content describing the characters, actions, moods and locations; as well as the information about the position of the camera with respect to a character, such as, close-up, medium, and long-range shots. Content-based rules are used to compose shots.

Oomoto and Tanaka (Oomoto & Tanaka, 1993) use the concept of video object and a video model for segmenting a video. The model consists of hierarchical composition of video based on content or descriptive information associated with the clips. A video object (or segment) is described by a set of attributes/value pairs; where a value for an attribute can be a text description or another video object and an attribute can be created dynamically according to user needs.

In the Stratification System (Smith & Davenport, 1992; Smith & Pincever, 1991), the segmentation task is divided into two types. First, segmentation during a logging

process: this is a one-time process where users define a segment, and assign a free text description purposes. Results from this process are intended to represent the original content of a video where the free text descriptions describe a linear story of the video and the keywords provide contextual information among the frames in the video. Second, segmentation during the editing process: in this process, users can still define a segment and assign a number of keywords to add new context descriptions suitable for editing purposes. As a result, the stratification technique allows a frame to have multiple context information consisting of original logging contexts and new contexts built-up through editing. The original linear story provided by the free text descriptions during the logging is preserved and remains untouched.

Hjesvold (1994a; 1994b; Hjesvold & Midstraum, 1995; Hjesvold et. al, 1995) in his video database model, segments a video according to structure information and also considers multiple free text annotations for video segments. This approach allows a video document to be segmented into compound units, sequences, scenes and shots having a strong hierarchical relation between them. A shot can hold the annotations of a person, location and object. Apart from that, it also allows users to describe any segment of interest without requiring the segment to be a part of any video document structure.

The above segmentation techniques rely only on content for composition. Structure and time are also critical elements in composition. A structure depicts cinematographic rules like establishing a starting scene, intermediate scenes, and a closing scene. Time maintains the temporal sequence of events, and it is important for presenting information in the correct time series. A news clip, for example, is a series of sub-events or a cause and effects chain in which the time series must be maintained. In this thesis all the three features i.e. content, structure and time domains are taken into consideration.

3.1.2 Semantics in Video Segments

Video syntax is highly determinative of the semantics, as discussed in the Kuleshov Effect (Levaco, 1974). The Kuleshov Effect is named after Lev Kuleshov, a Soviet

cinematographer whose work at the beginning of the century deeply influenced the Soviet montage school and all later Soviet cinemas. Kuleshov was himself an engineer who, after only having worked on one film, was honoured by heading the Soviet film school after the October Revolution. Kuleshov was fascinated by the ability of cinema to create artificial spaces and objects through montage (editing) by the virtue of the associations that people create when viewing sequences of shots, which if the shots were taken out of sequence would not be created. In the classic Kuleshov example, Kuleshov showed the following sequence to an audience:

the passive face of an actor - a bowl of soup - go to black

the same face of the actor - a coffin - go to black

the same face of the actor - a field of flowers - go to black.

Upon interviewing audience members and asking them what they saw, they said, "Oh, he was hungry, then he was sad, then he was happy". The exact same image of the actor's face was used in each of the three short sequences. Hence, the Kuleshov Effect showed that the semantics of video information are highly determined by what comes before and what comes after any given shot.

The syntax of video sequences determines the semantics of video data to such a degree that any attempts to create context-free semantic annotations for video must be carefully scrutinised so as to determine which components are context-dependent and which preserve their basic semantics through re-combination and re-purposing. Any indexing or representational scheme for the content of video information needs to be able to facilitate our understanding of how semantics of the video changes when it is re-sequenced into new syntactic structures. Therefore, the challenge is twofold: to develop a representation of those salient features of video which, when combined syntactically, create new meanings, and to represent those features which do not radically change when re-contextualised.

Video indexes obtained automatically through low-level image and video processing are not enough to match the semantic level of the user (i.e. human semantics). Colombo (Columbo et. al, 1999) suggests that the most common way to enrich a visual information retrieval system's semantics is to annotate pictorial information manually at storage time through a set of external keywords describing the pictorial content. But he also suggests that this sort of system can be very expensive and too time consuming.

Most researchers agree that manually adding metadata (annotations) to the video segments is not the right solution (Cascia & Aridizzone, 1996 ; Pentland, et. al, 1994; Piccard, 1995), but nobody is able to automatically extract the high level information from a digital video. Extracting low level image and camera movement information is easier and being done (Cascia & Ardizzone, 1996; Pentland, et. al, 1994; Piccard, 1995), but this information is not enough for the user to query a video database. Zhang and Smolier (Zhang, et al, 1995) present a system that extracts low-level information (eg. Detect cuts and camera motion) and allow the user to add high-level information (metadata) in an organized way. Picard and Davenport (Pentland, et. al, 1995 ; Picard, 1996) developed a system at MIT Media labs that can associate higher-level semantics with certain low-level features while digitising the video.

On the other hand, Devadiga (Devadiga et al, 1995) and Cascia (Cascia & Ardizzone, 1996) describe systems that perform segmentations only on low-level visual features (colour histogram, texture descriptions). They facilitate query by example (queries are posted against key-frames extracted from the video).

Hybrid solutions i.e. having a mixture of manually annotated data and some low level image and video processing seemed to be a better solution. For example Zhang (Zhang et. al, 1995) implements queries by example, by template manipulation and by keyword (all queries are also posed against extracted shots, described by key frames and additional attributes). Zhang planned to add audio analysis and text parsing to extract more high-level semantics from the video during automatic analysis.

3.2 Video Indexing and Retrieval

Once segmentation (with or without semantics) has been done, the next step is to store the segmentation data or segment annotations in a database and create indexes for quick and efficient retrieval. This section of thesis discusses existing techniques and trends for video indexing and retrieval. Some techniques apply a complex data model and structure, but others apply a simple data model such as a linear story structure of a video.

3.2.1 Video Indexing

Video indexing can be divided into manual, semi-automatic and real time indexing techniques. They are discussed below:

3.2.1.1 Manual Indexing Approaches

All existing techniques in this category use keywords or textual descriptions in describing the contents of a video.

Oomoto and Tanaka (Oomoto & Tanaka, 1993) defined two types of indexes: first a hierarchical index of video objects and secondly a dictionary that indexes all *is-a* attribute value definitions of video objects in a hierarchical form. To create the first index, a user has to define a video object (a segment), create / select an attribute and provide a value (of either a textual description or a pointer to another video object) for the attribute. A video object may also be created by applying composition operations such as merge and overlap. The second index is created/ updated by defining / choosing an attribute for a video object and entering / choosing a value for the attribute.

The Generic Video Data Model (Hjesvold & Midtstraum, 1994; Hjesvold et. al., 1995) stores videos, audios, virtual video documents, video document structures and video segment annotations. VideoSTAR, the prototype system that is an implementation of this data model, provides content manipulation operations (such as operations for adding, deleting and modifying) for manually indexing its contents. The indexing

process is undertaken by composing a virtual video document, defining the structure of a video document and annotating video segments.

Weiss (Weiss et. al, 1995) indexed the video data by using as algebraic video data model. He used video expressions to index the data. To index a video, a user has to create segments, and add descriptions to the segments. Descriptions for a segment can be in textual or non-textual form such as icons or image features (colour shape or texture).

3.2.1.2 Semi-automatic Indexing Approaches

Content based processing is usually done by using motion vectors in a compressed video file. This is the case in research work of Akutsu (Akutsu et. al, 1992) to create a video index. Basically, video data was modelled as having shots and camera operations. The structure of shots in a video was considered as a natural index for the video. A block matching method was used to calculate the motion vectors required. To create the index, motion vector values were used to segment a video into shots and to estimate camera operation of a shot. Seven vector characteristics like fixed camera, panning, tracking, tilting, booming, zooming and camera dolly have been identified. This indexing method has been applied in a test system to provide a video index represented by three-dimensional icons visualising the camera operations of each shot.

Lee and Kao (Lee & Kao, 1993) introduced a semi-automatic video indexing procedure. The procedure was based on the existence of objects and their motions. To index a video, an indexing operator has to browse through the video and segment the video based on existence of objects. The results of this process are then used to calculate the tracks of objects. The information on the tracks of moving objects are then represented using 16 numeric motion symbols (north, north-east, east, south-east, south, south-west, west, north-west, close, away, clockwise, anti-clockwise, rotate-left, rotate-right, rotate-upward, rotate-downward) and stored in index files. A user can query the database by specifying objects and /or tracks.

Smolier and Zhang (Smolier & Zhang, 1994; Zhang et. al, 1995) have approached video indexing from two aspects: textual and visual features. The text –based index is a tree structure of *topical categories.* It uses a knowledge model called frame –based knowledge base. To create/ update this index, a user has to create/ update a number of knowledge-based-frames and create / update required attributes and attribute-values for each knowledge-based-frame. Specific attributes such as *Description* for a leaf knowledge-based-frame may hold indices in the form of textual descriptions.

The visual index is computed automatically during a video parsing process that performs an automatic segmentation, classifies camera movements and performs content parsing. The index consists of: 1) representative-frame (or key frame) based indices that hold representative–frame characteristics such as colour histograms, 2) object-based indices that hold object information such as location, colour and size and 3) semantic indices that hold information that has been computed from representative-frame and object-based indices for one or a group of shots. The semantic–based index (also called the event–based index) is created automatically by applying motion analysis to index camera movement, and frame-to-frame differences to index video contents with the assistance of a priori model of its spatial and temporal structure.

The indexing process developed by Swanberg (Swanberg et. al, 1993) involves automatic segmentation of domain objects from raw data into shots, typed shots and episodes, the calculation of descriptive features of those objects, and the storage of objects and features into the database.

3.2.1.3 Real time (capture-time) video indexing

Image Miner is a system that has been developed at the University of Germany (Alshuth et. al, 1997), which, while capturing the video fragments the shots into clusters, based on their visual similarity. A time-constraint clustering procedure is used to compare only those shots that are positioned inside a time range. This cluster information which

contains a list of shots and their matched-clusters, makes it possible to calculate scene bounds. A labelling of all the clusters is generated and indexed into a file. The Image Miner system consists of three analysis modules for content processing, e.g. colour, texture and contour analysis. This is all done real time by capturing the previous frames in a buffer and comparing it with the current frame.

Carriera (Carriera et. al, 1995) has introduced a capture-time indexing technique that has been realised through the use of Segment Definition Files (SDF). The technique has been designed and implemented for an authoring tool called Video Broadcast Authoring Tool. It requires an index for a video to be created at its capture-time (for a new video) or its creation-time. The SDF is an ASCII text file that is formatted to store segment information in a hierarchical structure. An initial index has one header section, a segment section where at least each segment has a start-frame number, and all segments are assumed as root segments initially (at the top hierarchy). An automatic segmentation tool called Motion Picture Parser (Deardorff et. al, 1994) has been integrated with the test system to provide an initial index for a video without a capture time index.

Yeung (1995) proposed a system that could index a video by applying an automatic segmentation method for compressed video, and clustering the extracted shots according to the similarity of visual contents such as colour and shape. The similarity measures have assumed that its representative frame represents a shot. The result of the clustering process is a hierarchy of shot clusters where each level has a group of shots that are very similar to each other. At this stage an indexing operator can verify the result by viewing and re-classifying. Then the result is used to automatically build a hierarchical scene transition graph. All shots in one level are grouped as one scene, represented by one representative frame and a scene is called a node in the graph. Temporal relationships between nodes in the hierarchy are preserved during the creation of the hierarchical scene transition graph.

Hampapur (1995) has developed a semi-automatic video indexing technique that emphasized the foundation of an application specific video data model. Basically, the indexing procedure is as follows:

- Define a complete video model for a specific video application based on motion features
- Apply automatic segmentation techniques to obtain shots
- Apply image motion based partial indexing techniques to extract partial video indexes
- Map the machine – derived model to the complete video model to obtain machine derived labels for each shot
- Apply an indexing operator, that can confirm the partial label or correct the label by choosing another label from a complete label set

In the second part of this section tools and techniques for retrieving or browsing videos are discussed. One of the earliest browsers was developed by Nagasaka and Tanaka (1992), known as Scene Browser, incorporating automatic segmentation techniques based on colour features to obtain shots for a video. All shots were then represented by moving icons (micons) and displayed in the scene browser.

3.2.2 Video Retrieval

3.2.2.1 Video Browsing

Otsuji (1991) in their system VBTools, used the five-frame comparison method to automatically segment a video based on grey-level features. The tool provides two browsing methods: 1) cut browsing, and 2) motion dependent browsing. For every shot detected, the cut browsing tool provides two methods: 1) shows the first frame of each shot five times, and 2) shows the first five frames of each shot. The motion dependent

browsing tool allows the production of a video summary that has more frames for a heavy motion scene and less frames for a quiet scene.

In OVID (Oomoto, & Tanaka, 1993), a user can browse the contents of a video database using a scrollable bar chart, called VideoChart. The horizontal axis of the chart represents frame numbers. Every video segment that exists within the specified frame range is represented by its object identifier and a bar showing its start- and end-frame. Each single segment is displayed on a new line. If a segment is composed of non-continuous segments, it is displayed on the same line but as a number of non-continuous bars. Any segment can be selected and used in performing other operations such as play, inspect, update, de/compose and query formulation.

Tonormura (1994) developed a series of browsing tools (Flash Browser, Stroboscopic Browser, Rush Browser) that use automatic segmentation methods. These tools have been developed, based on a model of three stages of how humans understand video (no idea, rough idea and full idea). The flash browser presents each shot representative frame in one window one after another. The stroboscopic browser also presents shot representative frames but in a smaller film-like window, more than one image at a time. The rush browser allows a user to play all the representative moving images from each shot to get more idea than the other two browsers.

Clipmap (Smoliar & Zhang, 1994) is a windows based browser, presenting a number of video clips in its micon form. Results from the automatic creation of the visual index can be displayed on the Clipmap. This provides an unstructured index for a video and direct access to each clip. The Clipmap was used to present results of news video parsing where each news item was presented as one clip icon. A user can select the icon and browse it using a browsing tool.

Video Parser (Smoliar & Zhang, 1994) allows a user to browse the contents of video clips and examine and manipulate their icons. It is divided into three parts: 1) a scrollable index strip that displays a representative image for each clip, 2) a soft video player that

plays a clip and displays its text description for any clip selected using the index strip, 3) a micon window that displays a micon and provides options for examining and manipulating a micon for any clip selected using the index strip.

Zhang (1995) applied an automatic segmentation procedure to obtain shots. Each shot is then abstracted semi-automatically with one or more representative frames (called key frames). A shot with moving objects or camera movements may have more key frames than a shot with no motion. The sequential access browsing tool is a video player that apart from supporting normal detail browsing with an option to stop at key frames. It also allows overview browsing by playing key frames at a suitable rate.

Yeung (1995) used the semi-automatic video indexing method to create an initial hierarchical scene transition graph for video browsing. The browser provide two ways for examining contents of a video: 1) a scene transition graph consisting of all scene nodes and temporal relationship arrows, and 2) a scrollable window for presenting all shots in the video in its linear structure form. A user can select a node from the graph to reveal shots represented by that node, and can also modify and re-arrange the graph to create a new story structure.

The Hierarchical video magnifier by Mills (1992; Smoliar & Zhang, 1994) allows a user to browse a video or a video sequence with six representative frames initially. The source is divided into six equal-length sequences where each sequence is represented by its mid-frame, has a scrollable icon for browsing its content and can be played by dragging the frame to a video player. When a user selects a representative frame, another hierarchy level is displayed that zooms in the content of the selected sequence into another six equal sequences. The user can continue magnifying the source for a number of times. Results of the hierarchical view can be labelled and stored for future reference.

Smoliar and Zhang also developed a hierarchical video magnifier that has the same conceptual idea as the magnifier by Mills. The difference is that it has only five windows

in a hierarchy level, and it has been provided to support browsing of a video without a well-defined narrative structure and also to support content-free content analysis.

Hjelsvold (Hjelsvold & Midtstraum, 1994; Hjelsvold & Midtstraum, 1995; Hjelsvold et. al, 1995) classified descriptions of video segments (called content annotations) into three types of contexts to allow a user to specify the kind of information the user would like to browse. The Video Document Browser works together with a video player. The video player is responsible for playing a video document and the Video Document Browser is responsible for showing the related content annotations and structure for the part of the document being played.

Three different contexts were derived depending on the relation between an annotation and the given document. An annotation is defined as having a primary, basic or secondary context with a given document. If the document has been annotated specifically, and the annotation has strong relations with any audio/video recording segment used in the given document, it is considered as a primary annotation. On the other hand if the annotations are primarily made for another video document but its audio/video recording segment is used or has some overlapping part with the segment used in the given document, then the secondary context of annotations is used.

When the player is playing a video document, a Content annotation list box in the Video Document Browser is updated accordingly to show the related content annotations to the part of the document being played, and the related structural component is highlighted in a Structure list box in the browser. The user can control the browsing operation by specifying which types of contexts (primary, basic and/or secondary) s/he is interested in.

The structure of a video document is composed of structural components (sequences, scenes and shots) that are organized in a hierarchical form, and are presented as a table of content for the document. The browser also allows users to easily navigate the

structure of the document being played; for example, selecting a structural component from the Structure list box causes the player to move to and play the selected part.

3.2.2.2 Video Querying

Rowe (Rowe et. al, 1994) surveyed a variety of users of multimedia database systems, characterised the types of video queries they needed, and identified the following "indexes " that should be associated with the video data in order to satisfy queries:

- **Bibliographic Data:** This category includes information about the entire video (e.g., title abstract, subject and genre) and the individuals involved with the production (e.g. producer, director, cast etc.)

- **Structure Data:** This level includes the indexes about hierarchy, scene, segment and shot, where each entry is composed of one or more entries at a lower level. For example, a segment is composed of a sequence of scenes.
- **Content Data:** Users of a video–retrieval system want to find videos on the basis of the semantic content of the video. Video contains visual content and audio content. In addition, because of the nature of the video, the visual content is a combination of static content (frames) and dynamic content. Thus, users may want to search 1) sets of keyframes (e.g. a key frame for each actor) ; 2) keyword indexes built from sound track and /or closed-caption ; 3) object indexes that indicate entry and exit frames for each appearance of a significant object or individual.

A video query is more complex than a traditional query of text databases. In addition to text (closed–captions and annotations), a video segment has visual and audio information as well as the dynamics associated with the presentation of such information.

The primary concern of any video retrieval system is that a query should be natural and easy to formulate, that the user-interface assist in a user-friendly way in the query formation process, and that the search results should be presented in an organized and sensible form. The systems discussed below have novel ideas about video querying:

VideoSQL (Oomoto & Tanaka, 1993) is a high-level query language that has been designed for a video database prototype called OVID (Object-oriented Video Information Database). A query is formulated by fill in the blanks for SELECT, FROM and WHERE clauses by selecting from a list of available options for each clause. The SELECT clause has options for specifying whether or not the required type of the resulting video segments (video objects) is a single continuous segment. The name of a video database is specified using the FROM clause, while the WHERE clause allows a user to choose the attribute, attribute value and comparison operator for the query. The value for an attribute can be a value or a video object. Three types of comparison operator were provided: *is*, *contains* and *definedOver*. The *is* operator allows video retrieval using specific attribute/value defined in video objects or using general definitions defined in the generalization is-a hierarchy. The *contains* and *definedOver* operators are for conditioning set-type attributes and frame numbers correspondingly.

Simonnot (1995) developed a video document retrieval system that uses *a facetted structure of descriptions*. A user can query the system by selecting facets and keywords, and providing a weight for each keyword. Then the query is processed using several intelligent agents such as thesaurus agents, language processing agents and probabilistic agents. Although the system allows a document to be described at different structure levels, it uses sequences as the basic units for results of a query. A frame and a short text description represent each sequence. Any user decision on the result is used by the system to enhance future retrieval.

Zhang (1995) has implemented a system that uses visual features in key frames to index videos. The system provides query formulation based on key frame features. Colour, texture and shape features in key frames were used during the indexing process. The

system allows a user to formulate a query by example where the example can be a sketch consisting of strong spatial information of the required frames or a sample key frame that has been chosen using a browsing tool.

Ahanger (1995) gave the idea of motion-based query. This type of query has been implemented in a database system called MovEase (Motion Video Attribute Selector). A user can formulate a query by specifying a combination of object and camera motions visually. This is possible because the system uses icon catalogue for managing objects, motions and formulated queries. Selecting an object from an object catalog, selecting camera motions and/or object motions from a motion catalog, and associating the selected motions to the selected object undertake the query formulation.

3.2.3 The MAVIS Project

The MAVIS project is a programme of research to develop Multimedia Architectures for Video, Image and Sound in the IAM (formerly MMRG) research group at the University of Southampton. As discussed in chapter 2, a modular approach is used and different modules are responsible for all the processing associated with a particular media-based feature type and, as new feature types are introduced, associated matching techniques are developed and added to the main engine. For example, to make use of the added richness which digital video presents, modules are being developed which understand the temporal nature of the video and which can extract combined spatial and temporal features. This will be further used in the multimedia thesaurus (MMT) and intelligent agent support for content-based retrieval and navigation

In MAVIS 2: an extension of MAVIS 1: semantic layer components are also included to enhance the efficiency of query retrieval. MAVIS 2 also allows content-based retrieval in the form of 'query by example'. A query may be in any medium for which appropriate plug-in modules are available. This also includes free text queries.
Content-based hypermedia navigation is also possible using Microcosm style specific and generic links (Fountain et. al, 1990; Heath, 1992; Hall, 1994). These generic links can

be authored on arbitrary media types in a similar manner to MAVIS 1. Intelligent agents are developed to classify and cluster information from the MMT, as well as additional knowledge extracted during information authoring and indexing, to seek out discriminators between object classes and also naturally- occurring groupings of media-based features and to accelerate media-based navigation. The idea of extracting semantic information between multimedia objects is novel, and to add a semantic layer to define the relationship between these objects is of much interest.

3.3 Discussion

Research in the field of video databases and video understanding is going on at a tremendous pace. Almost every multimedia research group, is coming up with some video solutions, ranging from high-speed compression rates to semantically incorporated segmentations. A huge amount of time and money is being spent on trying to develop new solutions for easy and efficient video retrieval and query systems. The following are the important issues, which need addressing, while working on video databases, segmentation and video indexing:

- Since video is a continuous media, it should be stored separately and should not be chopped or interlinked with the metadata. i.e., the raw video data should be kept raw and the other metadata or annotations should be stored in a separate database.

- One of the major issues is labelling of segments i.e. what sort of annotation should be provided for the segment or what constitutes a suitable description for a segment, is free text sufficient or are more structured descriptions necessary? Can keywords provide an adequate description?

- Providing semantics to the annotations is still one of the problems, as there is no standard available that can define up to what level of hierarchy a user can provide semantic. What sort of relation can a user define between two objects in

a video? What sort of relation should be incorporated between two segments? etc.

- It is seen that a framework is required for video retrieval and query formulation that should be based on an iterated sequence of navigation, searching, browsing and viewing. This framework should include certain capabilities of video query systems, in particular, search on dynamic visual properties of video and the ability for rapid non-linear viewing of video.

3.4 Metadata Standards

This section describes the relationship between video information system and the domain of digital libraries. Digital Libraries (DL) providing access to Multimedia Database Management Systems (MM-DBMS) are becoming more and more important due to the increasing popularity of information networks like the World Wide Web. The problem is that users of DLs usually have to browse through large amounts of data in order to find informative, i.e. relevant, parts of documents. Browsing means -in this context- that users want to scan quickly through data to inspect and compare information they specified only roughly by a previous query. A Digital Library system has to provide support for this kind of information-intensive work. Hence, the access to data should be based on task-related, conceptual, and content-based criteria, especially for time-dependent data like video, whose presentation is very time-consuming. In a browsing application users need efficient ways to access those parts of videos that contribute to this relevance in their given context.

Browsing applications not only have an impact on the required system's data access functionality, but also on the presentation and interaction capabilities. Depending on the users' preferences, some data are more and some less important. The main goal of users in video browsing applications is to find information quickly, rather than to actually view videos as in Video on Demand applications, since they will not have time to systematically go through and view the entire data. The system has to support efficiently

the browsing process by means of automatic selection and structuring of the relevant data.

With an enormous number of digital libraries emerging on the WWW, one of the major issues for digital video libraries is the standardisation of metadata and annotations. Resource Description Framework (RDF) and Dublin Core (DC), designed by World Wide Web Consortium (W3C), and the Moving Picture Experts Group - 7 (MPEG-7) are among the most important standards in this regard. RDF is a specification currently under development within the W3C Metadata activity (*http://www.w3.org/Metadata*). RDF is designed to provide an infrastructure to support metadata across many web-based activities. It is the result of a number of metadata communities bringing together their needs to provide a robust and flexible architecture for supporting metadata on the internet and WWW. Its design has been heavily influenced by the Warwick Framework work (Lagoze et. al., 1996).

RDF provides a uniform and interoperable means to exchange the metadata between programs and across the web. It also provides a means for publishing both a human-readable and a machine-understandable definition of the property set itself. RDF also allows different application communities to define the metadata property set that best serves the needs of each community. XML (Extended Markup Language) is used as the transfer syntax in order to enforce the other tools and code bases being built around XML. For example, SAMI (Synchronized Accessible Media Interchange), developed by Microsoft, and SMIL (Synchronized Multimedia Integration Language), developed by W3C (*http://www.w3.org/TR/WD-smil*), are the examples of web-based multimedia presentations encoded in XML. Generally, the goal of RDF is to define a mechanism for describing resources that makes no assumptions about a particular application domain, nor defines (a priori) the semantics of any application domain. The definition of the mechanism should be domain neutral, yet the mechanism should be suitable for describing information about any domain.

Dublin Core (DC) was designed specifically for generating metadata for textual documents. (*http://www.purl.org/DC*). It is a cross-disciplinary international effort to develop mechanisms for the discovery-oriented description of diverse resources in an electronic environment. The Dublin Core Element Set comprises fifteen elements (DCES), which together capture a representation of essential aspects related to the description of resources.

The majority of work on the Dublin Core has addressed the definition of semantics rather than syntax or structure, allowing rapid conceptual development free of the constraints imposed by specific implementation environments. Whilst beneficial in many ways, this has led to a certain lack of clarity at times, especially in relation to the development of 'qualification' mechanisms, which enrich descriptions in the Dublin Core. It has also made interoperable implementation difficult, as individual implementers have typically developed their own internal mechanisms for actually encoding Dublin Core; mechanisms which are not always compatible with those of their potential collaborators elsewhere.

The Dublin Core Data Model working group (DCDM) was set up to look at means by which the richness of the Dublin Core model might be expressed with in the limitations of HTML. This document (Kunze, 1999) represents a technical report on two specific outcomes from this process; a means by which the model may be considered, extended, tested and manipulated within the Resource Description Framework (RDF); and suggested mechanisms by which both simple and complex Dublin Core might be expressed using XML. Although RDF is the 'language' used to express the data model, users are not limited to using only RDF in their own applications. Similarly, although examples throughout this document are expressed in XML, this does not mean the Dublin Core may only be encoded in this way. *(http://www.ukoln.ac.uk/metadata/resources/dc/datamodel/WD-dc-rdf/)*

The elements of Dublin core are: Title, Creator, Subject, Description, Publisher, Contributor, Date, Type, Format, Identifier, Source, Language, Relation, Coverage and

Rights. A complete description and definition of these elements is available at Dublin Core Web Page. *(http://www.purl.org/dc/docments/dcmes-qualifiers)*.

The objective of MPEG-7 (Hunter, 1999) is to provide standardized descriptions of audiovisual information to enable it to be quickly and efficiently searched. MPEG-7, formally called `Multimedia Content Description Interface', will standardise:

- A set of description schemes and descriptors, and
- A language to specify description schemes, i.e. a Description Definition Language (DDL)

MPEG-7 will address the coding of these descriptors and description schemes. The combination of descriptors and description schemes shall be associated with the content itself, to allow fast and efficient searching for material of a user's interest. AV material that has MPEG-7 data associated with it, can be indexed and searched for. This `material' may include: still pictures, graphics, 3D models, audio, speech, video, and information about how these elements are combined in a multimedia presentation (`scenarios', composition information).

The development of the MPEG-7 standard is still at a very early stage. The call for proposals was scheduled for October 1998 and the Draft International Standard is not expected to be published until July 2001. But given the overlap in objectives between MPEG-7 and Dublin Core, it makes sense for the MPEG-7 community be aware of the work of the Dublin Core community and vice versa, to ensure compatibility and interoperability, where possible. A further extended discussion on MPEG-7 is provided in Appendix A.1.

3.5 Conclusion

This chapter dealt with the issues related to digital video and metadata including video segmentation, video data indexing and retrieval and metadata standards are discussed

in this chapter. To cater for these problems, we have developed a video database model tv-DbMS, with a novel technique of thematic indexing.

While developing tv-DbMS, we have kept the above-mentioned problems in view and have tried to answer to some of them. In the next chapter we will be discussing the data model for tv-DbMS.

Chapter 4

The tv-DbMS Model: Initial Development

This chapter deals with video data modeling. This process starts from digitizing a video in any standard format, performing segmentation, applying some database management techniques to store the data. The next step is to provide annotations/metadata to a particular video, to create a comprehensive video document. Once the video document is ready, then queries can be performed on that video data. This chapter deals with segmentation, relational and object oriented data model of tv-DbMS.

4.1 Video Databases

Figure 4.1 shows a general form for a video database. A user interacts with a query interface. Then the user query is processed by a match engine, which in turn extracts data from the video database. Because of at the huge size of video data, some sort of filters and indexes are required to retrieve the information more efficiently. In the end, the best matches are returned to the user in the form of selected video clips.

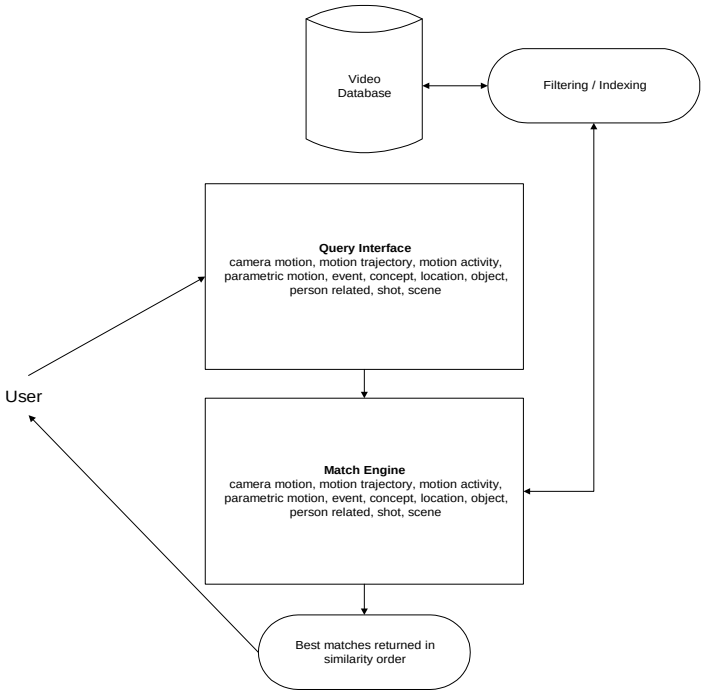


Figure 4.1: General database model

4.2 A new approach to video information management

To enable tv-DbMS to support the management of digital video, it is necessary to define the characteristics of a video document. In the context of tv-DbMS, a video document is a piece of video that has some metadata to support the raw video and some hyperlinks to provide the non-linear access to the video. The general architecture of tv-DbMS consists of some offline and some online processing. During the offline phase the video editor (human annotator) provides the metadata to the video catalogue in tv-DbMS, along with some other processes, which also extract audio and visual information from the digitised data. This is shown in figure 4.2.

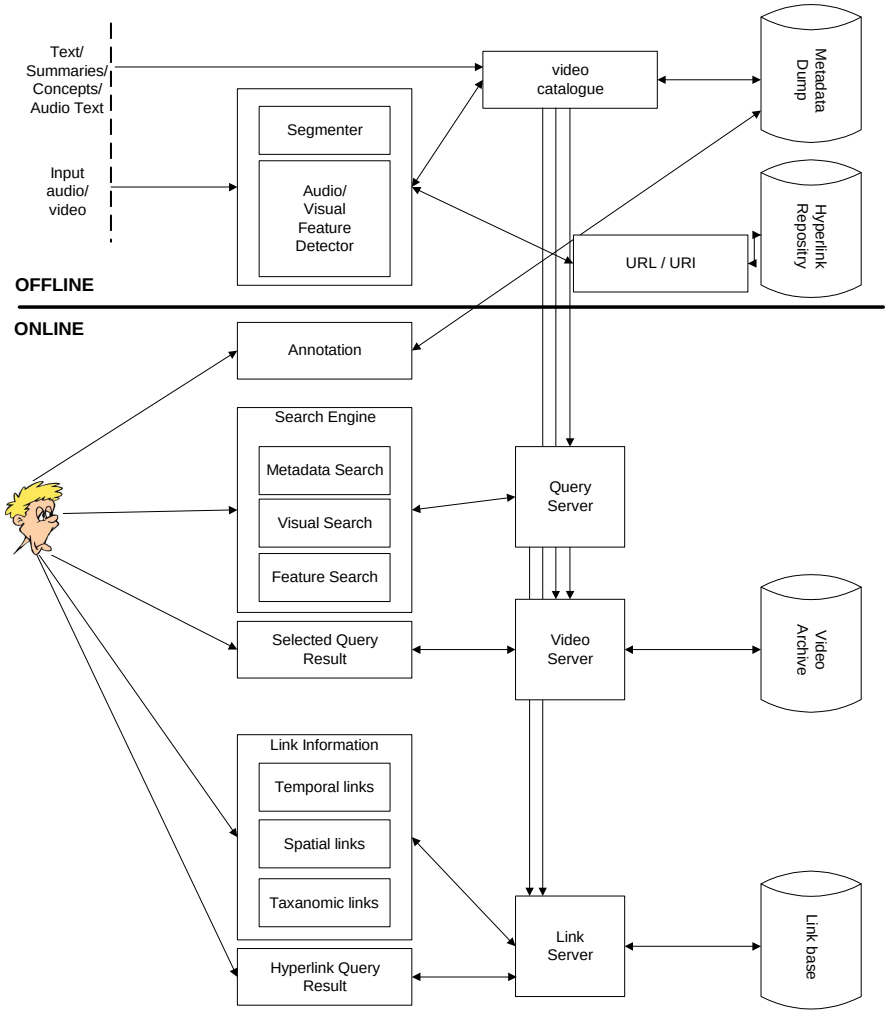


Figure 4.2: tv-DbMS video architecture

During the online or real time phase, the user can provide further annotations to the different segments or the events in a video clip. The user can interact with the search engine to perform any query on the video using metadata or any visual or feature search, and is also able to navigate through the link information engine to retrieve hyperlinks.

The search engine is connected to a query server, which retrieves the data from the video archive through a video server. Apart from searching for video clips, the user can also navigate into a particular video segment or scene and can look for temporal, spatial

or taxonomic links. Here it should be noted that in the tv-DbMS model, the video is kept separate from the metadata and the hyperlink repository. This sort of architecture is used, so that the original video can be used or re-used by other applications or by other commercial video players. (All the videos stored in tv-DbMS are either in avi, mpeg-1, in asf (mpeg-4) format)

Digital video is divided into streams, sequences, scenes, shots and frames, as defined in chapter 1. A video database can be defined as a collection of sets where the elements (video objects) of each set are elements of the same type. Here the object – oriented approached is used in tv-DbMS, so that each element is identified by a unique object identity. Objects may be complex with attributes that are objects on their own.

The following list gives a short description of each of the different object types, which are relevant for the data model of tv-DbMS:

- **Stored Media Segments:** Audio and video data which are generated during recording.

- **Video Streams:** Video documents are composed of pieces of stored media segments. A video document represents a logical stream of video data that may be explicitly stored.

- **Media Streams:** Media streams are a generalisation of stored media segments and video streams.

- **Stream Interval:** A stream interval represents a contiguous sequence from a media stream that is explicitly identified.
- **Video Content Indexes / Annotations:** For browsing and querying purposes there is a need to relate entities from a real-world model to pieces of video. This relation is made by annotating a video object that identifies the stream interval of interest, which is linked to an element of a real-world model.

- **Video Document Structure:** Video documents can be represented by a set of structural components, where each component identifies a stream interval.

Video data is stored as contiguous groups of frames called stored video segments. A video document is represented by a video stream which is mapped to one or more stored video segments. An important and flexible video information unit is frame sequence which is an interval of subsequent frames from a video document. One single frame sequence can represent any sequence of video frames ranging from one single video frame to an entire video document. The frame sequence is also responsible for connecting structural units and thematic annotations to the video material. A frame sequence can also be defined as a 'part of' relationship to a video stream, as shown in figure 4.3.

The structure of a video document is represented by a hierarchy of structural components. Each structural component identifies a frame sequence, which consists of the frames that belong to the component. The entire video document or a single frame can also become a structural component, but a whole document is too coarse as a level of abstraction and a single frame is rarely a unit of interest.

Experiments with documentaries such as Earl Mountbatten's videos (Khoja & Hall, 1999a) have shown that abstractions such as scenes or events make it easier for a user to make references to video information and hence make it easier to comprehend its contents. This is also reinforced by the work of Grosky(1997). Therefore, more emphasis in tv-DbMS is given to metadata, which are categorised as Event, Location, Person, and Object_of_Interest (This is further discussed in section 4.4.).

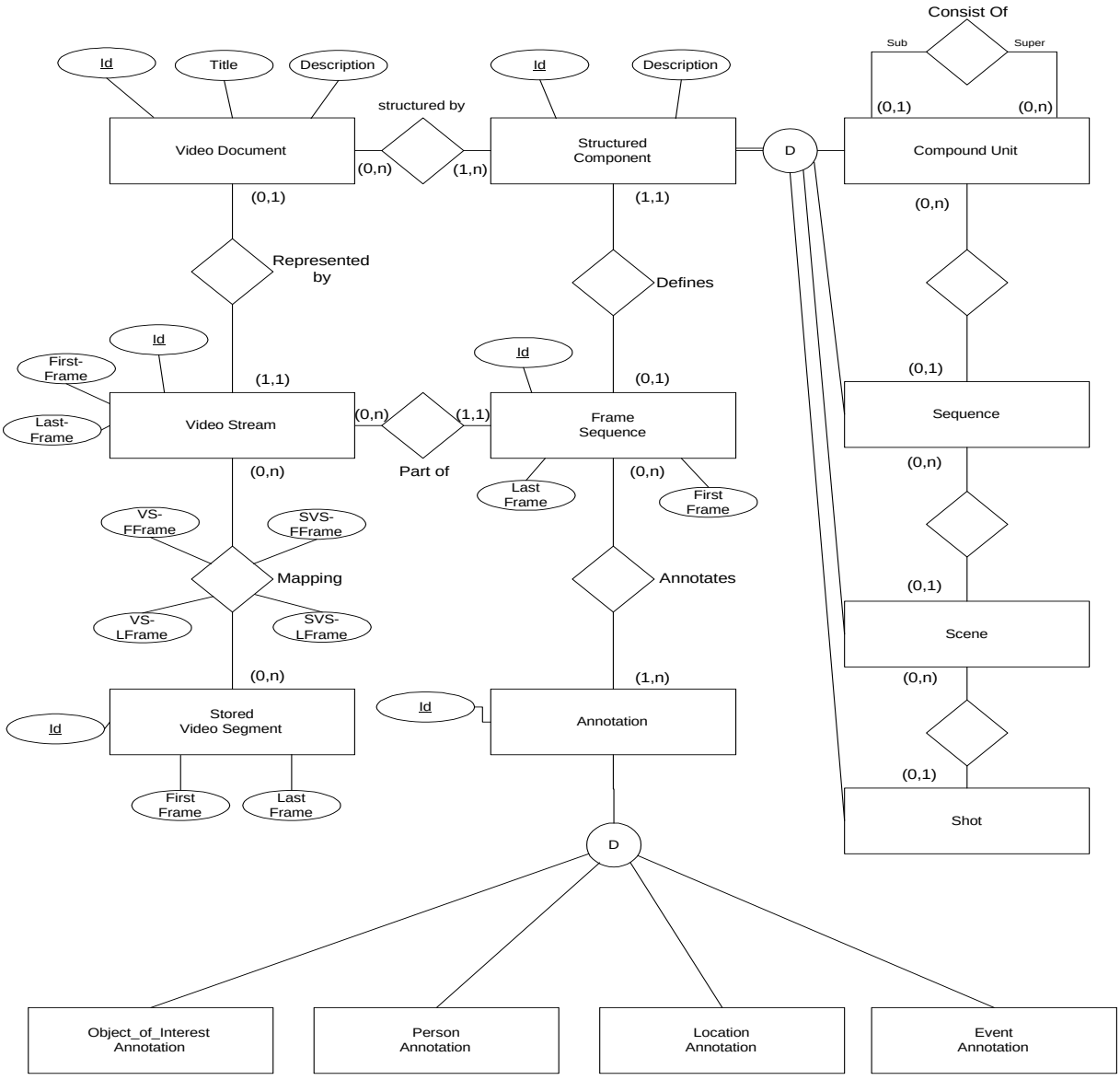


Figure 4.3: A common data model for video information sharing using an ER-notation, suggested by Hjesvold (1994a). Here the video document is categorised into frame sequence, which is further divided into different types of annotations

4.3 Segmentation

First, using a basic scene detection algorithm, the video was divided into different segments. The algorithm used to create segments is by means of the histogram count of the current frame and the previous frame. If the difference between the histograms is bigger than a certain threshold value, then the particular frame is marked as the start of a new segment.

The code shown below perceives the frame as a bitmap (bmp) image divided into blocks of 64*64 pixels. Then the rgb (i.e. red, green and blue) colour values are calculated for these blocks and stored in an array. The same process is then repeated on the next frame. Once the histogram (in rgb) of blocks is calculated, the colour difference is calculated. After many trials, it was found that a threshold of 0.5 produced optimum scene breaks. Hence a threshold of 0.5 was set for segmenting the videos.

```
for( frame_y_size = 0; frame_y_size < height; frame_y_size+=block_size_y )
        for( frame_x_size = 0; frame_x_size < width; frame_x_size+=block_size_x )
        {       activity1 = 0.0;
                activity2 = 0.0;
                for( block_y_size = frame_y_size; block_y_size <frame_y_size+block_size_y;
                block_y_size++)
                for( block_x_size = frame_x_size; block_x_size <frame_x_size+block_size_x;
                block_x_size++)
                        {
                        activity1 += (float)(*(red + frame_y_size * width + frame_x_size) +
                        *(green + frame_y_size * width + frame_x_size) + *(blue +
                        frame_y_size * width + frame_x_size));

                        activity2 += (float)(*(image2->red+ frame_y_size * width +
                        frame_x_size) + *(image2->green + frame_y_size * width +
                        frame_x_size) + *(image2->blue + frame_y_size * width +
                        frame_x_size));
                        }
```

```
                activity1 /= (float)(block_size_x * block_size_y);
                activity2 /= (float)(block_size_x * block_size_y);
                if( fabs(activity1 - activity2) > 100.0 )
                        global_activity ++;
                        number_of_blocks++;
        }
// See if frame is a scene break.
if( (float)(global_activity)/(float)(number_of_blocks) > 0.5 )
        {       cout << (float)(global_activity)/(float)(number_of_blocks )
                << "*" << global_activity <<
                   "*" <<number_of_blocks<<"*"<<
                        (int)((float)(height)/(float)(block_size_y)) << "*"<<
                        (int)((float)(height)/(float)(block_size_y)) << "*\n"  ;
        }
for( frame_y_size = 0; frame_y_size < height; frame_y_size+=10 )
                for( frame_x_size = 0; frame_x_size < width; frame_x_size++ )
                {
                        *(red + frame_y_size * width + frame_x_size) = 255;
                        *(green + frame_y_size * width + frame_x_size) = 255;
                        *(blue + frame_y_size * width + frame_x_size) = 255;
                }
}
```

After segmentation, the scenes were given captions manually and stored in a database. Here it should be noted that it is not sufficient to store the captions or the metadata in a normal database. Since a digitised video contains temporal as well as spatial properties, an efficient and comprehensive data model is required to store the video metadata.

4.4 tv-DbMS object model

It was also found that in documentaries, such as that of Earl Mountbatten, video modeling was heavily dependent on annotations or metadata of a video. Since algorithms for computer vision are still not able to make sense of or interpret all the data in a scene or a sequence, especially concepts, themes and ideas, special importance to the metadata is given in tv-DbMS model.

Audio data is the other domain that can provide a lot of information about the video. Applying a speech-to-text converter, the system can generate streams of data that can be used to extract metadata. Informedia II (Christel, 1999) is one of the examples of these systems.

Apart from metadata, data regarding frame size, frame rate, colour, texture and other features are also of important value. To create a comprehensive data model, all the above mentioned entities should be modeled in such a way that issues such as data storage, query processing and data indexing are optimally resolved.

During the literature review phase, it was found that Hjesvold (1994a) had developed a very precise model for annotating a segment. Metadata is generated from audio components as well as video components. Textual information about an event, person, location or an object is also inserted, as shown in figure 4.4. Hjesvold's (1994b) idea was the foundation of the tv-DbMS model, with slight modifications such as adding the entity of concept in the metadata and placing the event entity on top of the segment entity in the hierarchy.

For tv-DbMS, considerable modifications are made in the data model shown in figure 4.3. Since in the tv-DbMS model an entity is considered as an event, and a segment is a subset of an event, so the event has been made the pivotal item, which comprises of segments, as shown is figure 4.5. Furthermore, the segment also describes the annotations for the objects inside the scene, such as person, location, etc. Each object has its own metadata, which creates a many-many relationship with the segment metadata. This is then normalised to create a relational database for accessing any object present in the video scene.

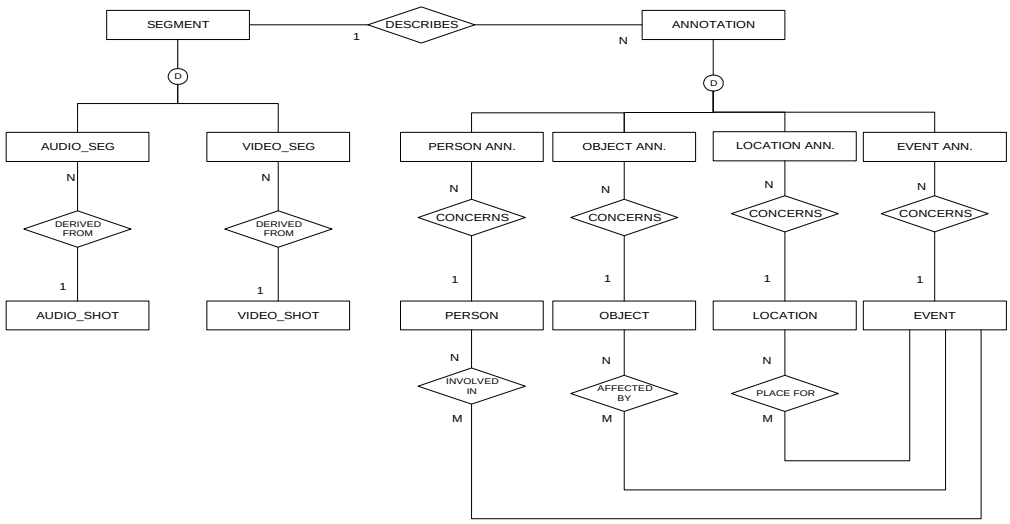


Figure 4.4: Data components of a generalised video segment, suggested by Hjesvold

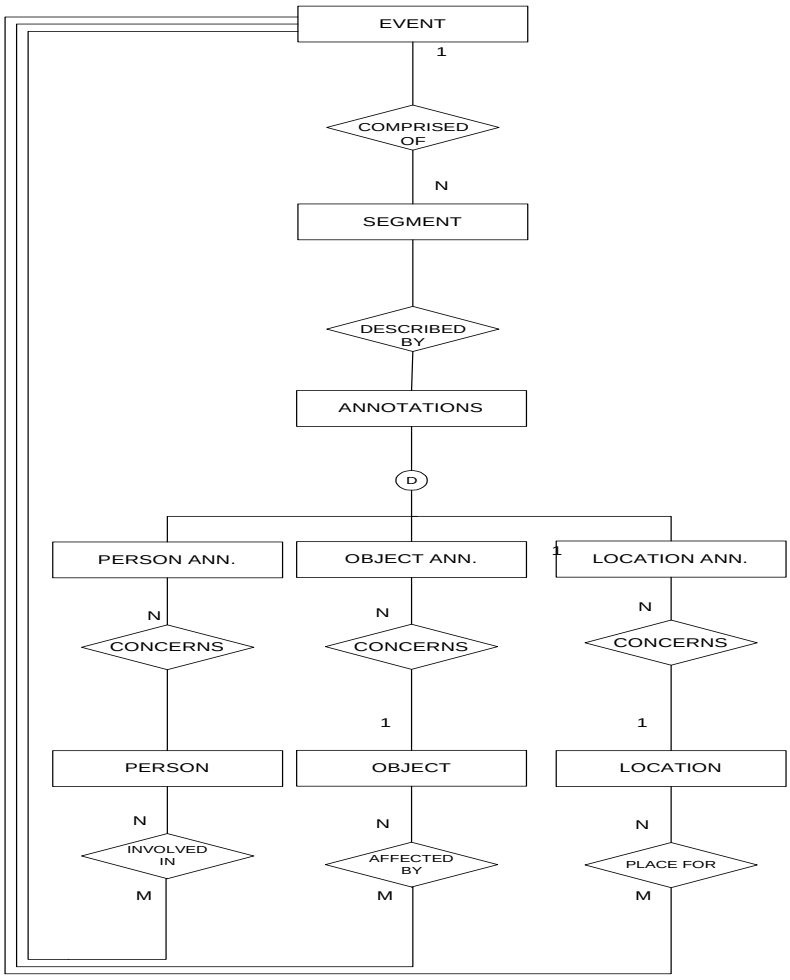


Figure 4.5: Suggested annotation model for tv-DbMS. Here the 'event' entity is added and assigned a top level in the hierarchy, as opposed to Hjesvold's model.

4.5 tv-DbMS Relational Model

In this section, all the operations and attributes applicable to a composite entity are discussed. The attributes of a composite entity are:

Object Identity:	The unique identity of the component
Symbolic Name:	A text string that the author may optionally assign to the component, i.e. to have a user-generated name for identifying the component.
Synopsis:	Textual description of the component.
Architecture Operations:	This group includes operations for creating, modifying or deleting components in the component architecture and for manipulating the synchronisation of components in a narrative presentation.

In tv-DbMS, an event is defined as the set of scenes, constituting a part of a story. In the tv-DbMS data model the major object entity is an event. When a person starts thinking or remembering any scene, he or she always starts thinking in terms of events: a complete part of story: rather then thinking in terms of scene changes or eye (camera) movement, which are detected by machines. When people recall something, they often say "*I played a movie of that event in my mind*". This tells us that an event plays a very important role in our memory. An event comprising one or more segments, can be generated by another event or can generate many sub events, and many events can generate a single sub event, as shown in figure 4.6.

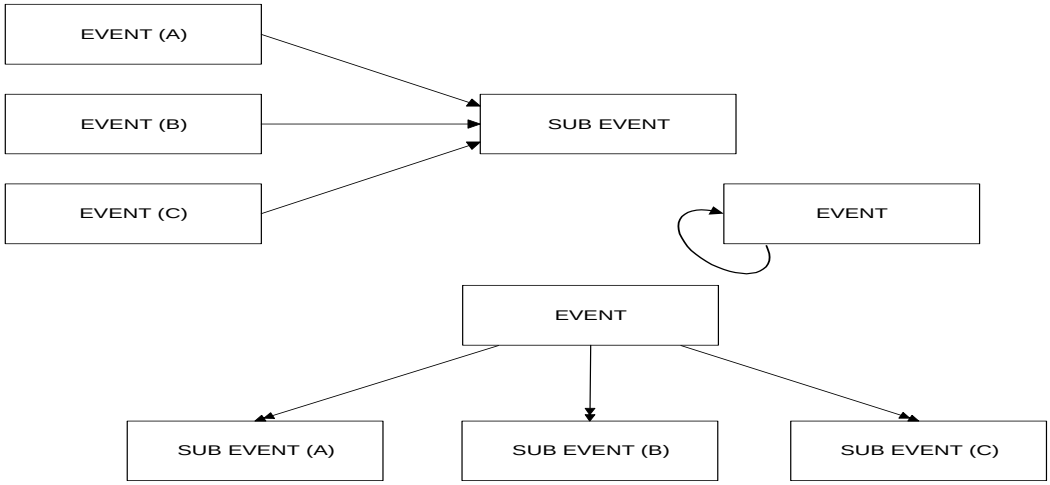


Figure 4.6: Inheritance properties of an Event

Hence the entity 'event' is proposed over the entity 'segment', as shown in figure 4.6, as compared to Hjesvold's generalised video model in figure 4.4. The main objective in this modification is to maintain the sense of a story. Again, the other entities, such as person, location, object_of_interest, etc. are the sub group of the event. The ER diagram is shown is figure 4.7.

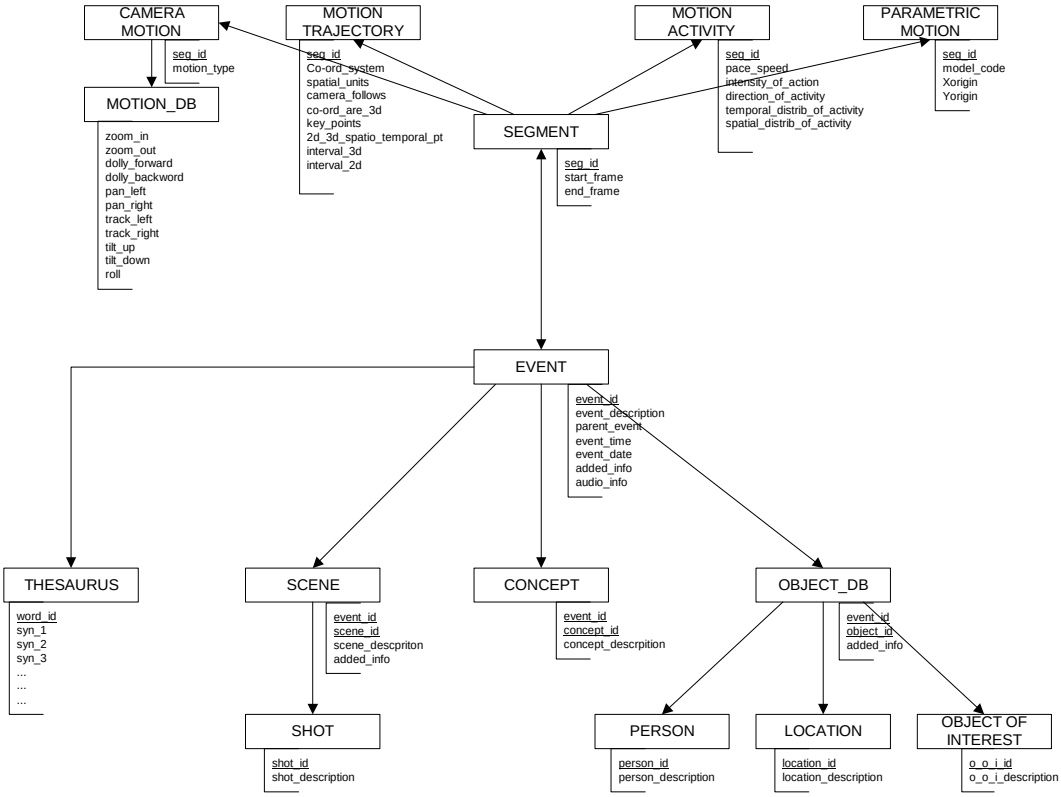


Figure 4.7: ER diagram of annotation model

An entity of 'concept' is introduced in the data model, where a user can provide some additional textual information about the scenes, which are hard or impossible to categorise under person, location or object_of interest entity. The idea of concept as a textual study is also used in the Vane System (Carrer et al, 1997), where the entities are

classified as tangible and conceptual entities, but as opposed to the tv-DbMS model, are not the part of the event entity.

The attributes to the entities shown in figure 4.7 are described as:

4.5.1 Segment

This is the main entity of the tv-DbMS data model. These segments are generated through the scene detector module. The segments are the smallest visual part of a scene or a shot.

Elements of the segment entity are:

Segment id An unsigned integer value, which is the primary key to this table and foreign key to all the related tables.

Start frame An integer that depicts the frame number of the starting time.

End frame An integer that depicts the frame number of the ending time.

4.5.2 Event

An event in tv-DbMS terminology, is defined as the set of scenes, constituting a part of story. This entity plays the key role in the annotations, as the whole idea of thematic indexing (discussed in Chapter 5) revolves round it. The event entity has a many-to-many relationship with segment entity, which is resolved by introducing a dummy entity named seg-event. The main components of event entity are:

Event id An unsigned integer, constituting the primary key for the event entity.

Event description

An array of strings describing the particular event. The maximum size is 255 characters.

Event time This is a redundant attribute of the event entity. It is used to note the time the particular event occurred. This is in the format of hh:mm.

Event date This is also another redundant attribute of event entity. It is used to note the date the particular event. This is in the format of dd:yy.

Added info This is a memo type parameter that can store the textual information up to 5000 bytes. It is used to store any other information related to the event entity.

While generating a story line, some cross-references are additionally required to connect two or more elements. This sort of information is stored in this parameter.

4.5.3 Concept

In the tv-DbMS model , a concept is a term used to name any object or idea, which is very hard to describe in terms of physical objects. For example a person laughing or crying, or in the case of the Earl Mountbatten videos, the army troops having a free time etc.

Concept id This field specifies a unique id number assigned to any concept stored in the database.

Description String value, that describes the description about the concept. Here a user can provide complete data about the concept.

4.5.4 Person

This entity contains the information about the people in a video, the brief history about that person, his titles, and activities can also be stored in the description part of this video.

The activity entity includes the following attributes:

Person id This attribute is used to store the unique id assigned to a person.

Name A string value storing the name of a person.

Title A string value storing the title of a person.

Description This attribute is used to store the further information if required.

4.5.5 Location

This entity stores all the locations described in the video database. This consists of the following attributes:

Location id This integer value assigns unique id to different locations stored in the database.

Description Location description is used to describe additional information about a particular location.

4.5.6 Object

This entity stores all the physical objects used in the video database. This is made of following attributes:

Object id This integer value assigns unique id to objects stored in the database.

Description Object description is used to describe the additional information about a particular object

4.5.7 Thesaurus

This entity is used to assign an online lexical dictionary to the video database. This entity mainly deals with similar nouns, verbs and adjectives, related to a particular scene, event, object or location in the database. This table consists of the following attributes:

Word id The word id indicates the words that are to be used in the tv-DbMS thesaurus.

Synonym (1..n)

Synonym'*n*' are the similar or related nouns, verbs or adjectives stored in the database for the word_id, mentioned above. They are usually derived

from the "Word-Nnt" (Miller, G.A., et. al. 1994) directory or otherwise a user can also provide to the system manually. For example in the Earl Mountbatten's videos, words like battle, clash or fight are used to depict the word war.

To complete the data model, the following entities are also included in tv-DbMS:

4.5.8 Camera Motion

This entity describes 3-D camera motion parameters. It is based on 3-D camera motion parameter information, which can be automatically extracted or generated by capture devices, but in our model the user has to provide this information manually.
A camera motion can be fixed, panning (horizontal rotation), tracking (horizontal transverse movement, also called traveling in the film field), tilting (vertical rotation), booming (vertical transverse movement), zooming (change of the focal length), and dollying (translation along the optical axis) and rolling (rotation around the optical axis).

The sub-shots for which all frames are characterized by a particular type of camera motion, which can be single or mixed, determine the building blocks for this camera motion entity. Each building block is described by its start time, the duration, the speed of the induced image motion, by the fraction of time of its duration compared with a given temporal window size, and the focus-of-expansion (FOE) (focus-of-contraction – FOC).
The mixture mode captures the global information about the camera motion parameters, disregarding detailed temporal information, by jointly describing multiple motion types. On the other hand, the non-mixture mode captures the notion of pure motion type and their union within a certain time interval.

Elements of Camera Motion Entity:

start_time An unsigned integer value. This represents the relative starting time of the segment, with respect to the beginning of the video sequence. This is

the foreign key gathered from the segment table and an instance of segment_start_time.

Duration A 16 bit unsigned integer value. This represents the duration of the given temporal segment, in a specified time unit *(to be defined, issue related to the TimeDS).*

Fractional Presence

These values represent the temporal presence of the different motion types.

Amount Of Motion

These values represent the amount of motion of the different motion types.

FOE FOC Horizontal Position/FOE FOC Vertical Position

The values giving the normalized coordinates of the Focus Of Expansion (or Contraction).

4.5.9 Motion Trajectory

Motion trajectory is a high level feature associated to a moving 2D or 3D region, defined as the spatio-temporal localization of one of its representative points (such as its centroid).

Elements of Motion Trajectory Entity:

Camera Follows

This field specifies whether or not the object is followed by the camera. When it is not present, it means this notion is not specified in the Descriptor.

Co ord System This field specified the spatial coordinate system used to express trajectory: 'local' if the trajectory is expressed in the image reference, 'world' if it is expressed in an absolute reference taking into account

global motion. If this field is not present, the local coordinate system is taken as the default reference system.

Spatial Units This field will be used in all descriptors needing spatial units, describing a particular spatial region, depending on a particular object's trajectory in world co-ordinates.

Co-ords Are 3D

This Boolean indicates whether the trajectory is specified using 3-D or 2-D coordinates.

Key Points Number

This field contains the number of sampled positions, denoted as *key points*, used to represent this trajectory.

2D/3D Spatio Temporal Point

These types contain the times and coordinates associated to each key point. Time values are strictly monotonically increasing in time

Interval 3D This field specifies the interpolation functions used for 3D trajectory.

Interval 2D This field specifies the interpolation functions used for 2D trajectory.

4.5.10 Parametric Motion

Parametric motion addresses the motion of objects in video sequences, as well as mosaic description by modelling global motion. If it is associated to a region, it can be used to specify the relationship between two or more feature point motion trajectories according to the underlying motion model.

ModelCode This integer number specifies the model type used in the description.

Incr Duration

This field specifies the length of the temporal interval to which this Descriptor is associated.

MotionParameters

This is a floating point array that keeps the values of the model parameters.

X origin, Y origin

These are the coordinates of the origin of the spatial reference, with respect to the image coordinates.

4.5.11 Motion Activity

A person watching a video or animation sequence perceives it as being a slow sequence, fast paced sequence, action sequence etc. The activity descriptor captures this intuitive notion of 'intensity of action' or 'pace of action' in a video segment. Examples of high 'activity' include scenes such as 'goal scoring in a soccer match', 'scoring in a basketball game', 'a high speed car chase' etc. On the other hand scenes such as 'news reader shot', 'an interview scene', 'a still shot' etc. are perceived as low action shots.

The activity entity includes the following four attributes.

Intensity of Activity

A high value of intensity indicates high activity while a low value of intensity indicates low activity.

Direction of Activity

The Direction parameter expresses the dominant direction of the activity if any.

Spatial distribution of Activity

The spatial distribution of activity indicates whether the activity is spread across many regions or restricted to one large region. It is an indication of the number and size of "active" regions in a frame. For example, a talking head sequence would have one large active region, while an aerial shot of a busy street would have many small active regions.

Temporal Distribution of Activity

The temporal distribution of activity expresses the variation of activity over the duration of the video segment/shot. In other words, whether the activity is sustained throughout the duration of the sequence, or whether it is confined to a part of the duration.

Intensity

This attribute can be expressed as a single integer lying in the range [0, 128], as used by semantics to express the intensity of activity.

DirectionFlag

This flag indicates whether the directional attribute has been specified or not.

4.6 tv-DbMS front end

The major requirement for our experiments was the need for a digital video player, which allows the user to annotate, insert metadata and to insert links into digital video. After considering the current experimental and commercial digital video players, we decided to develop our own video player, which can allow us to do the above mentioned task, as well as to meet the following criteria:

- It should be capable of playing all current video formats, e.g. avi, mpeg, quicktime, asx, etc.
- It should connect the video file with a database, which can provide metadata for the video.
- Once the start and end frames have been selected, the player should be able to run the selected part of video .

A player was developed in the Microsoft windows environment, which was capable of fulfilling the above mentioned requirements. Then a database of annotations, based on the methods stated in section 4.3 was developed in Microsoft Access 97, and was then connected to the video player. Figure 4.8 shows the front-end of the tv-DbMS video

player. This is a tool for editing, maintaining and browsing video collections that can receive and display the items selected from the video retrieval.

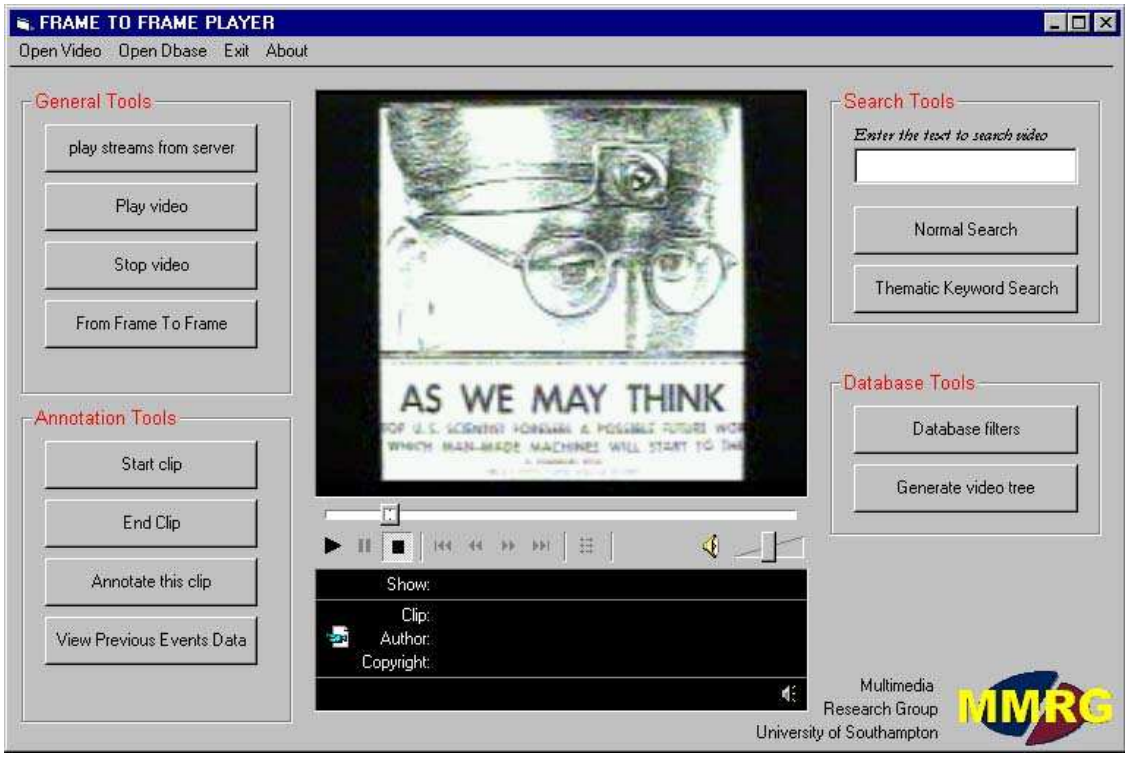


Figure 4.8: Front end of tv-DbMS

A number of options have been incorporated into this front end. On the left hand side are the general options such as playing streams, playing a video, stopping a video, and playing a particular clip of video from certain frames. The other buttons are for annotating a particular segment or an event. The View Previous Events Data button opens another window, where a user can view or edit the metadata. Here the option is also provided to establish relationships between segment annotations and event annotations. This window is shown in figure 4.9.

On the right side of the main player are the tools for querying the video database. Two options are provided; the first is a simple query search, where a user can perform simple queries. The second is the thematic search, where a hierarchy of events or objects is

created. Here the system also searches for the similar words or notions in the thesaurus database and track the similar words and then perform the search again to fetch some related data form the video server. On the bottom right of the front end are the database tools, which are used to re-index the data tables and for generating the video tree of objects.

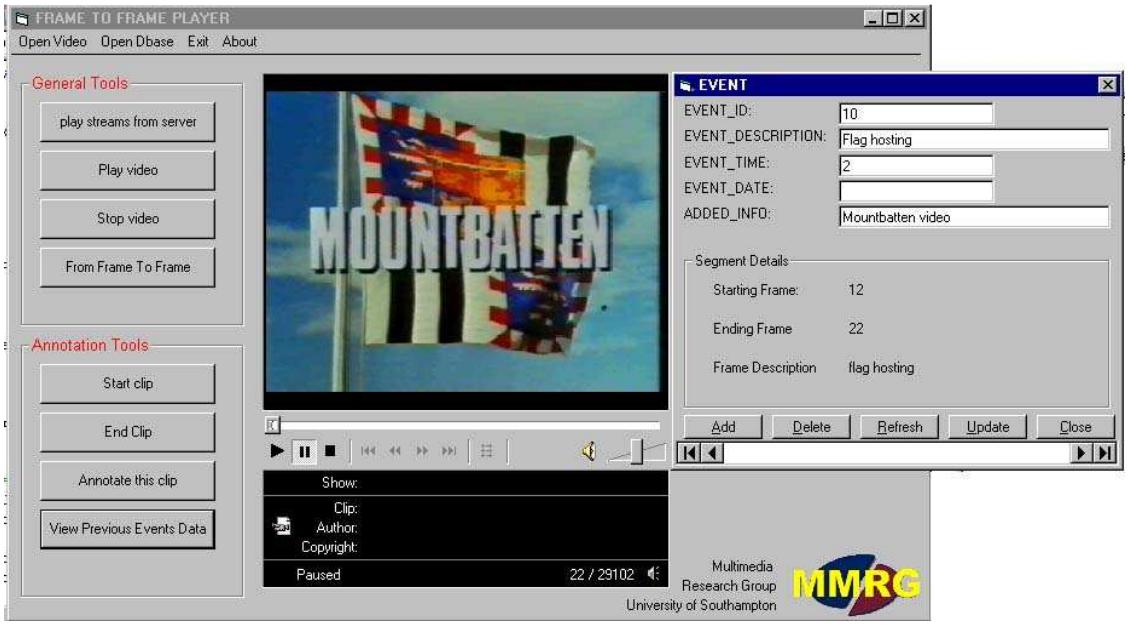


Figure 4.9: Event annotation editor

4.7 Data Query & Retrieval

The efficiency of a database is evaluated by the nature and complexities of the queries that will be made about the data. In terms of video, the query and retrieval process becomes more complicated because of the numerous demands placed on the system. Elmagarmid & Jiang (1997) describe a video data retrieval system in the following simple steps. First, the user specifies a query using a facility provided by the user interface. The query is then processed and evaluated. The value or feature obtained is used to match and retrieve the video data stored in the database. Finally, the resulting video data is displayed on the user interface in a suitable form.

4.7.1 Query types

Since video data is spatial and temporal, the queries are heavily dependant on their data content. Along with the architecture of the video data model and intended applications, there are also many other factors that modify a query. A query can be divided into:

Query by Content: These queries are further categorised as semantic information query (content information in the scene) , meta information query (scene description information) and audio-visual query (audio and visual feature of a scene).

Query by Nature: These queries depend on the nature of the video content and can be further categorised in spatial or temporal aspects of the video.

4.7.2 Query Certainty

The certainty of a query can be specified in terms of the type of matching operator used to satisfy the query. A query can fall into an exact match, inexact match, or similarity matched queries.

Hjelsvold (Hjelsvold & Midstraum 1994; Hjelsvold & Midtstraum, 1995), in his Video-STAR data model, defined a video query algebra, that allows the user to define complex queries based on temporal relationships between video stream intervals. These operations include normal Boolean set operations *(AND, OR)*, temporal set operations *(i.e. stream A equals stream B, A is before B, A meets B, A overlaps B, A contains B, A starts B and A finishes B)*, annotations operations that are used to retrieve all annotations of a given type and have non-empty intersections with a given input set and mapping operations that map the elements in a given set onto different contexts that can be basic, primary or video stream.

4.7.3 Query Processing

This process usually involves query parsing, query evaluation, database index search and the returning of results. In the query parsing phase, the query condition or assertion is usually decomposed into the basic unit and then evaluated. Along with text based search for annotations, feature based search is also applied to extract spatial contents like colour, motion, texture of the video data. Here thematic indexing will also help to retrieve data for a query. Next, the index structure of the database is searched and checked. Video data are retrieved, if the assertion is satisfied or if the similarity measured is maximum (a good fit). The results are usually displayed by a GUI, developed by the user.

The tv-DbMS query tool combines the support of a simple search and of the hierarchical thematic indexing search (discussed in Chapter 5). Users can query the database by specific keywords. The keywords can be based on a keyword index such as segment descriptions or event descriptions, camera motion, video titles, object of interest, person in a video or the location described in a video.

The tool also allows the user to browse the contents of the database by traversing the hierarchical description structure of the video document and at the same time to specify the keywords for search. This type of query is useful when a database has become very large.

4.8 Displaying the Query Results and Enabling User Selection

In response to a request for displaying the results of a video query, it is necessary for the tv-DbMS to be able to display the requested video clips. In the query output window, up to 12 video clips can be shown at a time, providing the user with a very wide variety of video clips. Then by clicking on any of the video clips, the user can view that particular clip. This also allows a user to view the whole video in a non-linear fashion, i.e. the user

can view the last segments of video before watching the first ones. This non-linear viewing of video clips is one of the novelties of the tv-DbMS, as none of the commercially available video players allows their users to watch them non-sequentially. Figure 4.10 depicts the output window of tv-DbMS. Here the user has searched for the query of "Air Bombing" in the Earl Mountbatten documentary. The tv-DbMS retrieved 12 video clips showing scenes of air bombing. On the left side, the details of clips are gathered from the metadata and at the top an SQL window is created automatically which shows the SQL statement generated for this particular query. Now a user can click on any of the twelve players to play a particular scene or can also look into the query, which is generated automatically by tv-DbMS, as in figure 4.10, shown for "air bombing".

Figure 4 10: Query output window of tv-DbMS

4.9 SQL Mapping Procedures

It should be noted that all the queries vary from each other depending on the level of the topic selected, the type of description and what sort of entity the user is querying for. Since the main tables in our database are event and segment, they should be present in every sort of query. If the user is asking for camera motion, or some other spatial related event, only the segment table can cater for the query, but if the user is looking for a particular object, or an event, then an inner join between the segment and event tables is created and the query is then entertained. An example of a simple search query is:

```
SELECT
        EVENT.EVENT_ID, EVENT.EVENT_DESCRIPTION, SEGMENT.SEGMENT_ID,
        SEGMENT.START_FRAME, SEGMENT.END_FRAME
FROM
        SEGMENT RIGHT JOIN (EVENT RIGHT JOIN EVENT_SEGMENT ON
        EVENT.EVENT_ID = EVENT_SEGMENT.EVENT_ID) ON
        SEGMENT.SEGMENT_ID = EVENT_SEGMENT.SEGMENT_ID
WHERE
         (((EVENT.EVENT_DESCRIPTION) Like "*"+[keyword search]+"*"));
```

When the user provides the keyword, an inner right join between event and segment is created. This fetches the data from the segment table corresponding to the particular event, then the segment start and end frames are also gathered from the segment table, which are then exported to the tv-DbMS visual module, which in turn displays those video clips in the query output window.

While resolving the thematic query (discussed in Chapter 5), the thesaurus module is used. The thesaurus also generates similar keywords. For example in the query:

```
SELECT
        THESAURUS.Word, THESAURUS.Syn1, THESAURUS.Syn2,
        THESAURUS.Syn3, THESAURUS.Syn4
FROM
        THESAURUS
WHERE
         (((THESAURUS.Word) Like "*"+[keyword_search]+"*")) OR
        (((THESAURUS.Syn1) Like "*"+[ keyword_search]+"*")) OR
        (((THESAURUS.Syn2) Like "*"+[ keyword_search]+"*")) OR
        (((THESAURUS.Syn3) Like "*"+[ keyword_search]+"*")) OR
        (((THESAURUS.Syn4) Like "*"+[ keyword_search]+"*"));
```

Here all the tuples of thesaurus are searched to find the similar words or synonyms and then these synonyms are provided to the query shown above in a loop to retrieve all the possible video clips. This is done as:

```
        For j = 1 to no_of_synonyms_found
                SELECT
                        EVENT.EVENT_ID, EVENT.EVENT_DESCRIPTION,
                        SEGMENT.SEGMENT_ID, SEGMENT.START_FRAME,
                        SEGMENT.END_FRAME
                FROM
                        SEGMENT RIGHT JOIN (EVENT RIGHT JOIN
                        EVENT_SEGMENT ON EVENT.EVENT_ID =
                        EVENT_SEGMENT.EVENT_ID) ON SEGMENT.SEGMENT_ID =
                        EVENT_SEGMENT.SEGMENT_ID
                WHERE
                         (((EVENT.EVENT_DESCRIPTION) Like "*"+[keyword
                search]+"*"));
```

```
        If keyword_search > 12
            // put the new search into another group
            Keyword_search = 1
        End If
    Next j
```

Step :2 All the retrieved video segments are then forwarded to the visual module, to display then accordingly.

Another way of getting results quickly is to combine the above two queries. This procedure requires a lot of processing and the computers with small memories (like on Pentium166 with 32MB RAM) failed to complete this query. The query is stated as:

```
SELECT
    EVENT.EVENT_ID, EVENT.EVENT_DESCRIPTION, SEGMENT.SEGMENT_ID,
    SEGMENT.START_FRAME, SEGMENT.END_FRAME, THESAURUS.Word,
    THESAURUS.Syn1, THESAURUS.Syn2, THESAURUS.Syn3, THESAURUS.Syn4
FROM
    THESAURUS, SEGMENT INNER JOIN (EVENT INNER JOIN
    EVENT_SEGMENT ON EVENT.EVENT_ID = EVENT_SEGMENT.EVENT_ID)
    ON SEGMENT.SEGMENT_ID = EVENT_SEGMENT.SEGMENT_ID
WHERE
     (((EVENT.EVENT_DESCRIPTION) Like "*"+[keyword_search]+"*" And
    (EVENT.EVENT_DESCRIPTION)=[THESAURUS].[WORD]));
```

Here the idea is to search the thesaurus table and the event table in parallel and retrieve the results. Since searching the thesaurus table and event table concurrently and on top matching them with each other, makes it a very slow process or even impossible for some computers with less memory.

4.10 Conclusion

In this chapter, we have defined that data model of tv-DbMS, and the description of entities attached to it. The tv-DbMS front end and the various database offered options at the front end are also defined in this chapter. Finally how SQL queries are mapped onto the database, was also been discussed. The next chapter deals with the novel indexing techniques of tv-DbMS.

Chapter 5

Indexing in tv-DbMS

This chapter describes the novel indexing techniques used in tv-DbMS. One of the approaches used is thematic indexing, where a theme is traced by using a similarity match and a pre-defined hierarchy of events and segments. The chapter also discusses the development of the video object tree, a method of creating the hierarchy of video objects in a video segment. The chapter ends by discussing how to track a story line by using the thematic indexing technique.

5.1 Video Data Indexing

Due to the huge volume of data in a video database, accessing and retrieving a particular item from a video becomes time consuming. For this reason, novel types of indexes are required to facilitate the process. In traditional text-based database management systems, documents are usually selected on one or more keywords that can retrieve the required data. Whereas in video database systems, it is not clear what 'key features' the database should be indexed on. These can be an audio-visual features, annotations, or other information contained in the video. Again, unlike textual data, video data indexes are difficult to generate or update automatically. The indexes need to

be closely related to how the video is presented (video data modelling) and the queries a user can ask.

Many researchers have tried to use the textual 'keyword' style techniques to create indexes for video data (Ahanger et. al, 1995; Bryan-Kinns, 2000; Caetano & Guimaraes, 1998). This keyword approach has been successful for organizing information in libraries and encyclopaedias and in the indexes at the backs of books, but has serious limitations when it comes to supporting the needs of retrieving conceptual or hierarchical information.

An example of the drawbacks of working with traditional indexing techniques can be looking for "car insurance" in the Southampton 00/01 yellow pages (Southampton Yellow Pages, 2000). This query will find nothing; similarly, words like "automobile insurance" or "vehicle insurance" will return nothing. However if you search for "motor insurance" you will get a lot of information, as the indexing used in yellow pages does not have any information about synonyms or the relationship between the words like automobile, vehicle or motor. A second example is putting "brake and clutch services" in the B section of the yellow pages, as the indexing is done alphabetically on the key words, but a conceptual and hierarchical index will put the "brake and clutch services" section under "car accessories" which is under the section "car". There is a cross link provided in the car accessories column in the main yellow pages, but when looking at the main index, where a yellow page user usually starts their query, the user will find no information about any natural hierarchy of objects.

These problems are typical of problems that occur in all indexes, whether in the back of a book, in the catalogue of a library, or on an online web search engine. All the above scenarios use text only databases and still the problems are there, so one can imagine the intensity of these sorts of problems, if classical keyword style indexes are used for accessing video databases. These problems grow exponentially while working with videos. This is due to the enormous storage requirement of the digital videos and the huge amount of content information. Another limitation of keyword indexes is that they

are static and do not evolve fast enough to adequately track the temporal aspect of the video.

In the early design of tv-DbMS, efforts were made to attach keyword indexes to the tv-DbMS data model, but the results strongly suggested that some novel form of indexing is required, which can tackle the following problems:

- It is not possible for a single keyword to describe both the spatial and temporal relationships of a video clip or a scene
- Specific keywords reduce domain space (the specific keywords used to describe the video data, the less chance the video content will match the query condition).
- A keyword, or a set of keywords, cannot fully represent semantic information in video data.
- There was no support for describing relations between the objects inside the video data.

To overcome these problems, the idea of thematic indexing was incorporated in the tv-DbMS model. Thematic indexing is based on the conceptual indexing model proposed by Woods (1995, 1997) also supports inheritance and semantics. This is achieved by creating multi layers of annotations within the video database and keeping a track of the hierarchy of the video data. A video object tree is also incorporated in tv-DbMS, which supports relationships between the video objects, in the video clip.

5.2 A Novel Approach to Thematic Indexing

Before going into the discussion of thematic indexing, certain definitions about conceptual indexing need to be established. Conceptual indexing is a technique to organise information to support subsequent access that can dramatically improve the ability to find information (Boris et. al., 1997). Conceptual indexing combines techniques from knowledge and natural language processing with classical techniques for indexing words and phrases in text to comprise a fast and efficient retrieval system. It does this by automatically analysing the conceptual structure of phrases extracted from the material,

and using semantic relationships between words and concepts to establish connections between the terminologies used by the user to ask a query (Brazilay & Elhadad, 1997).

An ideal conceptual index should automatically organize all the keywords and phrases of the indexed material into a "conceptual taxonomy" that explicitly creates relationships from each concept to its "most specific subsumers" and to other semantically –related concepts (Woods, 1997).

To overcome the problem of having many words with the same meaning, thematic indexing uses a thesaurus to expand a request by adding terms that are synonyms of the terms requested. Along with the synonyms, a user can also modify the thesaurus by making synonym sets in the database. For example, in the set: {automobile, car, truck, bus, taxi, vehicle}, the last element is clearly much more general then any of the other elements, but a user can modify the thesauri for his or her own convenience, by using the "add/modify thesaurus" option in the tv-DbMS front end.

The creation of a knowledge base of basic axioms is a very substantial job, and the lack of such basic knowledge has been a limiting factor in many conceptual indexing based applications. For tv-DbMS, Miller's WordNet (Miller et. al., 1993), which records basic relationships among more than 60,000 words is used. This thesaurus is further divided into nouns, verbs, adjectives and adverbs, and certain taxonomies are also used to match a particular word with its conceptual meaning.

Thematic indexing also uses a hierarchical structure, although both the synonym and hierarchical structures can work together or separately for a query. The hierarchical structure may include: part-whole relationships, geographical or political subdivisions of the video objects, hierarchical organizational structure, and arbitrary topic – subtopic relationships such as that between car and car-brakes. For example in the above set of cars and buses, the word automobile is on a higher hierarchy than the words car and taxi, and which are on the same level as truck and bus. But on the other hand, all these words are at a lower level to the word vehicle, as shown in figure 5.1.

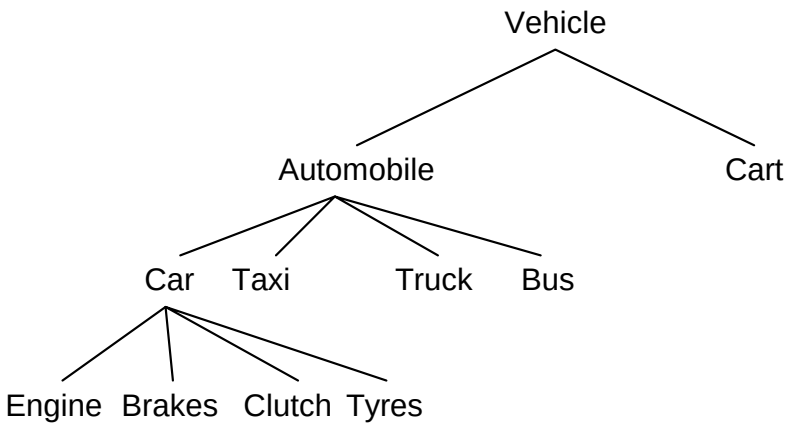


Figure 5.1: Hierarchy of vehicle

5.3 Building the Thematic Indexing and Retrieval System

In order to develop a thematic index, it is necessary that the application should have the following features incorporated into its database:

5.3.1 Lexical analysis

Lexical analysis is described as a knowledge base of syntactic and semantic information containing words and idiomatic phrases. This also has information about different senses of words and different interpretations of phrases. This knowledge base should also contain assumption axioms[5], which expresses generality relationships between concepts associated with known words and phrases and their senses. Word net is an example of a lexical reference system whose design is inspired by current psycholinguistic theories of human lexical memory (Miller, et. al, 1993). This sort of research started, when in 1985, a group of psychologists and linguists at Princeton University undertook to develop a lexical database. The initial idea was to provide a tool

[5] a statement accepted as true as the basis for argument or inference

that can help in searching dictionaries conceptually, rather than merely alphabetically. One of the advantages of WordNet over conventional dictionaries is that WordNet incorporates semantic organization for a particular word, rather then having its meaning only. This may increase the repetition of this word in different categories but it provides for all the possible contexts for that word.

5.3.2 Taxonomic Classification

The application should have a concept simulator that adds words into the conceptual taxonomy, and also determines any other related concepts and conceptual relationships that should be added.

5.3.3 Browsing and Retrieval

Browsing and retrieval process (in conceptual context) can be considered as a search and retrieval algorithm that can use the conceptual taxonomy. The conceptual taxonomy is used to make connections between terminologies used in a query and other related terminologies that might have been used in searching relevant material, which in turn can locate specific passages of video where answers to a request are likely to be found.

An interactive navigation system, which can display the portions of taxonomy and allow the user to move around in taxonomy, will be required. This will also felicitate in search of information and to move easily around the conceptual taxonomy and locations in the indexed material.

5.4 Developing Thematic Indexing

All of the above features are implemented in the tv-DbMS database's thematic indexing module. The database was developed using the MS-Access environment and the scripting code was written using MS Visual Basic version 6.0. The thesaurus for lexical analysis has been downloaded from WordNet. This thesaurus contains over 60,000 words, divided as verbs, nouns, adverbs and adjectives. On top of these divisions is a

lexical matrix that contains all the word meanings and word forms. The hierarchy module was developed so that for every data (entity or conceptual) entry, its parent node is also stored in the table. This simple method provides an unlimited number of hierarchy levels, and a hierarchy tree can be constructed at any time from this table.

In the context of tv-DbMS, the need to support an unlimited or a non-fixed number of hierarchies exists for two main reasons. The first is due to the nature of video, as a hierarchical description structure is intended to represent not only the contents in a video, but also to represent a complex video document that may have cross-references to segments of other videos. The second is due to the benefits it will provide for the implementation of hierarchical structures, especially when implemented in a relational database management system, as a non-fixed number of levels for managing digital video will improve the efficiency of storage and retrieval.

Figure 5.2, shows an example of how this thematic indexing model was used to achieve quicker and efficient results on the Earl Mountbatten video documentaries[6]. While working on the documentary of Earl Mountbatten, it was realised that two main themes substantially form the whole story. One theme is of Earl Mountbatten himself, his life, his career as a commander in the armed forces and as a dignitary in Burma and India. The second theme is about the war between the Allied Forces and the Japanese. A war is fought between nations, soldiers are involved, weapons are used and injuries and fatalities occur. Here the two themes combine as Earl Mountbatten himself is a soldier and a soldier fights a war.

[6] The Life and Times of Lord Mountbatten Series. Episode Name: "A March to Victory", produced by COLSTAR Group, UK.

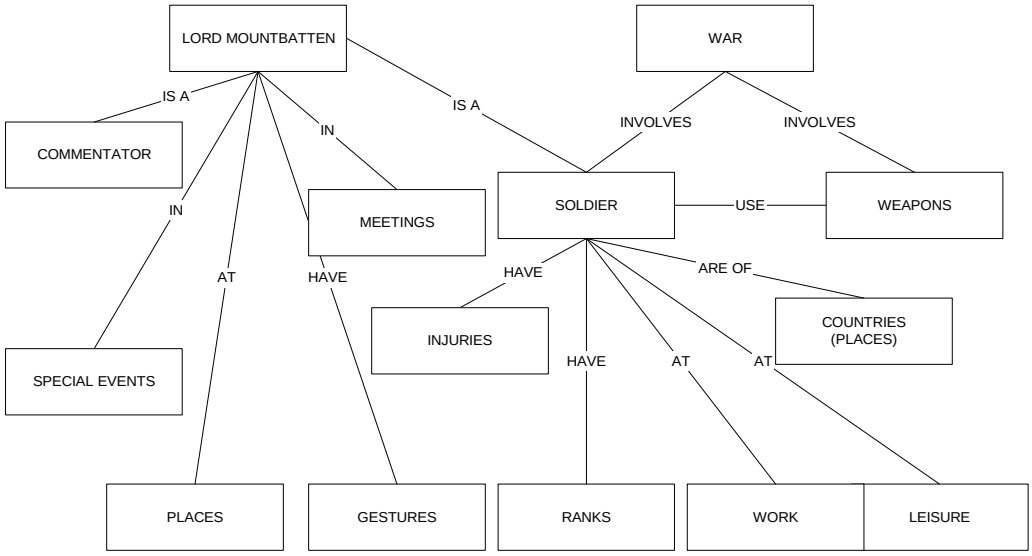


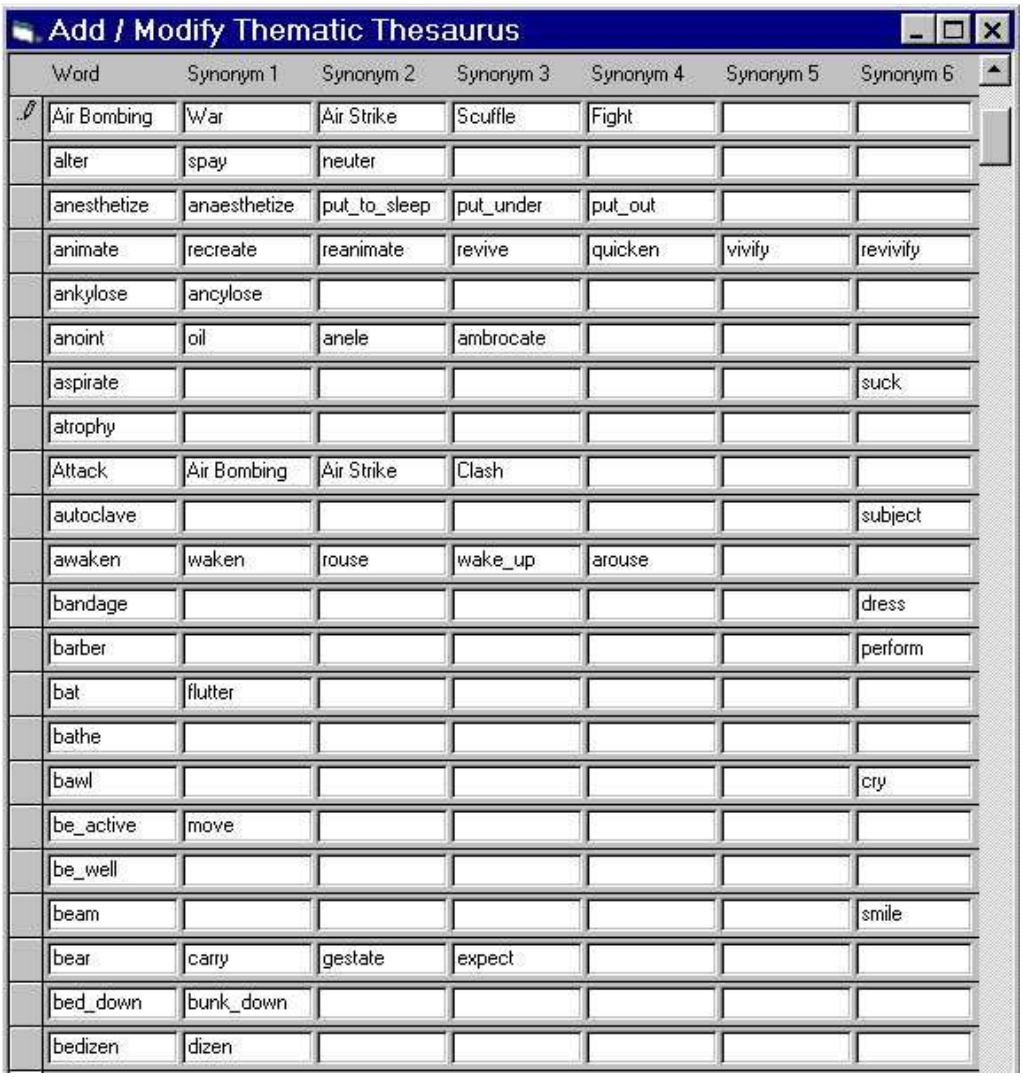
Figure 5.2: A snap shot of the thematic indexing model for the Earl Mountbatten's video

It should be noted that both the themes are implicitly related to each other, so indexing and query processing can easily be done at various levels in the hierarchy of entities. For example if a user makes a query about *air bombing*, this will come under the entity "weapons", which is a sub-part of the war entity. Another benefit of hierarchical modelling is that, at a certain level, if the search engine is unable to obtain some match, it can always go back to the parent level and fetch some related data.

Selecting the "Thematic Keyword Search" button, shown in the tv-DbMS interface (figure 4.9), creates this sort of query. This opens another window, where the required word is searched for using lexical indexing and hierarchical indexing and then returns the nearest matched options. These options are divided into nouns, verbs, adjectives and adverbs. By selecting one of those options, the user can view the video clips, which the system has retrieved by matching the new selection from the options to the video annotations.

A user can also modify the main thesaurus table, by selecting the database options from the main menu, and then selecting the "add/modify thematic thesaurus" menu, shown

in figure 5.3. This opens a new window, where a user can insert his / her own word meanings or concepts. However modifying the existing data may be quite dangerous, as the original word dictionary downloaded from WordNet is stored in the same database. So there is a strong possibility that the user may alter the original meanings of the standard words, which might result in erroneous output in queries afterwards.



Word	Synonym 1	Synonym 2	Synonym 3	Synonym 4	Synonym 5	Synonym 6
Air Bombing	War	Air Strike	Scuffle	Fight		
alter	spay	neuter				
anesthetize	anaesthetize	put_to_sleep	put_under	put_out		
animate	recreate	reanimate	revive	quicken	vivify	revivify
ankylose	ancylose					
anoint	oil	anele	ambrocate			
aspirate						suck
atrophy						
Attack	Air Bombing	Air Strike	Clash			
autoclave						subject
awaken	waken	rouse	wake_up	arouse		
bandage						dress
barber						perform
bat	flutter					
bathe						
bawl						cry
be_active	move					
be_well						
beam						smile
bear	carry	gestate	expect			
bed_down	bunk_down					
bedizen	dizen					

Figure 5.3: tv-DbMS thesaurus editor

5.5 Common Video Object Tree Model

Li, Goralwala, Ozsu and Szafron (John & Ozsu 1998; John & Iqbal, 1997) present the idea of a Common Video Object Tree (CVOT), where objects contained in a video frame can be accessed. To explain this tree model, the following axioms are required:

Let a video clip "C" be associated with a time interval $[t_s, t_f]$

Where t_s → starting time of clip

t_f → ending time of clip

Here t_s and t_f are relative (discrete) time instants and $t_s \leq t_f$

Since all clips have a start and finish time, a partial order can be defined over clips,

i.e. $C_i = [t_{si}, t_{fi}]$

Then the following axioms hold true in a digital video,

5.5.1 Partial Ordered Clips:

Let $C_i = [t_{si}, t_{fi}]$

$C_j = [t_{sj}, t_{fj}]$

Then $\prec$ is defined as the partial order over clips with $C_i \prec C_j$ iff:

$t_{si} \leq t_{sj}$ and

$t_{fi} \leq t_{fj}$

5.5.2 Ordered Clips

C_i is said to be ordered iff C is finite i.e. $C = \{C_1, C_2, \ldots\ldots, C_m\}$ and there exists a time order, such that $C_1 \leq C_2 \leq C_3 \ldots\ldots \leq C_m$

5.5.3 Perfectly Ordered Clips

A set of clips C = { C_1, C_2, …., C_m} is said to be perfectly ordered iff C is ordered and for some two neighbouring clips i.e. if C_i = [t_{si}, t_{fi}] and Ci+1 = [t_{si+1}, t_{fi+1}], we have t_{si+1} = t_{fi+1}

5.5.4 Strongly Ordered Clips

A set of clips C = {C_1, C_2, ….,C_m} is said to be strongly ordered iff C_1 < C_2 < C_3 ..< C_m

5.5.5 Weakly Ordered Clips

A set of Clips C = {C_1, C_2,….,C_m } is said to be weakly ordered iff C is ordered for two neighbouring clips, C_i = [t_{si} ,t_{fi}] and C_{i+1} = [t_{si+1}, t_{fi+1}] and we have t_{fi} >= t_{si+1} (∃i = 1,2,….,m-1)

Hjelsvold and Midtstraum (1994) also define some temporal operations on video clips. These operations are annotation dependent and can be expressed in a traditional query language. Some of these operations are:

C_i EQUALS C_j : Returns true if C_i and C_j are identical
C_i BEFORE C_j : Returns true if C_i happens before C_j
C_i MEETS C_j : Returns true if C_j starts with the next frame after C_i has ended.
C_i OVERLAPS C_j : Returns true if C_j starts while C_i is still active
C_i CONTAINS C_j : Returns true if C_j starts after C_i and ends before C_i
C_i STARTS C_j : Returns true if C_i and C_j start with the same frame and C_i ends before C_j
C_i FINISHES C_j : Returns true if C_j starts before C_i and C_i and C_j end with the same frame.

By looking closely at these operations, it is evident that they are similar to the above-mentioned axioms (section 5.5.1 – 5.5.5), however they are quite important for the foundation of tv-DbMS thematic indexing model, as they provide relationships between the video clips. Furthermore these axioms are used to perform operations required to query a video. For example the video operation C_i EQUALS C_j will achieve the same results as of a perfectly ordered clip axiom. The operation C_i OVERLAPS C_j will achieve similar results as of a partial ordered clips.

5.6 Objects in Video Clips

A video frame has a noticeable real finite number of common objects. These finite numbers of objects are catalogued for video segments and events in the tv-DbMS database (discussed in section 4.4 and 4.5). Let "C" be the set of all the video shots or video clips. Then the common objects for a given set of video shots are the objects which appear in every clip within the set. Then the set of common objects in *m* video clips can be defined as:

$$\{ \Upsilon(C_1) \cap \Upsilon(C_2) \cap \ldots..\cap \Upsilon(C_m)\}$$

where $\Upsilon(C_i)$ is the set of objects in Video clip C_i

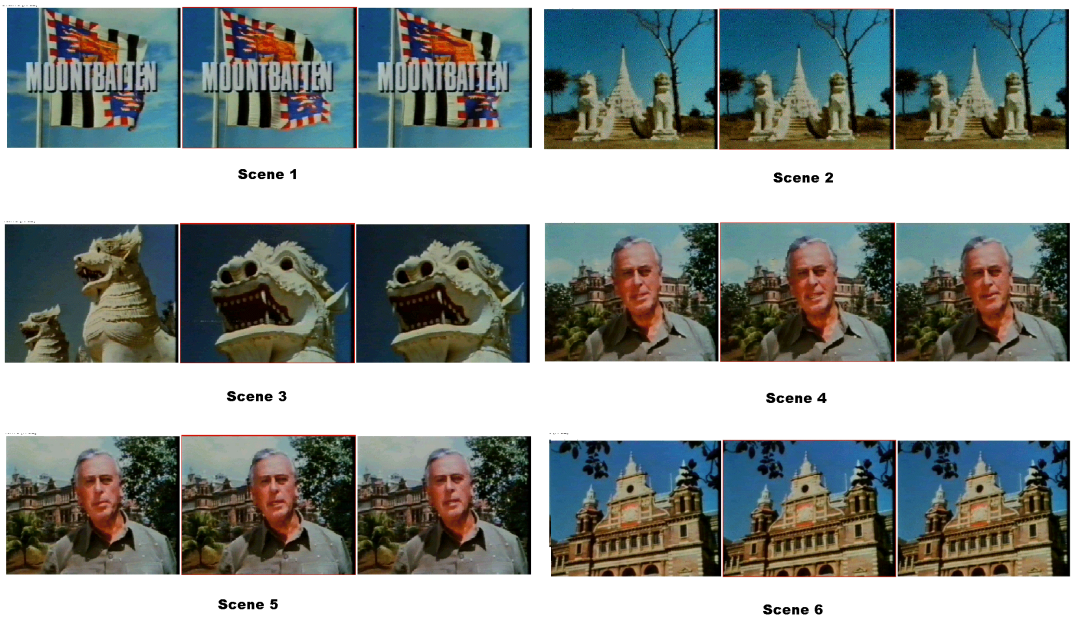


Figure 5.4: Noticeable objects in different video scenes

For example in Figure 5.4, we have six video shots, from the Earl Mountbatten video. Clip C_1 contains a flag, Clip C_2 contains titles, a tower and Burmese lions statues in the background. The other clips contain Earl Mountbatten, and a palace in Burma. This is expressed as:

$$C= \{C_1, C_2, C_3, C_4, C_5, C_6, C_7\}$$

Where,

$\Upsilon(C_1)$ = {Flag}

$\Upsilon(C_2)$ = {Tower, Burmese lion statues}

$\Upsilon(C_3)$ = {Burmese lion statues}

$\Upsilon(C_4)$= {Mountbatten, palace}

$\Upsilon(C_5)$ = {Mountbatten, palace}

$\Upsilon(C_6)$ = {palace}

From the object point of view:

Flag ϵ $\Upsilon(C_1)$

Tower ϵ $\Upsilon(C_2)$

Lion statues ϵ $\Upsilon(C_2, C_3)$

Mountbatten ϵ $\Upsilon(C_4, C_5)$

Palace ϵ $\Upsilon(C_6)$

Now using any of the axioms stated above we are able to develop a relational tree. For example if we use the intersection axiom, then we have,

$\Upsilon(C_1) \cap \Upsilon(C_2)$ = Null

$\Upsilon(C_2) \cap \Upsilon(C_3)$ = Burmese lion statues

$\Upsilon(C_4) \cap \Upsilon(C_5)$ = Mountbatten, palace

$\Upsilon(C_5) \cap \Upsilon(C_6)$ = Palace

Similarly we can build a hierarchical structure, such as:

$\Upsilon(C_4) \cap \Upsilon(C_5) \cap \Upsilon(C_6)$ = palace

In this way we construct an object tree, shown in figure 5.5.

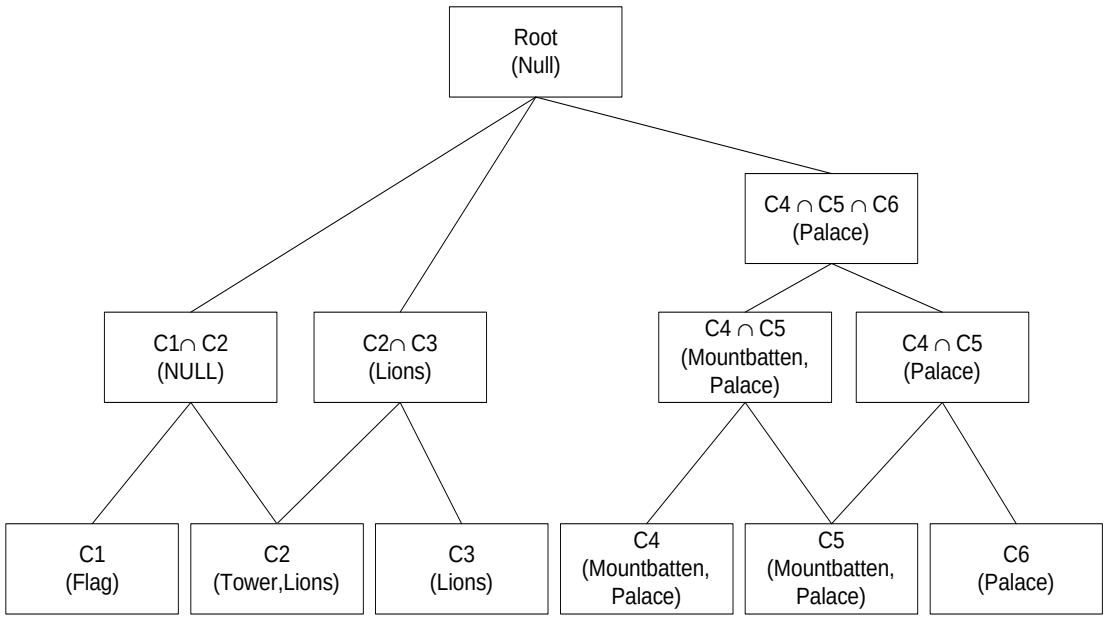


Figure 5.5: A Video object tree

Once the tree is established, a query can be processed, by using any of the axioms. For example in figure 5.5, perfect ordered clip axiom was used, and the query was to find all the video clips where object Burma Palace is shown. It should be noted that this sort of tree could be developed at the frame, segment or scene level. Once the tree is ready, the next step is to create relationship between these objects. For example Mountbatten is a Soldier. Here "is a" relationship that has been created between two objects 'Mountbatten' and 'Soldier'. This relationship will also provide themes of the story. In this way one can also follow the story line of the video with thematic indexing.

One can argue that using keywords only can solve a video query, in this case a query of finding the video clips containing the object "Palace". Though this argument is valid, if we look closely in figure 5.5, a keyword query will only be processed at the lowest level of the video object tree, where there is no relation between the video clips. One the other hand, the CVOT approach will cater for this type of query at a higher level where video clips have related contents. This relationship between video clips can be provided by the system to the user, who has initiated the query. Another advantage is the speed of processing using common video object trees, the amount of processing is reduced.

The objects defined in the CVOT provide spatial relationship between video shots. For example in figure 5.5 the location object "palace" is 'common' in video clips C4, C5 and C6. As discussed earlier (in section 4.4 and 4.5), the event entity in the tv-DbMS data model is a set of video segments, which convey meaning. Whereas the event entity also contains the annotations for person object, location object and object of interest (shown in figure 4.7.). The event entity provides temporal relationships between video clips.

This entity also provides the conceptual relationship between the video clips. This is achieved, by annotating the concepts related to the event entity (discussed in section 6.2.3). Once these relations are obtained, the next step is to generate the thematic index. At this point the thematic index contains all the relations and concepts between the video clips. For example, in the thematic index shown in figure 5.2, the relation 'War involves soldiers' is first achieved by first generating the CVOT using the perfectly order clip axiom for the concept "war" and the person object "soldier". The concept "war" is used to annotate the events which contain World War II segments in the Mountbatten videos.

5.7 Conceptual Story Tracking By Using Thematic Indexing and Video Object Tree

Editing a movie (whether online or offline) has always been a very artistic job, and no standard rules are used. It very much depends on the creativity of a particular editor or director as to how he or she wants to mould the story or how the shots are to be arranged. This problem again intensifies, as every editor or director has a very individual style of editing a movie. So it becomes nearly impossible to develop algorithms that can track a story or a part of story from a video clip.

Mandler (1984) in his book "Stories, Scripts, and Scenes: Aspects of schema theory", suggests that a story schema depends upon the hierarchical structure of level events with their serial structure in conjunction with some location or object of interest. He describes story as a literary expression that a human can read or hear (in the context of videos, this definition should be modified to: a literary expression that a human can read, hear or see). Mandler describes an event as a hierarchical set of units describing generalized knowledge about a sequence, which includes knowledge about what will

happen (or is happening) in a given situation, and often the order in which the individual events will take place.

Considering Mandler's definition with respect to documentaries such the Mountbatten's videos, we can see that a query about Mountbatten himself is not an event, but "Mountbatten carrying a gun" is an event, and "Mountbatten carrying a gun in a war" can be the basis of a story. So a query such as "Mountbatten + Gun + War" will reveal particular war scenes, which if watched linearly will provide an initial form of story.

While working on the Mountbatten videos using tv-DbMS, attempts were made to track the story line by using thematic indexing and video object tree (discussed in chapter 6). For example, by looking at figure 5.2, it is evident that Mountbatten is a soldier and has a rank (in this case a commander) and is in meetings. On the other hand the video object tree efficiently tells us in which scenes the location object "palace" (in this case, a palace in Burma) is present. So a query of the form "Mountbatten + Soldiers Meeting + Palace" will provide us with all the segments that have the above-mentioned entities present in them. This is also evident by looking at the common video object tree in figure 5.5. Here the importance of the CVOT becomes evident, as by picking up the higher node (i.e. $\Upsilon(C_4) \cap \Upsilon(C_5) \cap \Upsilon(C_6)$ = palace) it provides all the video clips that have location object "Palace". Viewing these video clips linearly can provide a story track about Earl Mountbatten presiding over a meeting in Burma.

Another problem in tracking a story line is maintaining and retrieving the hierarchy of events. That is while retrieving a complex query, they system may retrieve some other sub-events which are not part of the required hierarchy of events. For example, if a user while using thematic indexing techniques asks the system for "Mountbatten drinking wine" and is expecting that the system will reveal a story about "Mountbatten attending a party in London", may get some scenes about "Mountbatten visiting the Indian leaders", as the sub-event of "drinking wine" is common in both event hierarchies. Whereas if the user asks for "Mountbatten + drinking + London", this would had retrieved all the relevant scenes about attending a party in London.

The types of queries stated above are very useful while tracking a lower level story, but not while tracking high level stories or stories spread over various events and episodes. This is due to the fact that the taxonomic knowledge spread between long stories has a complex structure and is also based on many conceptual ideas that are hard to understand even by a human, let alone machines. Nack and Parkes (1997) in their system "AUTEUR", define that a story is a combination of episodes, sequences, scenes, actions and sub-actions. This sort of story is considered as a high level story.

From the examples shown above, it is evident that tv-DbMS is able to retrieve lower level story lines, however its one of the future tasks to enable tv-DbMS to track higher-level stories.

5.8 Summary

This chapter dealt with the indexing techniques used in the tv-DbMS data model and included detailed discussion on developing thematic indexing. A methodology for developing a video object tree and how to use particular video axioms was also discussed. Finally, the chapter demonstrated how the thematic indexing technique can be used to track a story line. The next chapter will present an evaluation of some results of working with the tv-DbMS system on documentaries.

Chapter 6

Evaluation I: Documentaries

This chapter deals with the evaluation of tv-DbMS, and its thematic indexing technique. The work started with digitizing three, forty-minute episodes, "Life and Times of Lord Mountbatten" a documentary series made by Colstar Group, consisting of 12 programs of approximately one hour each, narrating the biography of Earl Mountbatten of Burma (1900 – 1979), the last Viceroy of United India and the First Governor General of Independent India. The reason for selecting this documentary for tv-DbMS experiments was that it contains a mixture of old black and white movie clips and new colour video clips. The majority of black and white video clips were recorded by Earl Mountbatten himself (some by his family members) and are relatively unstructured. Since this series was made in the mid seventies, no closed captions or sub-titles are available for the documentaries. Hence, these documentaries were perfect for experimenting with tv-DbMS, as they do not have any structured format as compared to structured CNN or BBC news clips. They have a mixture of black and white and colour video clips. There is a running commentary by Early Mountbatten himself, but this is again relatively unstructured.

The first part of this chapter deals with the experiments carried out and the results achieved using the Earl Mountbatten videos, and the second part deals with the hyperlinks created in the video document.

6.1 Building an application

This section discusses the necessary steps required to build a tv-DbMS video application. The application development process starts from capturing the video, then analyzing the contents, synthesising the contents (i.e. providing metadata), computing the inter-relations between them, developing the integration between the metadata about the video clips and finally delivering it to the user. These steps are discussed below:

Capture: During this process, a video archive is converted from analogue media to digital media. Here the user has the option to use any compression format like avi, mpeg, quicktime or asx. However, it is recommended to use MPEG, because of its compression efficiency and portability to any operating system.

Analysis: In this phase, automatic segmentation is done (discussed in chapter 4). During segmentation, the continuous digital video is divided into small segments. These segments are generated by looking at the histogram difference of each frame. Then every segment is given an auto generated unique identifier, which is later stored in a database.

Synthesis: During this phase, annotations are provided for the video segments. This phase can also be considered as the database development phase, as all the annotations and metadata are provided during this phase.

Computation: The synthesis and computation phases can be considered as one, as they both relate to metadata provided to the system. In the computation phase, modules such as the common video object tree developer, and the thematic indexing developer are used to compute the relations and hierarchies between the video objects and video clips.

Integration: Query development and database integrity are the two main modules of this phase. Indexes and lookup tables are automatically generated, once the annotator has provided annotations. At this point, the system is ready to handle video queries.

Delivery: In this phase, the user interacts with tv-DbMS and performs queries. These queries can be simple or complex, depending on the type of information the user requires.

6.2 The Earl Mountbatten Videos

6.2.1 Framework

Using a Creative Video Blaster Series 300, we digitised the documentaries of Earl Mountbatten and stored them in Microsoft's avi (Audio Video Interface) format. The codec used for compression was Indeo 3.2. We digitised the three episodes. The first episode "The March to Victory" is about the World War2, where Earl Mountbatten was stationed in Burma, as the Supreme Allied Commander, for South East Asia. The result is an avi file of 498 Mbytes. The second episode "United We Conquer" is also about World War 2, and the avi file is of 432 Mbytes. The third episode "The Azure Main" is about the period where Earl Mountbatten was Viceroy of India, and the digitised avi file of this episode is of 520 Mbytes. The metadata was stored in Microsoft Access '97 database and ODBC (Open Database Connectivity) components were used to connect the database with the main application.

In the first step, the whole video was segmented using code written in Visual C++ (discussed in section 4.3). This segmentation module looks for the RGB histogram of difference of each video frame and creates video segments by comparing the difference of histograms of corresponding frames. The segmented frame numbers are stored in a database, and a unique auto-number is also generated for every segment. This automatically generated number will work as the primary key for a particular segment throughout the rest of the metadata development.

Once the segments are generated, an annotator can provide the relevant metadata by using the annotation module in the tv-DbMS front end (discussed in section 4.6), by dividing the segments into corresponding events.

6.2.2 Event Creation and Annotations

The next step is to create the events (discussed in sections 4.5 and 4.6). The parameters for the event entity are Event_id (auto generated number), Event_time, Event_date and Event_added_info. Event_time and Event_date are redundant parameters. For

Event_added_info, the annotator provides the information about the cross-references for connecting this entity with other events in the database.

Figure 6.1: Event annotation module of tv-DbMS

Figure 6.1 shows the event annotation module, where the annotator has provided the annotations for the event namely "Mountbatten in soldiers uniform", during "WW2". In the tv-DbMS data model (discussed in chapter 4), a video segment entity has a many-to-many relationship with the event entity. This is normalised by using a dummy "event – segment" entity. Once the annotations and the metadata have been provided, a tuple (row entry) of the segment and event tables contains enough data to perform a simple query on videos. An example of single data entry for segment table is shown in table 6.1.

Creation of annotations is a time consuming job. For Mountbatten videos, the annotations were created by the author, and it took 15 hours (approx) to annotate one 50-minute episode. One can argue about the enormous number of hours spent to annotate the video, but this can be justified by the fact that the author was not an expert in this field. The amount of hours spent will reduce rapidly, once the annotator receives some sort of training and becomes familiar with the tv-DbMS system. However, it can also be argued that providing metadata is a one -time job only, which should be done at

the time of creation of the video (i.e. under the supervision of video director). This metadata will be used later by a huge number of viewers. This argument is further discussed in chapter 8.

6.2.3 Generating the Thematic Index and Video Object Tree

For thematic indexing, a video object tree is required. Figure 6.2 shows the front- end of the video object tree builder, which is used in the tv-DbMS software. In this module, an annotator can provide the metadata for person, location, concept and event, for a particular segment or a key frame. For example, in Figure 6.2, the segment ID is 991 and its description is "Mountbatten Meeting". The people involved in this scene are Mountbatten himself and some soldiers. It is a scene from World War 2, where Mountbatten is "presiding over" the meeting and "reading" a document. The segment is also interconnected with segments 1023 (Mountbatten) and 75(World Ward II). All these segments are 'part-of' event no. 157(Meeting). The video objects of this video segment are strongly ordered. The relationships between segments and events are provided by the annotator in the in the event annotation module, shown in figure 6.1.

In figure 6.2, the segment 991 was cross-referenced to the event 157 in the event annotation module (figure 6.1). Other segments that are also cross-referenced with this event are 1023 and 75. This information was automatically extracted from the seg-event table and provided to the video object tree builder module. The noticeable objects have been annotated, such as "Mountbatten ", "soldiers" as the person objects and the location is "World War II". These objects are strongly linked (ordered) to the objects of other segments, which are cross-referenced for the event 157.

While generating the annotations, it is totally up to the annotator, to annotate when objects are present in the video clip. For example in figure 6.2, the annotator has ignored or had no knowledge about the names of the other soldiers present in the meeting and has not at all considered the soldiers standing in the background. Other annotators might consider them more important.

Figure 6.2: Video object tree builder

6.2.3.1 Annotating semantics to facilitate thematic indexing

One of the additional features of tv-DbMS is that its database has additional columns to store semantics (or concepts) concerning a particular video segment or event. Concepts can be expressed in a few keywords or can be as long as a few sentences, depending upon the annotator. An example can be seen in figure 6.2, where in video segment no 991, Earl Mountbatten is 'presiding' over a session and 'reading' some documents. These entities are stored as concepts. This concept entity can also be taken at a higher level of abstraction and can be related to previous or subsequent video segments or events. For example in the same video clip (figure 6.2), Earl Mountbatten is discussing the 'role of Mr. Gandhi's civil disobedience movement in Indian politics'. Though 'Mr. Gandhi' is not present in the video clip, the annotator can insert the entity of 'Gandhi' in the concept box, which can later be used for thematic indexing.

Text annotations or description represents a rich concept that can be related to several levels of abstraction. Annotations vary according to types of data and abstraction. Furthermore, different types of annotations are necessary for different purposes of the

categorisation. It can be argued that these annotations could be produced automatically by analysing the visual aspect and spatial domain of a video clip. But, as demonstrated in the above example, it is currently not possible for an algorithm or automatic convention to deliver as higher level of semantics as a human annotator can interpret.
Once a video document is ready; we can undertake different experiments while using tv-DbMS. We divided these experiments into three different stages, i.e. performing a simple query, performing a thematic query and finally tracking the story line of the video.

6.2.4 Results

In tv-DbMS queries are divided into two types: simple queries and thematic queries. It is very difficult to show the results of video retrieval in textual form. However, by using screen shots, we have tried to show the results of tv-DbMS.

6.2.4.1 Performing Simple Video Queries

In terms of tv-DbMS, a simple video query simply matches the user input with the annotations of the video clips and retrieves the matching video clips. This is shown in figure 6.3, where a simple query has been generated about "Air Bombing" in the Earl Mountbatten videos. In the figure 6.3, the top window contains the query generated for retrieving the video clips. This type of query is discussed in detail in section 4.8 of this thesis. Figure 6.4 shows the video clips retrieved when a query about "Japanese Soldiers" was performed in the Earl Mountbatten's archives. Tables 6.1 and 6.2, show the annotations for the first video clip obtained, in figure 6.4. By clicking on any of the video clips the user can view the particular clip. By right clicking the video clip, the user can get options such as zooming the video, viewing the video at full screen, viewing the video data, changing the options for the player and selecting a particular codec to view a video clip. The right click options for the tv-DbMS viewer are shown in figure 6.5.

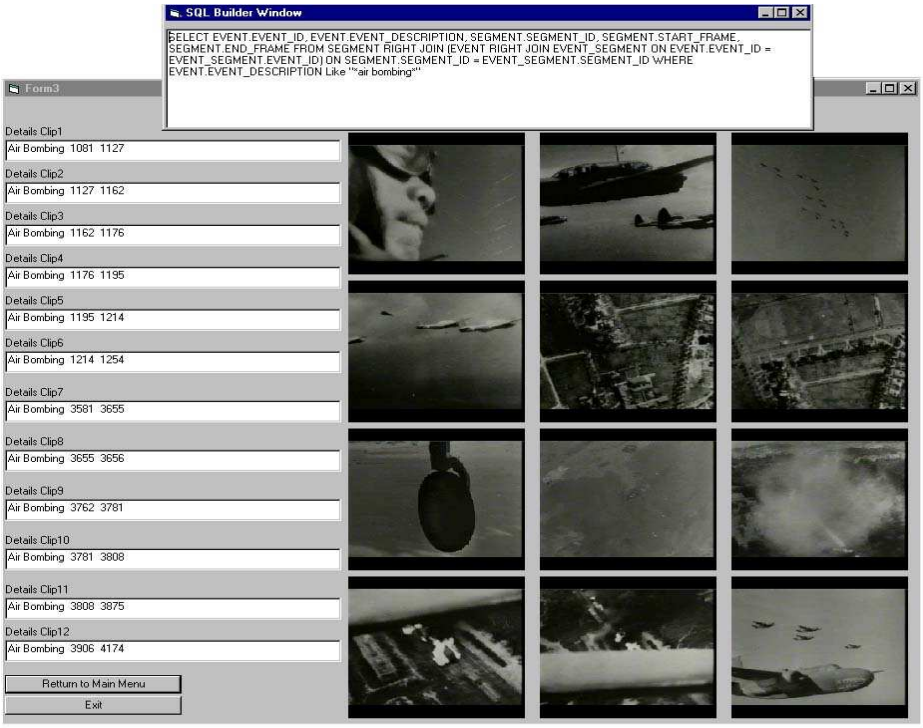


Figure 6.3: Simple query for "Air Bombing" in Earl Mountbatten videos

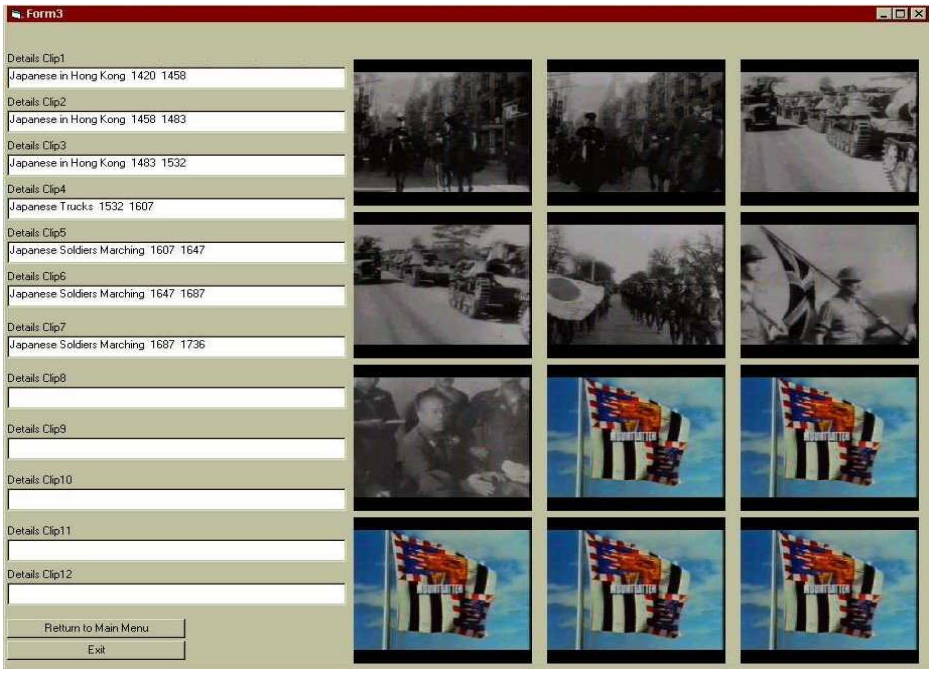


Figure 6.4: A Simple query about "Japanese Soldiers" in the Earl Mountbatten videos

SEGMENT TABLE	
SEGMENT_ID	43
START_FRAME	1081
END_FRAME	1127
PERSON	----
LOCATION	----
OBJECT	Planes
CONCEPT	Air Bombing, Japanese Planes
MOTION_TYPE	Simple
CO_ORD_SYSTEM	XY
SPATIAL_UNITS	NA
CAMERA_FOLLOWS	YES
CO_ORD_ARE_3D	NO
KEY_POINTS	-----
2D_3D_SPATIO TEMPORAL	2D
INTERVAL_3D	None
INTERVAL_2D	None
INTENSITY_OF_ACTIVITY	Fast
DIRECTION_OF_ACTIVITY	Straight
TEMPORAL_DISTRIB_OF_AC	None
SPAITAL_DISTRIB_OF_ACTIVITY	None

Table 6.1: Segment tuple stored in tv-DbMS database

EVENT TABLE	
EVENT_ID	6
EVENT_DESCRIPTION	Air bombing
EVENT_CONCEPT	
EVENT_TIME	
EVENT_DATE	
EVENT_ADDED_INFO	Bombing

Table 6.2 An Event tuple stored in tv-DbMS database

Figure 6.5: Right click options for tv-DbMS viewer

SEGMENT TABLE	
SEGMENT_ID	82
START_FRAME	1420
END_FRAME	1458
PERSON	Soldiers
LOCATION	Hong Kong
OBJECT	----
CONCEPT	----
MOTION_TYPE	Simple
CO_ORD_SYSTEM	XY
SPATIAL_UNITS	NA
CAMERA_FOLLOWS	YES
CO_ORD_ARE_3D	NO
KEY_POINTS	-----
2D_3D_SPATIO TEMPORAL	2D
INTERVAL_3D	None
INTERVAL_2D	None
INTENSITY_OF_ACTIVITY	Slow
DIRECTION_OF_ACTIVITY	Straight
TEMPORAL_DISTRIB_OF_AC	None
SPAITAL_DISTRIB_OF_ACTIVITY	None

Table 6.3: Segment tuple for first result shown in figure: 6.4

EVENT TABLE	
EVENT_ID	20
EVENT_DESCRIPTION	Japanese In Hong Kong
EVENT_CONCEPT	
EVENT_TIME	
EVENT_DATE	
EVENT_ADDED_INFO	Japanese Army

Table 6.4: An Event tuple for first result shown in figure: 6.4

The results shown in figure 6.3, are based on normal searching of the annotations of segments and events. For example in figure 6.3, the user has searched for the keyword "air bombing", and the segment and event entity for the first result obtained is shown in table 6.1(for segment) and table 6.2 (for event).

6.2.4.2 Performing Thematic Queries on the Mountbatten Videos

It is said that a picture is worth thousand words. If this saying holds true, then a one second video clip with 25 frames per second is worth 25000 words. This illustrates the scale of the problem of using keywords to describe the video. In addition, users can have different perspectives while watching a video clip (Salam, 1996). One of the applications of thematic indexing and CVOT can be explained by this example. Consider a video document about a car accident. The police might be interested in finding out the details of the scene, the paramedic staff might be looking for the people who are injured in the accident, and the insurance company will be looking for the initial part of the video clip to find out whose fault it was! This sort of video clip can become even more useful, if some sort of metadata is provided along with the video clip. This metadata then can be formatted to the users personal requirement. Thematic indexing on the other hand is a tool to organize the metadata in a hierarchical way and by providing relevant words from the tv-DbMS synonyms database. The vehicles involved in the scene will be

considered as object_of_interest and are stored in the segment-event table. Assume that the accident scene is based on many video clips and that one of the vehicles involved has been shown in some other video clips, which are not the part of the accident scene. A normal keyword search for the vehicle will retrieve all the video clips regarding that vehicle, without considering the user's interest, whereas the thematic indexing search will only retrieve the accident clips. The query provided here will be "vehicle + accident", where accident is stored as a concept. To make this example more complex, we assume that the above-mentioned vehicle was first owned by Mr. A, and then by Mr. B, and when the accident happened, the vehicle was the property of Mr. B. Now if someone is interested in viewing the status of the vehicle before and after the accident and under the ownership of Mr. B, then a thematic query like "vehicle + Mr. B" will retrieve the video clips. This query will be using the same CVOT, but here the concept "accident" is made redundant. The advantage of using thematic indexing is the retrieval of precise video clips, in a very efficient manner as compared to the normal keyword searching.

Another example of the application of metadata is in large video archives. For example, all the TV companies have huge archives of their programs. The normal practice to store these archives is to put some labels on the magnetic media and then the labels are stored in a database. These labels usually have the name of the program, date and time when aired, people involved, and location of the program. If a user is looking for any video clip about a particular subject, he or she may not get much help from the names of the programs or the time when it was aired. Thematic indexing in addition to the standard catalogue can tackle all the complex queries, where the user was not sure about the query. Thematic indexing while categorizing the metadata into hierarchies and video object trees, also enables the user to be more flexible with the keywords for which he or she is searching and can provide a broader scope of video results.

In the thematic query, the user is provided with a thesaurus table and with the video object tree to select the query. In addition, in the concept attribute of the segment table, the annotator can also provide semantics to the database. Semantics for events are also

stored in the event database they are used to create relationships between two or more events and to detect a particular story line in the video clips. An example for the single entry of Event (Air bombing in this case) is shown in table 6.2.

The thematic indexing query is initiated by the user clicking the "thematic indexing" search button in the tv-DbMS front end (discussed in section 4.6). This opens a window with a thesaurus and a search button. The data generated by the video object tree is also inserted in the thesaurus table. This is done to simplify the thematic search, as otherwise the database has to search on two separate tables, whereas creating a simple lookup table in the database can speedup the search process. The thesaurus used in this database is picked up from the WordNet project (Miller, et. al, 1993). Once the trees and indexes are generated, the user is free to perform any query. Figure 6.6 shows the window of the thematic indexing module. Here the user can type in any key word, then the similar or synonym words from the thesaurus are shown. Then from clicking on any of the words provided by the thematic indexing model, the user can perform the query on the video clips.

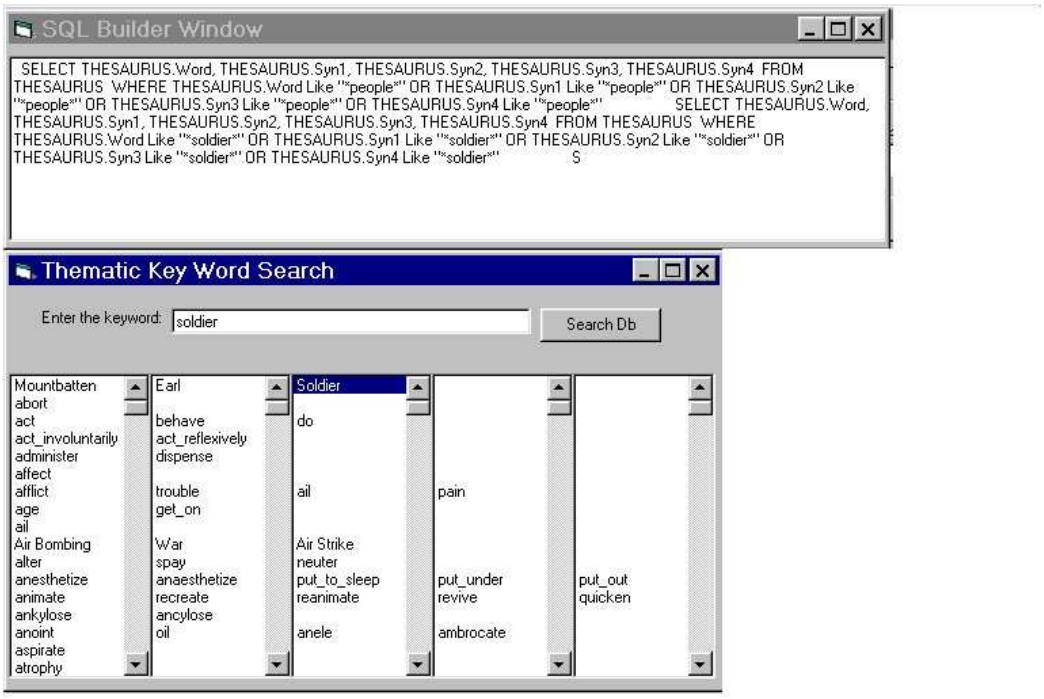


Figure 6.6 Thematic indexing for the term "Soldier"

The results in figure 6.7 shows the video clips of Earl Mountbatten as a soldier. These were gathered by using thematic indexing and searching for Mountbatten and Soldiers. In the hierarchy tree, (shown in figure 5.2) Mountbatten is a soldier, so we get the video clips in which Earl Mountbatten is involved as a soldier.

Thematic index search primarily depends upon the annotations for events and segments. For example in figure 6.1, an annotation of "Mountbatten in Soldiers Uniform" is provided by the annotator. Then the information about video objects present in the scene is extracted from the segments and event annotations to develop a common video object tree. For example, figure 6.2 shows the video objects gathered from segment '991', entitled 'Mountbatten Meeting'. This segment is a part of event 'Meeting'. Once the video objects from all the annotations are collected, by joining different events through ordering techniques (discussed in section 5.6), a common video object tree is generated. This object tree also defines hierarchies of video objects, and is stored in the thematic index database, along with the thesaurus (dictionary), shown in figure 6.6. It can also be observed in figure 6.6 that, 'Mountbatten' has the title 'Earl' and was a 'soldier' (hierarchy shown in figure 5.2). The cross-reference between 'Mountbatten' and 'soldier' was developed by using the strong ordering technique between events '159' (Mountbatten – soldier, shown in figure 6.1), event '157' (Mountbatten – meeting – soldier), and the other events related to the military life of Earl Mountbatten.

In the same way, figure 6.8 shows two video clips of Lord Wavel (former Viceroy of India). This result was generated by making a slight modification in the thematic indexing search and instead of searching for Earl Mountbatten as a soldier, we searched for Lord Wavel as a soldier, and two clips of Lord Wavel wearing a soldier's uniform are retrieved. Here we are using the same CVOT, but instead of Earl Mountbatten, we asked the system to search for Lord Wavel.

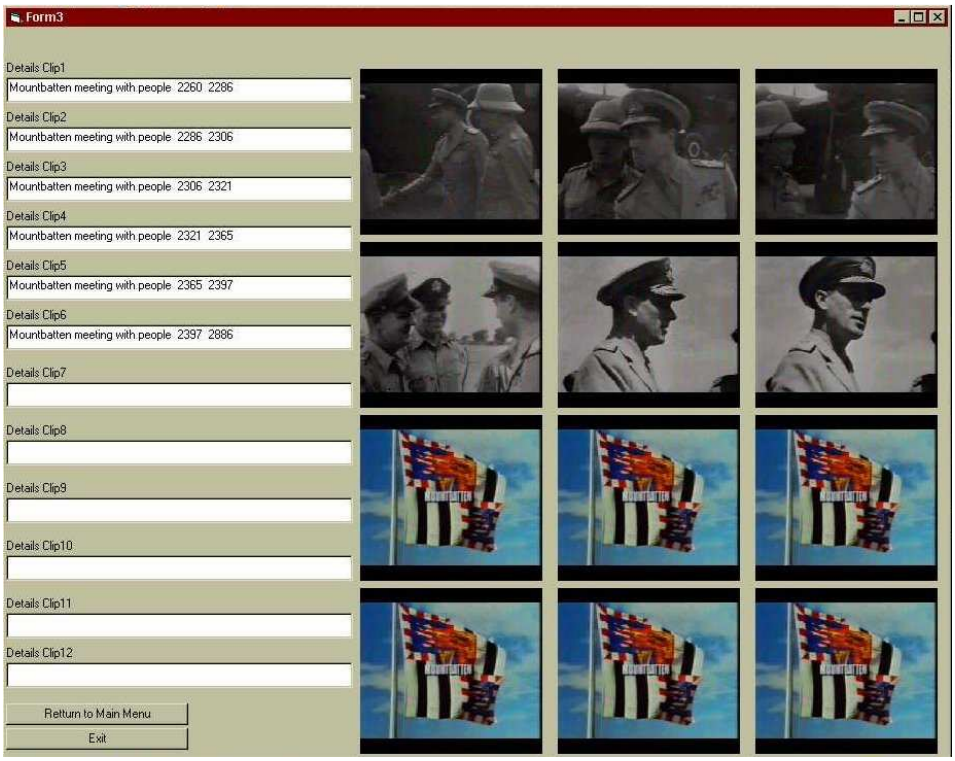


Figure 6.7: Thematic indexing search for Earl Mountbatten as a soldier

Figure 6.8: Thematic indexing search for Lord Wavel as a soldier

6.2.4.3 Tracking the story line

As discussed in section 5.7, we try to find some events with a subject (i.e. person or a concept), a verb (an action word) and some object (again a person, a concept or a location), and apply the thematic indexing query on it. Here we used the query of soldiers marching in a jungle. The soldiers are the subject, marching is the action word and jungle is considered as an object.

A slight modification was made in the tv-DbMS query retrieval, so that tv-DbMS can store the output of a query in a temporary file, and perform another query on the metadata of the video clips generated from the first query and stored in the temporary file. Hence, the query for 'Soldiers marching in a jungle' was broken into two queries. First, the query 'soldiers marching' was run and the output was stored. Then by using the thematic index hierarchy a search was made on the stored output about the word jungle. The following results were generated by tv-DbMS (shown in figure 6.9):

Figure 6.9: Soldiers marching in a jungle

6.3 Integration of hypermedia links to tv-DbMS

This section deals with the integration of hyperlinks in the video data. The first step is the creation of links from video, and the next step is to provide the facility of "browsing" access to the video data. tv-DbMS is a video database system that uses relational and object oriented techniques to access continuous non-textual media. These techniques use a special purpose built thematic indexing model to retrieve video queries using semantics and concepts.

On the other hand, hyperlinks allow users to browse the information and access it according to the user's particular subject of interest. Hyperlink systems as compared to traditional database systems provide an ad-hoc jump or non-linear access to information. Another application of hypermedia systems is to create interactive

documents. For tv-DbMS documents, the of hypermedia system feature is implemented in the browsing or navigation of video, and the second feature is obtained while creating or authoring links inside the video clips.

6.3.1 Open Hypermedia Systems

Open hypermedia systems[7] address the issues of integrating hypermedia functionality into existing applications in the computing environment. An Open Hypermedia Systems(OHS) is typically a middleware component in the computing environment offering hypermedia functionality to applications orthogonal to their storage and display functionality. To become "hypermedia enabled", applications must be extended to make the hypermedia functionality available in the user interface and must be able to communicate hypermedia requests to OHS. The term open hypermedia is used to cover both the OHS and the set of hypermedia enabled applications. An open hypermedia environment is a subset of the overall computing environment in terms of applications, programs and services (Garzotto et. al, 1995).

An important matter in hypermedia systems is the distinction between structure and content. A hypermedia system that imposes a specific data model format (specifying both structure and contents formats) on its hypermedia-enabled applications can be considered closed. For example, a simple html document with embedded links will be considered as a closed hypermedia application. Allowing applications to store content in different formats potentially outside the hypermedia system is a basic requirement for integrating and using existing applications in an open hypermedia environment (Lowe & Hall, 1998).

An open hypermedia system allows an open set of applications to participate in the hypermedia services and supports an open set of data model formats.

[7] In Open Hypermedia Systems, the links and their management are kept separate from the main document. The Hypermedia management system becomes much more of a backend process then a user interface technology, as compared to closed hypermedia systems, that provide a fixed set of encapsulated applications which are normally tightly integrated with the hypermedia linking mechanisms (Legget, et. al, 1993)

6.3.1.1 Link Traversal in an Open Hypermedia Environment

1. The application communicates a traverse link request to the open hypermedia system due to some action, which can be triggered by a user click on an anchor or by some other event taking place in the application.

2. The OHS resolves the link and determines the file(s) at the other end of the link. Some OHSs support n-array links in which case step 3 will be repeated for each file at the other end of the link.

3. The OHS can now do one of the following things to display the file:
    a. It can request that an already running hypermedia enabled application display the file
    b. It can launch a hypermedia enabled application to display the file
    c. In case no hypermedia enabled applications can display the given file type, the OHS can launch a non-hypermedia aware application to display the file. Here the user has reached to a dead end with no available links. In general, new data model formats can be supported in an open hypermedia environment by enabling an (existing) application that can handle the required data model format.
4. Based on the information from the OHS, the hypermedia enabled application opens the requested file

5. The hypermedia enabled application sends a message to the open hypermedia system requesting anchors for the displayed file

6. The OHS replies with a list of anchors

7. The hypermedia enabled application displays the anchors and highlights the particular anchor that was connected by the traversed link

8. The user can now traverse available links from the newly displayed file (Wiil 1997; Grønbæk & Wiil, 1997)

6.3.1.2 Incorporating Audiovisual Links In Open Hypermedia Systems

Audiovisual links can be provided in a video document in many ways(Blackburn & DeRoue, 1998; Goose & Hall, 1995). The most common method, which is used by many web search engines (*www.altavista.com; www.google.com*), is to link (to or from) the complete video document. Once the user clicks on the link the video is streamed or downloaded from a particular site and is then played as a single blob object. The second method concerns the provision of browsing inside the video itself by either looking into its metadata or annotations. In this chapter, it is the second aspect, that we are concerned with, i.e. creating and accessing links inside the video.

tv-DbMS is a video database application that uses database techniques, i.e., structured access to video document. The techniques use indexes to assist video retrieval where the index defines and confines the way data can be accessed from the application. Providing links within the contents of a video is a novel way of watching it. Apart from saving resources of bandwidth and time by just providing the required portions to the user, hyperlinks provide a non-linear method of accessing the video.

6.3.2 tv-DbMS and Hypermedia

The aim of this integration is to support tv-DbMS users to use their annotations and metadata in hypermedia applications, and at the same time, to allow the user to author or browse hypermedia links within video documents. tv-DbMS has a complex database structure, which provides information to the user in an ordered form. This order depends on the type of the indexing or sorting techniques being used, and is always in a linear fashion. On the other hand, hypermedia systems offer a more natural method for exploring the information space using hypermedia links (Nielson, 1993). Links allow users to browse the information and access it according to their particular subject of

interest. However, studies have shown that it can be easy for users to become disoriented and that browsing on its own is not sufficient to provide access to large information systems (Nielson, 1995).

Supporting hyperlinks in tv-DbMS system can bring remarkable features as compared to other video browsing systems. Thematic indexing plays a very important role here. As discussed in Chapter 5, a video story can have many themes and these themes are inter-connected at many places and along with connecting them semantically (as discussed in section 5.4), a user can also create links between these themes, and can browse the video clips in a personalised way. Secondly, hypermedia links also provide the capability of accessing the World Wide Web and creating external links while browsing or querying video clips within the tv-DbMS domain.

The integration of hypermedia applications and tv-DbMS video documents requires not only considerations concerning hypermedia issues but problems concerned with continuous media also need addressing. This is discussed in the rest of this chapter.

6.3.2.1 Initial Integration Design

To incorporate hypermedia functionalities in tv-DbMS, we have to define the scopes of these two systems and consider their functionalities. A hypermedia system can support the existence of different kinds of links, which can be authored or computed. For an authored link, the link has to be manually created, whereas a computed link does not require any manual link creation. These links are stored separate into the main data archive, though integrated with the archives by providing the required information retrieval function to the hypermedia system. Users can reveal such links by specifying a subject of interest (e.g. keywords) and then choosing the related information retrieval function. In order to create hyperlink modules for tv-DbMS, we have to consider the above metaphors and have to enable the following functions to the hyperlink modules:

- A link authoring system
- A link parsing system

- A link storage system (i.e. the linkbase)
- Integration tools to enable the linking procedure in tv-DbMS

tv-DbMS, on the other hand, already has the features to annotate a video clip, query a video by using simple or complex queries and applying semantics to archive the story line features. The metadata regarding the video files is stored in a separate database (the data model about the metadata has been discussed in chapter 4). This database can be used as a linkbase (database of links), provided we keep the video-clip differentiation unit as video segment.

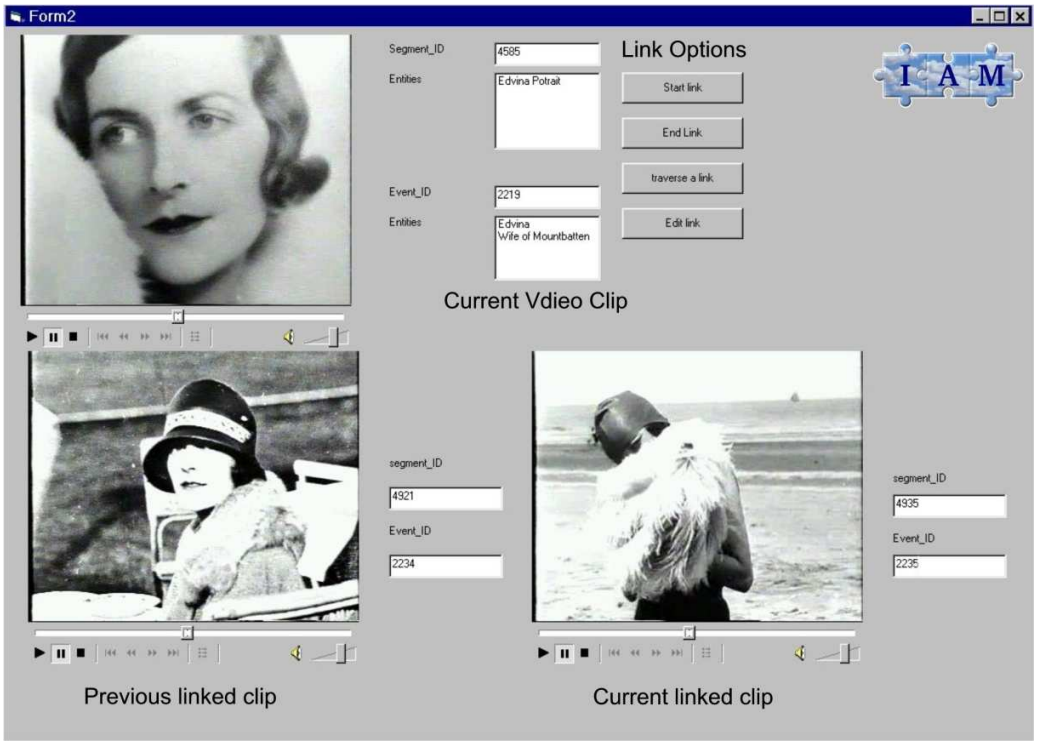


Figure: 6.10: tv-DbMS link module

For tv-DbMS, we adopted tailor-made viewers, as shown in figure 6.10, that can start a link, traverse a link and end a link. For authoring purposes, the viewer also has the

capabilities to create a link and can add a link to its link base. Another feature of this viewer is the access to WWW, i.e. the user can also follow a link to the World Wide Web as well. This is achieved by using the OLE (object link and enabling) techniques of Microsoft Internet explorer (*http://www.microsoft.com/IE*). This also provides a feature of creating data repositories of text and other media in the data archive and linking them to a particular video segment in the tv-DbMS video document. In figure 6.11, the tv-DbMS hypermedia module is linked with a video clip of Earl Mountbatten and a word document file.

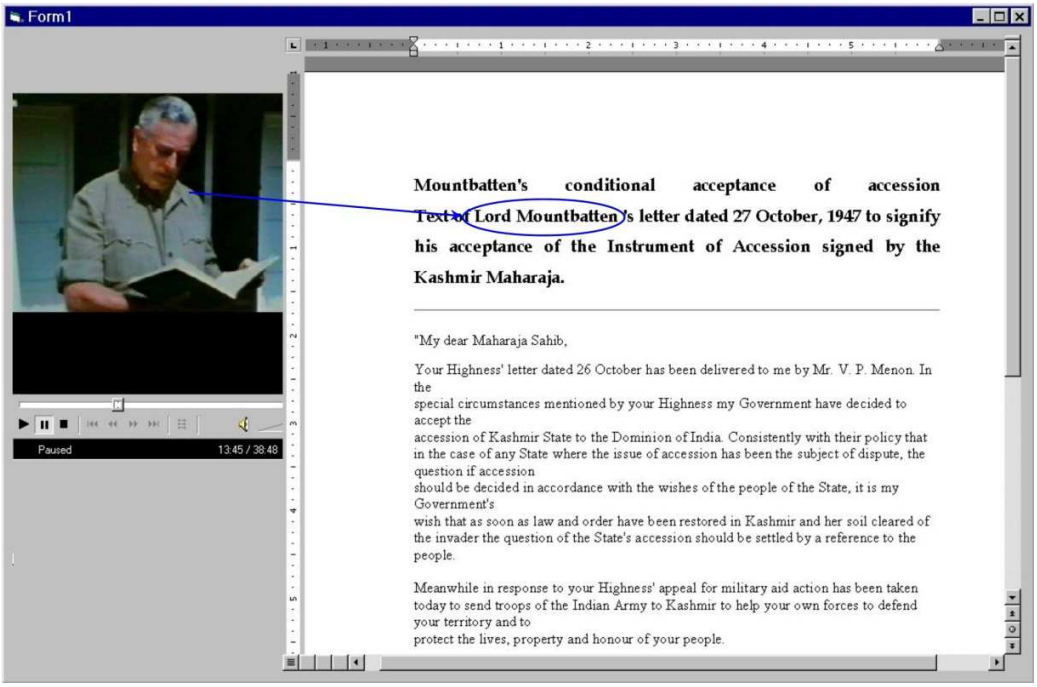


Figure 6.11: The Mountbatten video player linked to a Word document

6.3.2.2 Accessing Thematically Indexed Tables and Creating the Linkbase

In tv-DbMS data structure, the primary keyword is considered as a video segment, and then segments are further grouped into events and on top of events, a thematic index is

created by accessing video objects, locations and concepts in the video clips and providing semantic relations between them. Then these events are further interconnected with each other to create a hierarchy. This process results in a thematic index, which aids in quick and efficient query processing and provides a range of likely video clips. The metadata, which is, now stored in a thematic format can be linked at any level to any other or even to the same theme. This is done by running the add-link module, which is discussed in the link authoring tools section.

Another advantage of using the thematic indexing technique is to create personalised tracks[8] while following a video. This sort of linking is stored separately for a particular user and can be merged with the main annotations database, when the user logs into the system. This personalisation of links also supports the idea that different linkbases can be created for different people with different backgrounds. For example, a simple categorization could be dividing users in novice, experienced, experts and advanced users, and while loading the linkbases, the user can select that on what track he or she wants to browse the video. Further, in these categories, some access level can also be set. For example, a novice user will not be allowed to create new links or modify the existing links, whereas the experienced user can be allowed to create links. On the other hand, an expert can create links and can also modify the existing links, whereas the advance user's job is maintain the hypermedia module of the tv-DbMS database. This includes modifying the linkbase structure, creating new user levels and any change in the system functionality.

6.3.3 Link Authoring Tools

6.3.3.1 Creating / Modifying Links

In this module a user can view any video clip and can click on the viewer to create a link. This will pause the video and the system (a VB Script) will first find out from the

[8] Here a track is considered as the linkbase created by a user personally. This personal linkbase may be different for other users.

'segment' table which clip is running. This script will also find the relations of the segment tuple with the 'event' table and will find out all the related events. Further, these selected events are then searched for in the thematic index, and all the semantics (in our case concepts) and hierarchies are found out. In this way, a lookup table is created for the selected video clip. In the next step, another VB script will search the linkbase to find out whether this video clip is already linked or not. If this clip is linked the links will be shown to the user. Then the user is asked to provide the destination-anchor for that link. This destination-anchor can be either to the video itself, or some external object (text or media) in World Wide Web, or any other document (e.g. Word Document). Once the destination-anchor is determined, the link is stored in the linkbase. This destination-anchor is also stored in the lookup table created while selecting the video clip. This provides a very exciting approach in that all the thematically linked clips will also be sharing the same hyperlinks. For example while creating a link in the Mountbatten videos on the video clip of 'Edwina Mountbatten', will also create the same link on the video clips on 'Vicerene Edwina'. This can be a very useful option as semantically 'Edwina Mountbatten' and 'Vicerene Edwina' are different names for the same person, but textually the computer will take them as separate entities.

The above state approach was made possible by inserting a module for accessing multiple links. That is when a user clicks on a link, instead of accessing the linked object. This interconnecting module will provide the set of links, which are connected to this media object and then the user can select a link from the set to retrieve that link. By selecting the view linkbase table option, a user can view the linkbase database. Here the user can select a particular link and can add, modify or delete a particular link manually. This option should be handled carefully as its always risky to delete a link.

This is explained further by the following example: while viewing a text document, the user wishes to generate video links for the word 'clash' in the Earl Mountbatten video. Since the world 'clash' is not used in any annotations a standard retrieval system will retrieve nothing. On the other hand tv-DbMS linkbase will use thematic indexing thesaurus support to first retrieve the similar words for 'clash', which are 'war', 'scuffle',

'battle' and 'attack'. Then in the second step the tv-DbMS module will search for these words and will find the hierarchy of war as 'air bombing', 'soldiers fighting', 'soldiers marching', etc. Hence, by clicking on 'air bombing' the user will retrieve the video clips about Japanese Air fighters bombing in Burma.

6.3.3.2 Video Query Result Window

The tv-DbMS video query result window (discussed in chapter 4 and later in chapter 6) has the feature to play multiple video clips at the same time. In a single window it can show up to twelve video clips. The next step is to add the features of video browsing via hyperlinks in this window. To link other documents outside the domain of tv-DbMS (i.e. WWW, text documents, etc.), this window should also be web enabled with MS Office filters incorporated. This is done by connecting the video query result window with the link authoring panel discussed above. This means that the query result window can also tackle the outputs from the links outputs. i.e. after generating links, if the user wants to view multiple videos in one go, the user can use the video query result window. The hyperlink pad is also opened in the background, so that if the user is interested in viewing any external link, he or she can access the external links via the browser, which has Internet Explorer features built-in.

Another requirement is to provide full video segment context, so that the user can view the segment description or the event description along with viewing the video clip. This feature is already incorporated in the tv-DbMS query output window, by providing the segment descriptions on the left side of the window (figure 4.4, and figure 6.2).

6.3.4 Comparisons to Other Work

In relation to other hypermedia systems, tv-DbMS is unique. It is the only system tailored to handling video clips with a semantic approach, and also supports hyperlink techniques. When comparing tv-DbMS with other work in the field it is not the user interface, which is of interest, but the underlying data model; how the video contents are structured, what semantics they have and how they can be linked.

The demonstration in the sections above shows that tv-DbMS can describe an example of video content modelling and converting the thematic index database into a linkbase. This also supports the notion that how metadata which was used specifically to support video retrieval, can be re-used in the hypermedia world

In comparison to pure hypermedia models such as MAVIS (Tansley, 2000; Lewis et. al, 1996) and the Amsterdam model (Hardman et. al, 1993;1994), tv-DbMS is more concerned with the actual creation and manipulation of the metadata, instead of altering it for the hypermedia use, also how the two different themes can be linked together and overall how the thematic indexes can be personalized.

tv-DbMS also differs from the MAVIS data modelling approach, though both systems use semantic layers. A MAVIS link base is divided into four layers, which are raw, selection, selection expression and the conceptual. This means that the semantic data in MAVIS is stored at a different level from the media object data (annotations) and are stored permanently into these hierarchies, whereas in tv-DbMS, the semantics and the metadata are stored at the same level and are generated in real time, in a temporary lookup table, when the user requests to create a link.

6.4 Conclusion

The objective of this chapter was to show some results obtained by using tv-DbMS, the importance of metadata / annotations attached to video segments, usage of content based techniques and creation of a hybrid structure by combining these techniques. tv-DbMS is a system that has both the properties of content based and annotations based. At the content-based level, the system can perform automatic segmentation of the video clips by comparing the histograms and provide a unique identifier to each clip. At the annotation level, annotation is provided, and then at the higher level, the system creates a data structure, where these segments are divided into events or vice versa (as discussed in section 6.2), and then on top create a video object tree. Once this object tree is generated, a thematic index is generated by looking at the hierarchies of the video

segments and the video objects. This thematic indexing provides a novel way to perform complex queries on the video document. In Section 6.1.3.3, an attempt was also made to track the story line of a particular character by using simple English grammar techniques.

This chapter discussed the integration of tv-DbMS with open hypermedia systems. The integration was undertaken to demonstrate the role of tv-DbMS, a video database application, in the different domains of hypermedia environment and World Wide Web, with the motivation that database and hypermedia techniques can complement each other in supporting browsing access to video information.

The primary philosophy is picked up from Microcosm (Fountain et. al, 1990; Davis et. al, 1992; Hall, et. al, 1996), which is an open hypermedia system that supports the creation of personalising or prioritising links. This is made possible by separating link information from the original documents. This allows the system to provide hypermedia services to any third party application. The philosophy for tv-DbMS model is the same, metadata which was only generated to support the continuous video data, is also been used to browse and navigate the video clips. This also provides an opportunity to user to view the video into a non-linear way. However, this sort of viewing can be confusing but is rather very helpful in some cases. For example in an educational video, where a teacher links the current topic with some previous topic which he or she had discussed earlier, can be viewed non-linearly to freshen up the ideas, what s/he is going to discuss later.

The implication of implementing personalised links on a client server based video management system, where the user is storing different links in the link base of the client machine; with the main linkbase at the server side is hard to tackle. One of the solutions for this implication can be the application of tv-DbMS functionalities with Distributed Link Service (Carr, et, al, 1996) that can update the links periodically. The scope of DLS is beyond the scope of this thesis.

In this way, we have shown that how metadata can be used for performing queries, and hence the thematic indexing technique has the capability to enhance the effectiveness of a video retrieval system.

To prove the efficiency of tv-DbMS, it was further tested on educational videos. Chapter 7 deals with the experiments conducted on the video of Professor Hall's inaugural lecture.

Chapter 7

Evaluation II: Educational Video

To validate the design of tv-DbMS we also used it to create a video document of the inaugural lecture of Professor Hall on "Making Links". The idea to make a video document of this lecture was to explore the potential of the tv-DbMS model in the area of educational videos. The idea was that if tv-DbMS provides adequate results on this lecture, then a proposal could be made to create video documents of all the lectures and talks stored in the University's multimedia library. The benefits of tv-DbMS, such as providing metadata for the video clips, performing a query about a video clip, and then viewing only the desired part of a particular video, can provide many efficiencies, like saving user time and utilising less bandwidth in the network system, by streaming only the desired video clip, instead of the whole video document.

The inaugural lecture titled "Making Links" was given by Professor Hall in 1997, at the University of Southampton. This lecture was chosen on the basis that it had a very detailed discussion of hyperlinks and considerable use of modern techniques, e.g., AV projectors, video clips, and other resources were used to deliver this lecture. The video of this lecture is also used in the Intelligence Agents and Multimedia (IAM) Research Group as the introductory lecture for the new coming research students.

In this lecture, Prof. Hall discusses the status and the future of hyperlinks in the computing world. She also talks about her vision of a powerful distributed information agent known as 'Al', which is based on the use of metadata and hyperlinks and was inspired by super computer 'HAL' in the movie "2001 – a space odyssey" based on the book by Arthur C. Clark.

The video of this lecture is stored in MPEG-1 format and takes around 244 Mbytes of space. The format of the video is different to the Earl Mountbatten videos, which were in AVI format, to experiment that with tv-DbMS model using other video formats.

7.1 Providing Close Captions

Since this is an educational type of video, most of the information comes from the audio. Hence the first step was to get the audio of the video into closed captions. These closed captions are stored in the Microsoft SAMI file architecture. The Synchronized Accessible Media Interchange (SAMI) files (*www.microsoft.com/enable/sami*) were built to benefit disabled people who are either deaf or hard of hearing, but can read text information. SAMI is based on HTML to provide a familiar, user readable format. The full implementation of SAMI is designed to support the various closed captions specifications in a single authoring format.

Closed captioning implies that the captions are synchronized with the audio content. It also includes the description of sounds (e.g., "... the dog barks...") as well as the use of symbols (icons) to represent the type of content (e.g., a musical note could be used for music). For tv-DbMS, these closed captions have been provided manually, but a speech–text converter can be used to capture the audio data. However the application of speech –text converter, its advantages and disadvantages are beyond the scope of this thesis.

```
<SAMI>
        <HEAD>
                <Title>WENDY'S INAUGRAL LECTURE</Title>
                <SAMIParam><!--
                        Copyright="(c) Copyright 2000, IAM labs"
                        Media="wendy.mpg", none
                        Length=75361
                        CaptionMetrics=scalable
                        CaptionLineLength=2000
```

Figure 7.1: SAMI file for Prof. Hall's inaugural lecture

Figure 7.1 shows the SAMI file of the inaugural lecture, which starts with the tag <SAMI>. Then in the <HEAD> section we define the type and content of video data. In the <STYLE TYPE> section, the output format of the closed captions is defined and finally in the <BODY> section, the caption itself and the time to start in milli-seconds is defined. The media player enabling SAMI features is shown in figure 7.2. Once a SAMI file is attached to the media player, its captions property should be switched on. This can be seen by a 'CC' button on the media player's property toolbar. Closed captions can be turned off by clicking the 'CC' button in the toolbar.

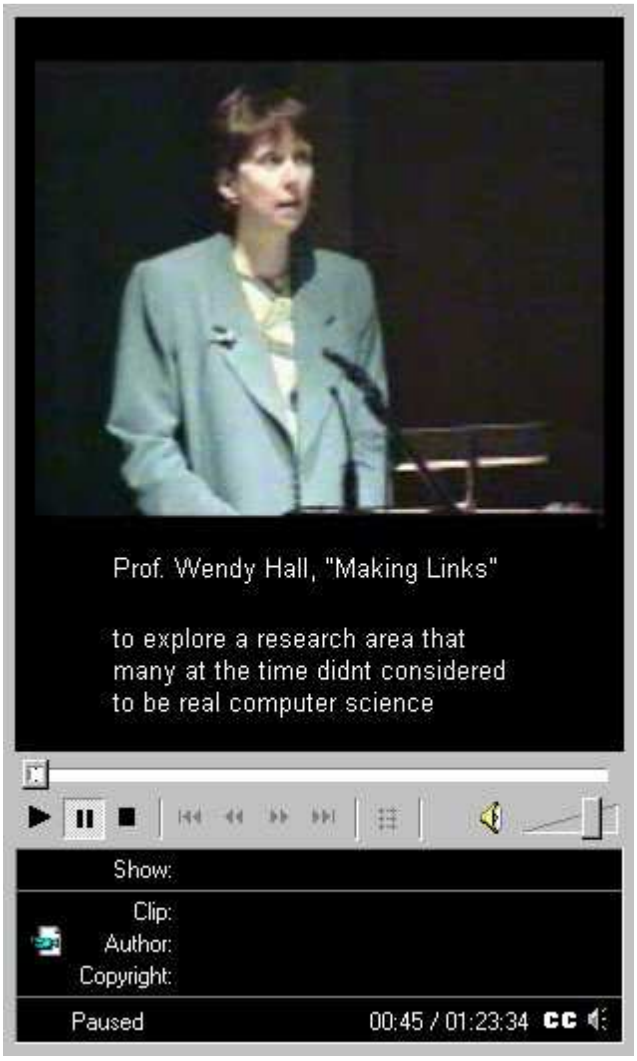


Figure 7.2: Media Player enabling the display of closed captions

7.2 Performing Simple Queries

Figure 7.3 shows the video clips retrieved, when a query about 'Bush[9]' was performed in the tv-DbMS. Here it should be noted that since this is a video of an educational lecture, the metadata generated is heavily dependent on audio content and concepts provided by the speaker. By looking at figure 7.3, one can see that the word 'Bush' was used more as a concept, and the video clips, where the speaker is talking about 'Bush' are retrieved.

[9] Vannevar Bush (1890 – 1974) Author of "As we may think" (1945)

Table 7.1 and Table 7.2 show the segment and event entries for the first video clip retrieved in figure 7.3.

Figure 7.3: Simple query search for 'Bush' in the Inaugural lecture

7.3 Queries Based on Closed Captions

These queries are similar to simple queries, but here instead of using metadata only, we also use closed captions to retrieve the data. This provides a bigger domain to search the query. In the tv-DbMS database model, these closed captions are stored as memo type data in the segment table.

In the educational videos, like this inaugural lecture, closed captions play a very important role. Since the visual movement is very slow, as the lecturer is standing still

on the podium and presenting her ideas or the talking head is speaking, so the techniques, such as visual segmentation or scene change detection, are not very useful, and usually a video segment contains information on various topics, that should be conceptually divided into mini-segments. Here we use the event entity to divide these visually detected segments into mini-segments. As discussed in section 4.6, an event can be a subset of itself, so this property of event entity is used here to decompose these segments into many smaller segments.

Figure 7.4 shows the result window of a query made about 'Al' in the lecture video. This provides some video clips about 'Al' but also lots of other related scenes that are not required here. For example, the video clips, where Prof. Hall is acknowledging her colleagues and friends. These video clips are not related to agent 'Al'.

On the other hand, if we use the thematic indexing for retrieving 'Al' as explained in the next section, there is less chances of the system doing errors. Table 7.3 and 7.4 shows the segment and event table for the last video clip retrieved in figure 7.4.

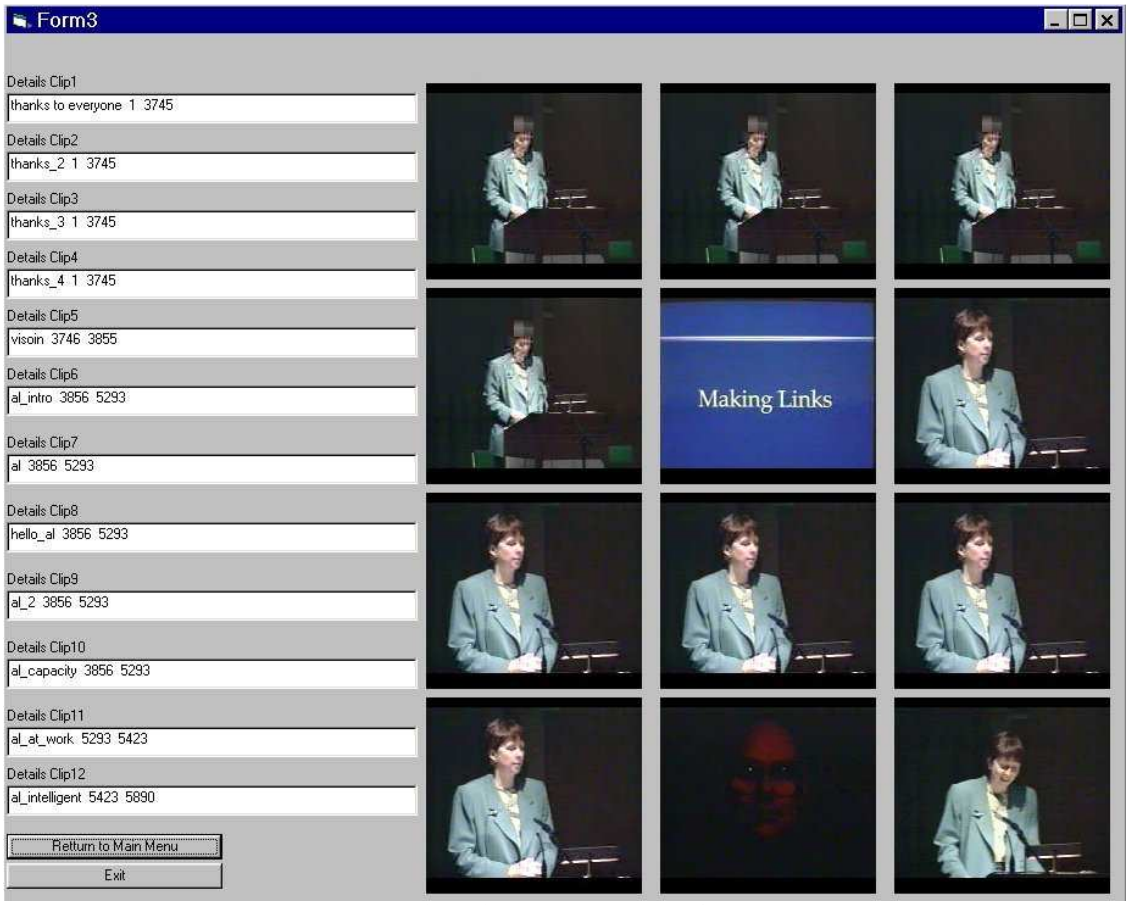


Figure 7.4: Closed Caption search for agent 'Al'

SEGMENT TABLE	
SEGMENT_ID	43
START_FRAME	7566
END_FRAME	8050
PERSON	Al
LOCATION	--
OBJECT	--
CONCEPT	Hartlib slide
MOTION_TYPE	Simple
CO_ORD_SYSTEM	XY
SPATIAL_UNITS	NA
CAMERA_FOLLOWS	NO

CO_ORD_ARE_3D	NO
KEY_POINTS	--
2D_3D_SPATIO TEMPORAL	2D
INTERVAL_3D	None
INTERVAL_2D	None
INTENSITY_OF_ACTIVITY	Slow
DIRECTION_OF_ACTIVITY	Straight
TEMPORAL_DISTRIB_OF_AC	None
SPAITAL_DISTRIB_OF_ACTIVITY	None

Table 7.1: Segment tuple for first result shown in figure 7.3

EVENT TABLE	
EVENT_ID	51
EVENT_DESCRIPTION	Bush_1
EVENT_CONCEPT	--
EVENT_TIME	--
EVENT_DATE	--
EVENT_ADDED_INFO	These are the people who have been the inspiration of the work I have taken The first of these was bush

Table 7.2: Event tuple for first result shown in figure 7.3

SEGMENT TABLE	
SEGMENT_ID	5
START_FRAME	5423
END_FRAME	5890
PERSON	Wendy
LOCATION	
OBJECT	
CONCEPT	AI
MOTION_TYPE	Simple

CO_ORD_SYSTEM	XY
SPATIAL_UNITS	NA
CAMERA_FOLLOWS	NO
CO_ORD_ARE_3D	NO
KEY_POINTS	
2D_3D_SPATIO TEMPORAL	2D
INTERVAL_3D	None
INTERVAL_2D	
INTENSITY_OF_ACTIVITY	Slow
DIRECTION_OF_ACTIVITY	Straight
TEMPORAL_DISTRIB_OF_AC	None
SPAITAL_DISTRIB_OF_ACTIVITY	None

Table 7.3: Segment tuple for first result shown in figure 7.4

EVENT TABLE	
EVENT_ID	47
EVENT_DESCRIPTION	Al Intelligent
EVENT_CONCEPT	Al
EVENT_TIME	--
EVENT_DATE	--
EVENT_ADDED_INFO	lets hope that Al is going to be a little better behaved than that tonight

Table 7.4: Event tuple for first result shown in figure 7.4

7.4 Thematic Indexing Search

A thematic indexing tree for this lecture is shown in figure 7.5. Here two themes are evident. The first is about Professor Hall herself and the second is about 'Al'. In the theme of Prof Hall, (as information available from the video document only!) she is a professor at Southampton and has designed Microcosm. Microcosm is a system that uses links. On the other hand 'Al' is a system, which also uses links, which have tags. Vannevar Bush and Samuel Hartlib have also worked on associative links. The tree shown in figure 7.5 is showing only the two main themes, whereas the tree shown in

figure 7.7 is a complete thematic indexing tree for Professor Hall's lecture. For the queries shown below, we will be using two themes, i.e. Professor Hall and 'Al', so a shorter version of the tree about these two themes is shown in figure 7.5.

The next step is to provide metadata about the names, objects, locations and concepts for the video clips, using Event Annotator module (discussed in chapter 6) and develop the CVOT. The procedures to develop CVOT discussed in chapter 6 are applied here, for the metadata of Professor Hall's lecture. After this procedure, we can perform thematic indexing search on the whole video document. Figure 7.6 shows the out put window for the query 'Al' done through thematic indexing. As it is evident from figure 7.6, that the majority of the video clips does not have 'Al' present in them, but are some how related to 'Al'. For example the first and second clips retrieved in figure 7.6, are about the introduction of the agent 'Al', and these clips are retrieved, as there is a relation 'Professor Hall introduce Al' between them. This relationship was created while developing the CVOT. For example, in the first clip retrieved in figure 7.6, Professor Hall is describing about 'Al'. Here the 'Al' is stored as a concept for this clip and while developing CVOT and was 'strongly ordered' with the other video clips discussing about 'Al'.

Once the video object model is developed, the next step is to create the thematic indexing tree. The complete thematic indexing tree for this video is shown is figure 7.7. This tree now depicts all the relations and concepts between the entities used in the video. For example, from figure 7.7 it is clear that, Professor Hall is discussing hypertext links, and the history of hypertext from the point of view of key figures in the development of the subject such as Hartlib, Van Dam, Brown and Engelbart (Hall, et. al, 1997). She also talks about the hypermedia systems developed for IBM-PCs, the Apple Macintosh (HyperCard) and Microcosm. How it was (in this particular case) used by medical students at Southampton University to understand the subject of cell motility. Additionally, Microcosm was also used to develop the digital archives of Earl Mountbatten and Sir Winston Churchill[10].

[10] Former Prime Minister of United Kingdom

7.4.1 Aiding Closed Captions to Generate a Common Video Object Model

Closed captions provide the textual form of the audio data. As discussed earlier this audio data has more impact in educational videos and on the other hand metadata and annotations can be easily extracted from closed captions. Since thematic indexing is heavily dependent on annotations, closed captions are very helpful in generating thematic index data. For example, the introductory part of the lecture provides information about the Multimedia Research Group and the development of Microcosm. Closed caption data for a segment can be easily copy/pasted in the video object tree module of tv-DbMS. However, some additional information is still provided by the annotator such as the fact that Professor Hall is based in Southampton (in figure 7.5), but most of the metadata for video information is easily extracted from the closed captions.

Another advantage of the thematic model is the removal of the time line from the digital video, so that the user can query the video objects or clips, without the constrains of temporal features of video. For example, the tree shown in figure 7.7 has no time features related to any video object, i.e. the notion that Prof. Hall first talked about the University of Southampton, then discussed the historical background of hyperlinks and then Microcosm is ignored, so that the user can observe all the entities at the same time. Once the user selects a particular entity for query, then that entity is first matched in the video segments, and then all the matched video segments are retrieved as the output for the query.

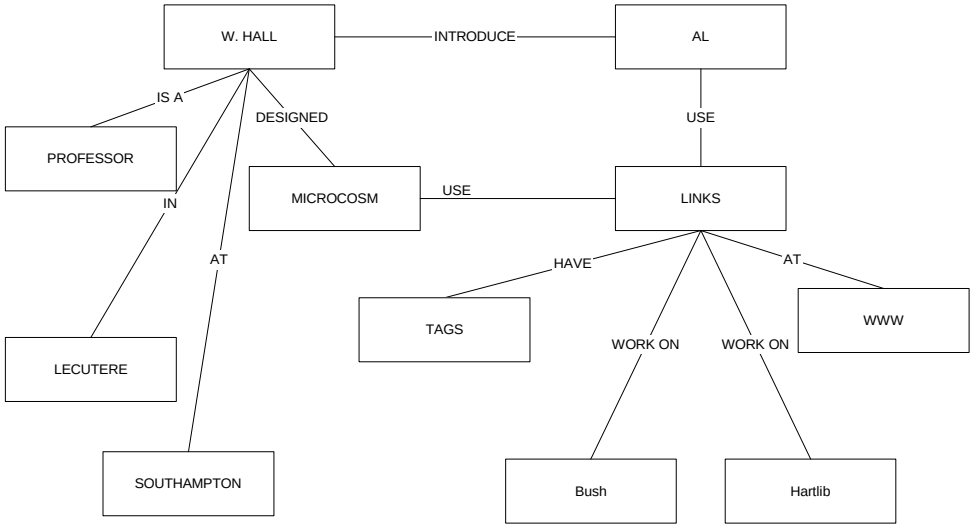


Figure 7.5: Thematic indexing tree introducing 'Al' in Prof. Hall's lecture

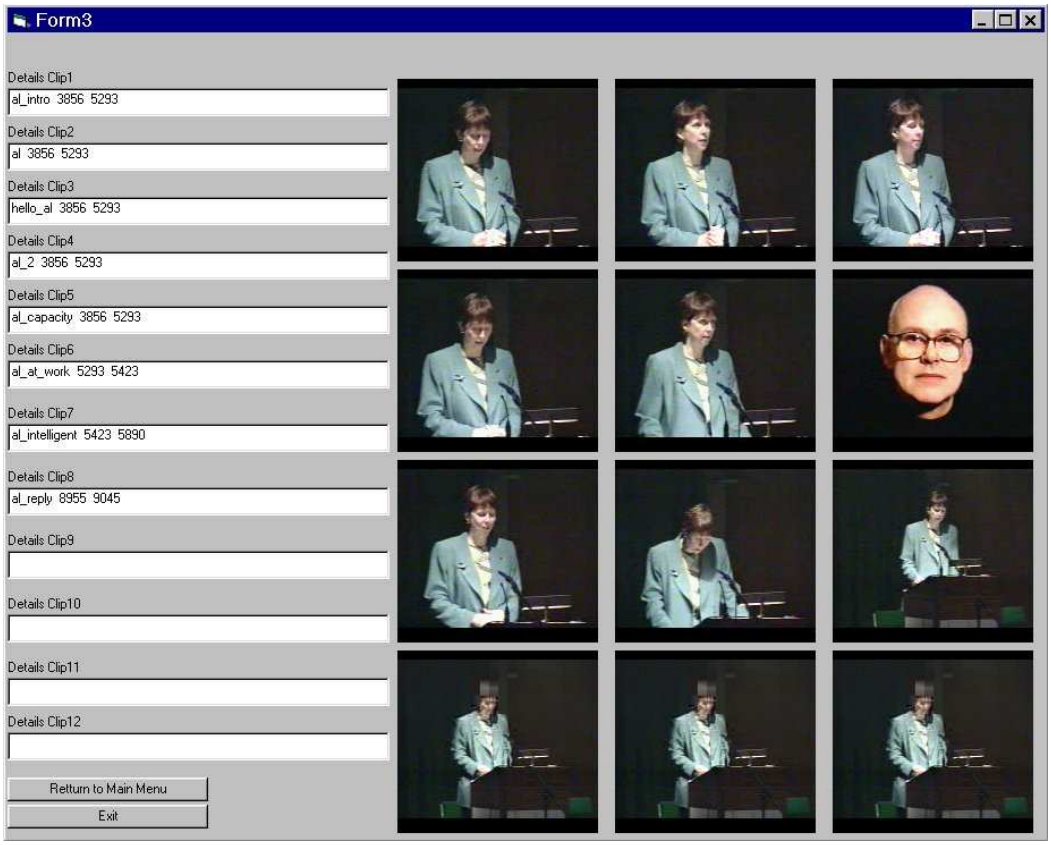


Figure 7.6: Video clips retrieved through thematic indexing search for the query of 'Al'

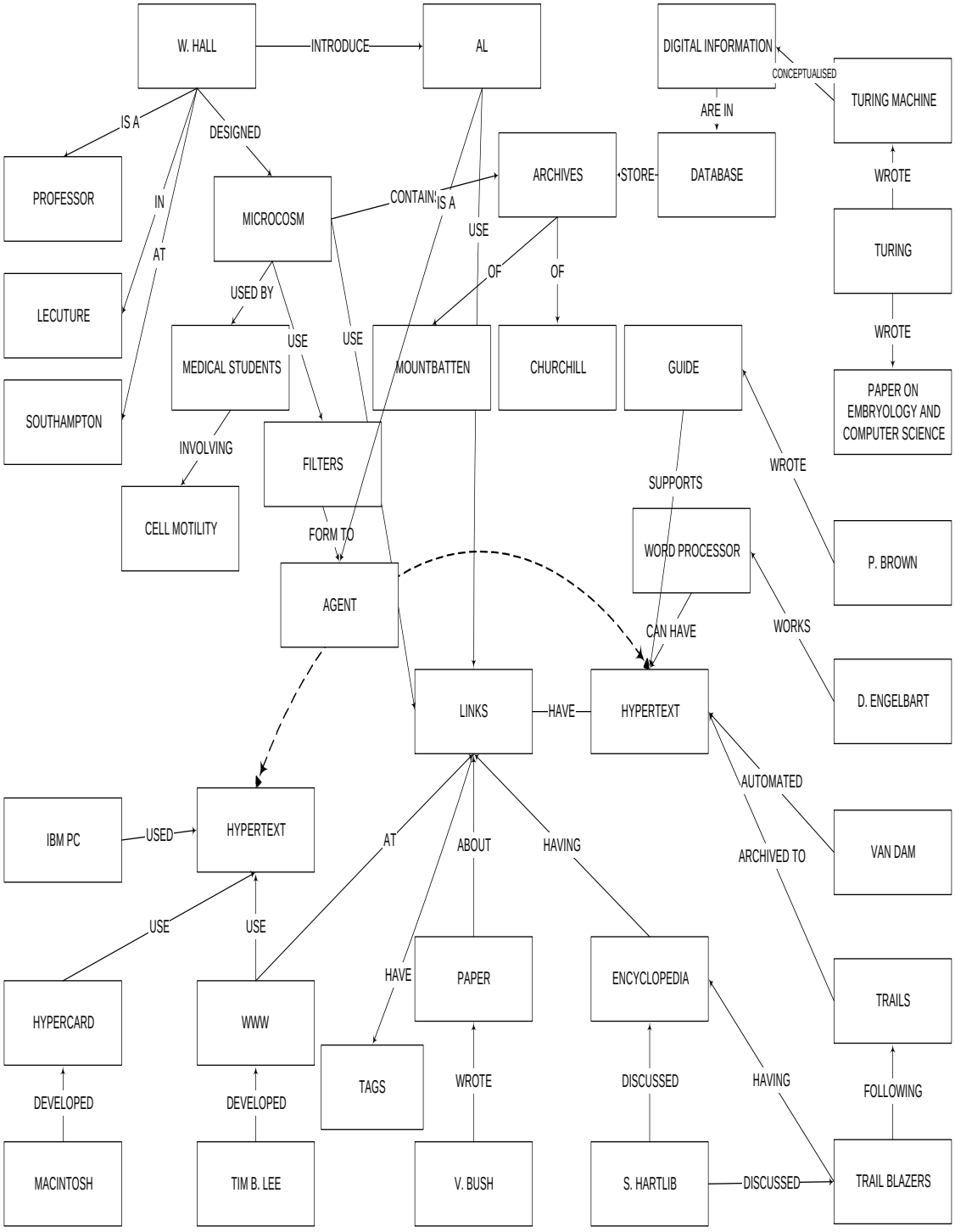


Figure 7.7: Thematic indexing tree for Prof. Hall's lecture

7.4.2 Evaluating Results

The development of tv-DbMS applications for an educational video is to prove the capability that this model can work for any sort of videos, whether documentaries, educational or entertainment. It has also been tried to show that videos, which have less visual movements (such as educational), heavily rely on metadata or annotations for intelligent querying. An attempt has been made to apply the thematic index model on Professor Hall's inaugural lecture. For example, a simple keyword (or normal closed caption) search about 'Al' (shown in figure 7.4) retrieved video clips, though most of them were only partially related to 'Al' but not describing 'Al' particularly. On the contrary, a thematic search in figure 7.6 retrieved only those video clips that were completely related to 'Al'. In this way it has been showed that thematic indexing works much better then simple keyword search or searching only on closed captions.

While working on Professor Hall's inaugural lecture, we have used closed captions to compile metadata. This was done because of the importance of the audio data in educational videos. One can argue that this should also be done for the documentaries of Earl Mountbatten (discussed in chapter 6). Closed caption module for educational videos can be justified by the fact that it is almost impossible to generate accurate or precise video segments depending on visual aspects only. The Earl Mountbatten videos are much more visual in nature and so we are less dependant on the audio track for thematic indexing.

As suggest in the beginning of the chapter, that it was intended to stream the videos of all the lectures held at University of Southampton. To initiate this gigantic task, it was felt that it would be better if we follow the current metadata standards to generate the metadata through tv-DbMS annotation module. The following section discusses about the integration of tv-DbMS with current metadata standards.

7.5 Incorporating tv-DbMS with metadata standards

This section deals with the implementation of tv-DbMS with existing metadata standards, discussed in section 3.6. For tv-DbMS documents, it was felt that a hybrid of all these standards could be created to standardise the metadata. Since Dublin Core (DC) has a very strict policy about defining elements in a document and its qualified elements were designed only for textual documents, it would be unwise to choose Dublin Core only. RDF, on the other hand, does not provide any description about continuous data, but it handles hierarchies and semantics very efficiently. On the other side, MPEG-7 is ideal for low-level and high-level video information, but does not provide any tool for publishing video on the net. So, the fusion of these standards, where the video details described using MPEG-7 schema with a wrapper of RDF metadata, could be an optimal solution for publishing video documents on the net.

The following example shows tv-DbMS metadata being transferred into RDF syntax. The database entities discussed in Chapter 4, are converted into RDF. An advantage of this conversion is the development of a model for defining relationships between resources. The hierarchical video structure is supported by defining RDF sequence collection nodes with each DC:Relation:HasPart and a separate RDF:Description for each element of the sequence collection. The notches between these standards contribute to the readability and ease of understanding of video structure.
The RDF Schema is used here to define the hierarchical structure of the video documents and the attributes associated with each level. It is used to generate the form fields for inputting and editing the metadata for both the complete video document and each of the scenes. It is also used to constrain and validate the input and to define the RDF output file format. For tv-DbMS the fundamental hierarchy is developed between video documents, segments and events. Figure 7.8 illustrates the RDF data model for the video documents in this application. The complete RDF schema describing the event and segment entities of tv-DbMS data model, is shown in Appendix A.2

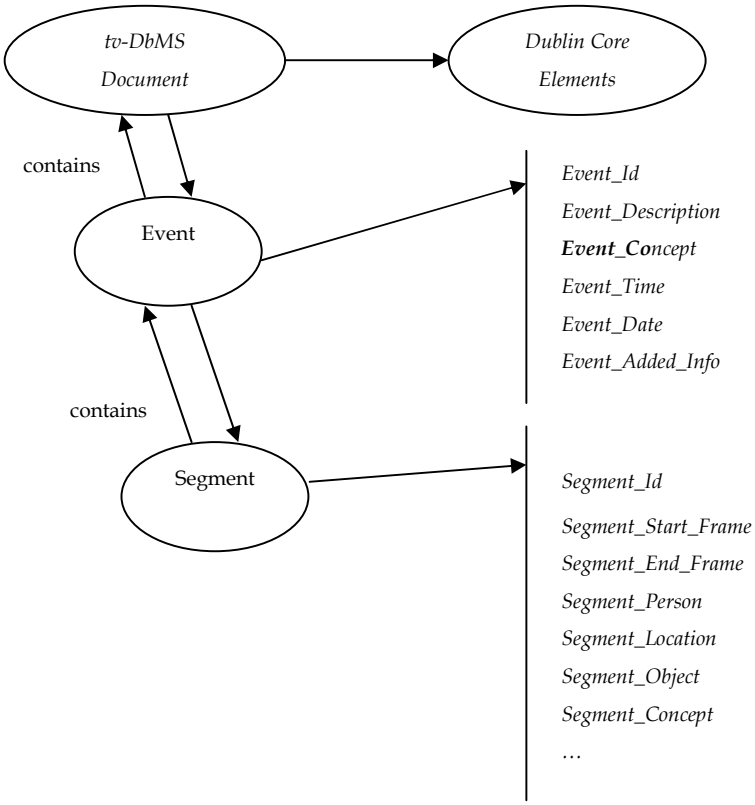


Figure 7.8: RDF model to tv-DbMS video document

In the schema, the fifteen Dublin Core properties are associated with the top-level Video_document class. In addition, there is a 'contains' property, whose domain is the Video_document class and whose range are the segment and event classes. These classes are defined as sub-class of Video_document so that they inherit all of the Dublin Core properties. In addition these sub-class have their own additional descriptive properties like: duration, startframe, endframe, person annotation, location annotation, etc.

One of the major problems with continuous media much as video is that there is currently no standard way of pointing to a section of the video, using a URL. Hunter (2000) suggests that the qualifying information that needs to be able to be specified in a URL referring to continuous media, includes:

- A specific time offset into the video / audio
- A specific time range within the video / audio
- A specific label within the video / audio where the label is resolved to a position and duration within the video / audio by some other service

While streaming data through tv-DbMS, we can define a link to a section of a video source by defining an anchor element with specific begin and end attributes. For example, in SMIL, a video tag can be defined as:

```
<video src= “ringading.ecs.soton.ac.uk/data/incoming/
                sak97r/wendy.mpg”>
<anchor id= “segment20” begin= “00:35:45.01” end= “00:36:32.21”
</video>
```

Using this approach, a particular video clip can be reached at:

```
“http://ringading.ecs.soton.ac.uk/sak97r/

 wendy.mpg#segment20”
<?xml:namespace href="http://www.w3c.org/RDF/" as="RDF"?>
<?xml:namespace href="http://purl.org/RDF/DC/" as="DC"?>
```

Below are the series of RDF-encoded metadata descriptions for the different levels of a tv-DbMS video document using Prof. Hall’s inaugural video document as an example. Each RDF file points to the corresponding actual content via the RDF:HREF value.

```
<RDF:RDF>
  <RDF:Description About="http://ringading.ecs.soton.ac.uk/sak97r/wendy.mpg">
    <DC:Title>Prof. Hall's Inaugural Lecture</DC:Title>
    <DC:Description>Inaugural Lecture on Making Links</DC:Description>
    <DC:Subject>Educational, lecture</DC:Subject>
    <DC:Publisher>MMRG, Univeristy of Southampton</DC:Publisher>
    <DC:Date>1997-06-04</DC:Date>
    <DC:Format DC:Scheme="IMT">video/mpg</DC:Format>
    <DC:Language>en</DC:Language>
    <DC:Format.Length>60 mins</DC:Format.Length>
    <DC:Relation.HasPart>
      <RDF:Seq>
        <RDF:LI Resource="http://ringading.ecs.soton.ac.uk/sak97r/wendy.mpg#clip1"/>
        <RDF:LI Resource="http://ringading.ecs.soton.ac.uk/sak97r/wendy.mpg#clip2"/>
        <RDF:LI Resource="http://ringading.ecs.soton.ac.uk/sak97r/wendy.mpg#clip3"/>
        <RDF:LI Resource="http://ringading.ecs.soton.ac.uk/sak97r/wendy.mpg#clip4"/>
        <RDF:LI Resource="http://ringading.ecs.soton.ac.uk/sak97r/wendy.mpg#clip5"/>
        ..
        ..
      </RDF:Seq>
    </DC:Relation.HasPart>
  </RDF:Description>
</RDF:RDF>
```

Figure 7.9: RDF metadata for the video document of Professor Hall’s inaugural lecture

The RDF metadata for the URL

"http://ringading.ecs.soton.ac.uk/sak97r/wendy.mpg#segment1 is given in figure 7.10

```
<?xml:namespace href="http://www.w3c.org/RDF/" as="RDF"?>
<?xml:namespace href="http://purl.org/RDF/DC/" as="DC"?>
<RDF:RDF>
  <RDF:Description About="http://ringading.ecs.soton.ac.uk/sak97r/wendy.mpg#event1">
    <DC:Type>Image.Moving.Educational</DC:Type>
    <DC:Description.text>"Prof. Wendy Hall, "Making Links"</DC:Description.text>
    <DC:Format.Length>148 secs</DC:Format.Length>
    <DC:Coverage.t.min DC:Scheme="SMPTE">0:0:0;1</DC:Coverage.t.min>
    <DC:Coverage.t.max DC:Scheme="SMPTE">0:2:28;0</DC:Coverage.t.max>
    <DC:Relation.HasPart>
      <RDF:>
        <RDF:LI Resource="http://ringading.ecs.soton.ac.uk/sak97r/wendy.mpg#segment1.1"/>
        <RDF:LI Resource="http://ringading.ecs.soton.ac.uk/sak97r/wendy.mpg#segment1.2"/>
        <RDF:LI Resource="http://ringading.ecs.soton.ac.uk/sak97r/wendy.mpg#segment1.1"/>
      </RDF:Event>
    </DC:Relation.HasPart>
  </RDF:Description>
</RDF:RDF>
```

Figure 7.10: RDF metadata for a video event

The metadata abstraction shown in figure 7.9 and figure 7.10 has been incorporated to stream video clips for Professor Hall's inaugural video. tv-DbMS supports access to provide links to these video clips. This is done by providing metadata shown above in figure 7.9 and 7.10. Hierarchies have also been kept intact by dividing the segments into further sub-segments. These sub-segments are basically the events created in a long video segment. In this way we can keep the data model of tv-DbMS (discussed in chapter 4) intact, while providing the standard metadata. Dublin Core and RDF provide metadata standards for video documents only (i.e. video title, video clips) and do not play any role in defining the video objects. This means that thematic indexing does not play any role while standardising metadata according to incorporate the standards. However, MPEG-7 standard works at the video object level, and the integration of thematic indexing to MPEG-7 is discussed in Appendix A.1.

7.5.1 Advantages of the Hybrid Approach

Dublin Core provides both 'core' and 'full' data descriptions to satisfy a range of user groups' needs. It also enables searching across different media types and can exploit all of the work already done on Dublin Core metadata generation and Dublin Core based indexing and search tools. It inherits the advantages associated with Dublin Core -

simplicity, semantic interoperability, scalability, international consensus and flexibility. (Though it could justifiably be argued that the proposed extensions for video destroy the simplicity.)

On the other hand, RDF syntax allows labeled directed graphs to be built which support the hierarchical structure of video. Since it is encoded in XML (eXtensible Markup Language) which is based on SGML and is better able to support multimedia than HTML, this has an edge over other standards and also provides the property to merge with Dublin Core easily. RDF can also leverage off other tools and code bases being built around XML e.g. SMIL and can be used as a container for different communities' metadata schemes.

Dublin Core was designed to do high-level interdisciplinary searching for complete textual documents across heterogeneous databases and schemas. It provides a simplified set of 15 elements which enables searching across the WWW. Dublin Core was not designed to provide metadata at a low level such as scenes and shots.

Consequently there are a number of disadvantages associated with using Dublin Core for describing complex video documents. These include loss of simplicity, need for a great number of sub elements (especially within the Description element), schemes and rules. There is no way to specify fine-grained synchronisation between the different components.

Dublin Core can be used to describe audiovisual documents as a whole and to enable searching for complete audiovisual documents i.e. search and query at a high level on the 15 core elements. For example: "Find all BBC News Clips" or “Find News Clips about American President Elections”. This would perform a text search on the 15 core elements for the string "American President Elections".

MPEG7 can be used to provide a detailed hierarchical description of the content. The MPEG7 data can be used to enable low level content-based querying such as: "Give me

close-up shots of the Prime Minister while meeting with the Queen". Since large components of the Dublin Core work do satisfy the MPEG7 requirements, it makes sense to exploit these aspects in MPEG-7. The exercise above has shown that Dublin Core, with extensions (particularly domain-specific qualifiers in the Description field), could form a basis for MPEG-7.

The advantages of the hybrid approach include incorporation of existing Dublin Core text-based search engines which can perform queries of continuous media types. It also satisfies the original intention of Dublin Core to provide a core description and not to replace specialised cataloguing methods. Apart from, MPEG-7 can be developed independently to provide low level fine-grained content-based querying. Figure 7.11 shows a core structure of such a hybrid example. A complete schema is given is Appendix A.2.

```
<?xml:namespace href="http://www.w3c.org/RDF/" as="RDF"?>
<?xml:namespace href="http://purl.org/RDF/DC/" as="DC"?>
<?xml:namespace href="http://www.mpeg.org/mpeg7" as="MPEG7"?>

                                                    <RDF:RDF>
 <RDF:Description About = "http://www.ringading.ecs.soton.ac.uk/
                    sak97r/wendy.mpg">
    <DC:Title>Prof. Hall's Inaugural Speech </DC:Title>
    <DC:Creator>S </DC:Creator>
    <DC:Subject>Educational, Lecture Series</DC:Subject>
        <DC:Publisher>Multimedia Labs, Uni. Of
                Southampton</DC:Publisher>
    <DC:Contributor.Presenter>Prof. Hall
    </DC:Contributor.Presenter>
    <DC:Format DC:Scheme="IMT">video/mpg</DC:Format>
    <DC:Language>en</DC:Language>
    <DC:Date>06/04/97</DC:Date>
    <DC:Format.Length>60 mins</DC:Format.Length>
    <MPEG7:Duration>10800</MPEG7:Duration>
    <MPEG7:Script>http://rignading.ecs.soton.ac.uk/
                        sak97r/wendy_speech.txt </MPEG7:Script>
    <MPEG7:Locale>S. Khoja</MPEG7:Locale>
 </RDF:Description>
</RDF:RDF>
```

Figure 7.11: A hybrid approach

7.6 Conclusion

The objective of chapter 6 and 7 was to show some results obtained by using tv-DbMS, the importance of metadata / annotations attached to video segments, use of content based techniques and creating a hybrid structure by combining these techniques. tv-DbMS is a system that has both the properties of content based and annotations based techniques. At the content-based level, the system can do automatic segmentation of the video clips by comparing the histograms and providing a unique identifier for each clip (discussed in chapter 4). At the annotation level, this system first stores annotation for each video segment, then at the higher level creates a data structure, where these segments are divided into events or vice versa (as discussed in section 6.2), and then creates a video object tree. Once this object tree is generated, then a thematic index is generated from hierarchies of the video segments and video objects. This thematic indexing provides a novel way to perform complex queries on a video document. In Section 7.3, closed captions were introduced to the tv-DbMS system, and how closed captions can be used for widening the domain information. In this way, we have shown how metadata can be used for performing queries, and to enhance the effectiveness of a video retrieval system.

The final section of chapter 7 discusses a hybrid approach to providing a protocol for publishing metadata on the internet. This approach combines the features of RDF, Dublin Core and MPEG-7, and provides an innovative method for storing metadata for digital videos, while incorporating the tv-DbMS data model.

The next chapter deals with conclusions and the future work regarding tv-DbMS.

Chapter 8

Conclusions and Future Work

8.1 Conclusions

Evaluation of a video based composition is a complex process, as various features associated with a video composition need to be analysed. Techniques such as efficient data model designing, video data integration, annotation authoring, thematic indexing and hyperlink integration evaluate the retrieval performance of a video database management system, and give some idea of the importance of the richness of video data information. These techniques also provide the user with a variety of facilities to retrieve information from a continuous medium. This continuous medium can be stored in digital format (e.g. mpeg, avi, quicktime, asx, etc.), or streamed through the inter or intra nets. Projects based on video information systems especially related to video indexing and retrieval techniques have received much attention in recent years. Existing video indexing and retrieval techniques for video systems are improving quickly, some seem very promising, and more video systems are likely to be available in the coming years.

Existing video systems can be classified in terms of their applications into four types: systems that provide information resources, systems that provide data analysis tools for video-based research, systems for video editing and authoring purposes and systems for video production and broadcast. Though these systems support different applications, the basic activity that a user performs using any video system is video retrieval. Different users will approach the task of retrieving video information from different perspectives and for different purposes. This is due to the reason that video itself is such a rich information type that apart from temporal and spatial information, video supports semantics (i.e. conceptual information) in different combinations to form different stories.

In this dissertation, we have introduced a complete video database system with a concept of thematic indexing that supports semantic abstraction in digital video. Thematic indexing techniques work on a common video object model that categorizes all the video objects in a common video-object tree. This object model also creates relationships between video objects from different video clips, and generates a hierarchy between these objects.

As well as the video objects present in the video clip, tv-DbMS also supports semantic relationships between them. To support there relationship, video segments generated through frame-to-frame histogram matching are sub grouped into different events, narrating a story or a concept. For example, a video of different people sitting in a room, could be classified as an event labelled 'meeting', instead of storing them in smaller segments, which are generated due to camera movement or video object transition. All these segments will be grouped in an event to form a small part of a story. An event can be further grouped into another event, or can be a part of another event. For example, in educational videos the camera and the lecturer are both usually stationary for a longer period of time to form a single segment, but the speaker can change the topic of his or her speech during that period. In such cases, the event entity can be a sub part of a segment.

The event further forms the basis of thematic indexing. Thematic indexing is a novel approach to index video data. The contents of the video are broken into one or more themes, semantically related to each other. The philosophy behind thematic indexing is to support complex queries. The whole video is indexed by words (or keywords) which form a type of wrapper around the video. These keywords are interconnected with each other through semantics. The time-domain limitation is extracted out, so that a user can see a tree of information flow of the video. Once this wrapper is generated, the user can hop around in this 'web' of information and can query the video more efficiently and quickly then in traditional video information management system.

To further support, a thesaurus is attached to the thematic query module, so that the user can also look for similar (synonymous) words in the database, and then retrieve the desired video clips. This technique provides another dimension to the common object video tree of synonym words, providing an additional feature for the user.

Audiovisual information plays an important role in our society, whether it is recorded in such media as film or magnetic tape or originating, in real time, from some audio or visual sensors and whether it is analogue or (increasingly) digital. While audio and visual information used to be interacted with directly by the human beings only, there is an increasing number of cases where the audiovisual information is created, exchanged, retrieved, and re-used by computational systems. This may be the case for such scenarios as image understanding (surveillance, intelligent vision, smart cameras, etc.) and media conversion (speech to text, picture to text, etc.). Other scenarios are information retrieval (quickly and efficiently searching for various types of multimedia documents of interest of the user) and filtering in a stream of audiovisual content description (to receive only those multimedia data items which satisfy the user's preferences). Audiovisual sources play a very important role in our lives, and there will be a growing need to have these sources processed further. This makes it necessary to develop forms of audiovisual information that go beyond the simple waveform or sample-based, compression-based or even object-based representations. Forms of representations that allow some degree of interpretation of the meaning of the

information are necessary. tv-DbMS is an application for describing the video content data that will support these operational requirements. The requirements apply, in principle, to both time-domain and spatial-domain aspects of video data. tv-DbMS aims at providing standardised core formats (avi, mpeg, quicktime, etc.) allowing description of audiovisual data content in multimedia environments.

This thesis also contributes an investigation of the feasibility applying tv-DbMS to the re-use of the video data, the database and its schemas. tv-DbMS supports the re-use of objects between data models whilst retaining access to the original context. The example is the provision of a hyperlink model. To make tv-DbMS a complete and comprehensive video database system, hyperlink support is also provided. A lot of video information systems include hyperlinks, but previously, there has been no investigation of the applicability of creating hyperlinks within the contents of video data and to link particular video objects.

The common video object tree generated for thematic indexing is used as the linkbase for tv-DbMS hypermedia, along with the segment and event entities stored in the database. This framework can be considered novel, as this provides a platform for further research to re-use objects in other application domains. This is achieved by creating hypermedia applications using the underlying video information, so that the user can either query the contents of a video or can also navigate within a video archive. The integrated system has two advantages: authors and users receive the benefits of having the facilities for retrieving video information, and at the same time they are able to use hypermedia navigation in creating their applications which means readers of their applications can explore the video information more naturally using unstructured hypermedia access. This integration demonstrated the advantages of creating links for conceptual story line tracking, navigate through different themes of video and to overall create hypermedia video application that can be accessed / viewed non-linearly, i.e. without the constrain of the time domain.

Another feature of tv-DbMS is its MPEG-7 compatibility. MPEG-4 is a standard that supports independent coding of an audio-video object. Information about object rendering for final presentation is also coded in MPEG-4, whereas MPEG-7 is the standard for providing further description about the object. The philosophy of tv-DbMS database is the same. A video object is considered as an entity and all its properties (like motion, speed, trajectory motion, location, 2D-3D, etc.) are stored in a table. Since MPEG-7 multimedia object information is rendered in XML, so a MS-Access → XML parser is also developed, that converts all properties of video objects into XML. The schema for this XML datatype is taken from the MPEG-7 audio-visual abstraction document (ISO/ITEC JTCI/ SC29/ WG11 N3545, July 2000, Beijing). MPEG-7 (discussed in detail in appendix A1) will become an international standard for multimedia coding in September 2001, and commercial products based on MPEG-7 are expected by July 2002.

Researchers in the area of computer vision have started accepting the fact that it is rather impossible to retrieve video information by using low-level features of the video, and metadata (in high level abstract) is utterly required to perform intelligent queries. This thesis has taken a significant step forward in this area, by developing a video database model, that can perform semantic and conceptual queries, by describing in detail how a video system can utilise the information of video object present inside the video clip, and how the knowledge about that video object can be used to perform intelligent queries. The result is a video information and retrieval system with thematic indexing capabilities.

8.2 Future Work

tv-DbMS concepts can take a number of new directions. In particular, manipulation of information within segments is an area that needs to be explored. Interactive-tv is also one of the domains where applications like tv-DbMS provide potential for the future. Agent technology is another domain which can be considered as a future direction for tv-DbMS.

8.2.1 Applying automatic video processing and retrieval techniques for tv-DbMS

To improve the scalability of tv-DbMS, an ideal system should combine the tv-DbMS functionalities with automated video processing techniques. To achieve this, existing automatic video indexing and retrieval techniques (discussed in chapter 2 an 3) could be used to provide indexing and retrieval tools for the original video materials in the system. The following concepts could be implemented in future to tv-DbMS:

8.2.1.1 Automating Scene Composition

The manipulation of information within a segment could involve the introduction of a new object or the removal of an existing object from a segment. Manipulation of information in a segment will not only allow us to create a segment with added information, it will also permit customisation of the information contained in the segment. Scene composition could be investigated as one of the potential techniques.

As shown in figure 8.1, given a set of objects (e.g. back ground, anchorperson, text), we can automatically compose the objects to create a scene. Similarly, all the scenes required in a composition could be created. Currently, scenes are pre-composed and stored in the video archive. The objective of automatic scene composition is to achieve a dynamic and visually rich composition for a digital video. A visually rich composition can be achieved by a selection of interesting and informative video objects from any composition to create new compositions. Also, the same objects can be reused to create different scenes. This is defined in the following two types of scene composition process:

- Aggregate Scene Composition: A composite of independent but related objects (figure 8.1)
- Partial Scene Composition: Replacing objects in a composite. E.g. replacing a talking head with a location scene

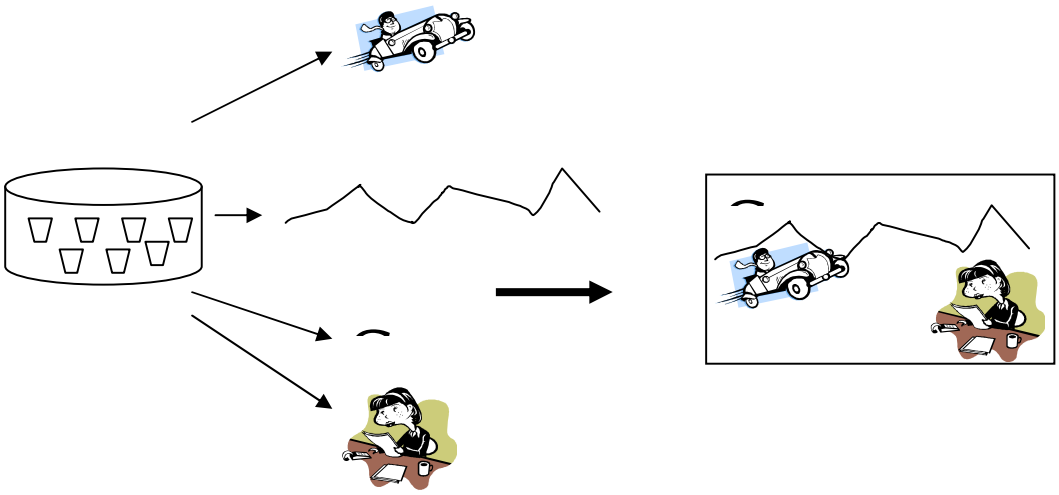


Fig 8.1: Segments composition

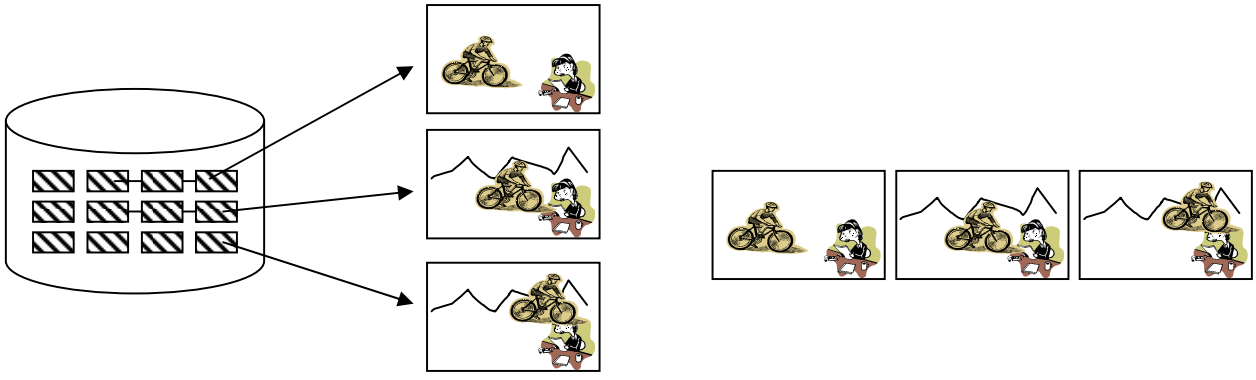


Figure 8.2: Automatic scene composition concept

Objects in a scene can be dropped or replaced only if visual objects are available and techniques to form a composite are there. Using this technique, the following objectives could be achieved:

- Replacement of interesting visual contents (e.g. location hosts) with dead visual content (e.g., talking head).
- Enrichment of content by including "added value" information
- Object archive will be smaller due to reuse of content

8.2.1.2 Automating Thematic Indexing

As discussed in chapter 5 and chapter 6, thematic indexing is created on top of video objects and supports hierarchies. To make this module more efficient, a tool to generate the hierarchies automatically would be essential. Agent technology is one of the ways to implement this sort of tool, where an agent traverses the length and breadth of the database and generates the hierarchies of the video data object. For example, an intelligent-mobile agent can efficiently retrieve data for thematic indexing, if the database is stored on some server.

Another improvement could be done in the form of personalised data sets. That is the user could have the option to create his or her own thematic indexing links and nodes. This could also support personalising queries and also personalising the complete video document.

To improve the speed and efficiency of thematic indexing to the latent semantic analysis (LSA) approach could be implemented. LSA is a technique for extracting the meaning and context of metadata by statistically analysing large bodies of text. One of its main features is that it induces correlations between words and topics without those correlations having to be explicitly specified. LSA obtains it 'knowledge' by first

processing a large corpus of text, creating a semantic space. This matrix of metadata, in which each row corresponds to a word, and each column to a passage of text. The values in the matrix are set to a weighted indication of the frequency of the word in each passage. Next, a process called singular value decomposition (SVD) is applied to the matrix, which decomposes it into three other matrices of much smaller dimension. These three matrices, when multiplied together, produce an approximation of the original matrix. It is when this dimensionality is decreased that the semantic relation between words is induced.

These three matrices can be used to derive vectors for each word or passage; the vectors could then be used to judge the similarity between topics. The cosine between the vectors is usually used; the higher the cosine, the greater the similarity.

The advantage of using LSA is that it divides the metadata into small clusters, and retrieving these small clusters is much quicker and efficient. The LSA technique has already been applied to textual information retrieval, in the form of latent semantic indexing (Dumais et al, 1988; Letshe & Berry, 1997). The LSA technique is not applied to multimedia information directly, but to text and HTML tags. The points in the semantic space and the points in the visual feature space are combined to make a single feature space, and this can be used as the basis for multimedia retrieval. This type of approach used at the lower level of thematic indexing generation, could rapidly increase the speed and efficiency of retrieving similar video clips.

8.2.2 Application of tv-DbMS framework in interactive tv

We have applied the tv-DbMS framework to the documentaries of Earl Mountbatten of Burma (13 one hour episodes) and an hour long video of Prof. Hall's inaugural lecture. It would be interesting to over time perform some evaluation on very large video documentary database such as the video archives of BBC or any other television channel. Target applications for tv-DbMS could be educational archives, documentaries, soaps, and interactive shows (e.g. Big Brother). An interesting application would be

story tracking a particular character in soaps. Since these soaps (e.g. EastEnders, BBC 1) are on air for many years (in the case of EastEnders, more then 15 years!) it becomes very difficult for the viewer to remember the background and story of every character. The story tracking module of tv-DbMS could be very useful for this sort of querying, e.g. the viewer can track the high level stories and 'histories' about 'Sonia' or 'Pat' in EastEnders.

The work on tv-DbMS started with a vision of a complete interactive television environment, where the viewer has the capability to watch what he/she wants to see from a variety of channels and can modify the contents of a particular program according to his or her own needs. Television or cinema is no longer a broadcasting media, viewers are now having their own choice. Channel 4's program 'Big Brother[11]' is one of the examples, where millions of viewers throughout U.K. were picking up an individual, to exclude from the show. Though this was one of the early signs of interaction where the viewers were sending their choice through telephones, but the popularity of this program showed that viewers are ready to get involve in the program and want to take decisions regarding the show. The eventual development of this interaction will lead to dramatic changes in television and films, where metadata will be considered as one of the important aspects, like sound and colour.

It may take a long time to develop these sorts of complete and comprehensive interactive programs, and at the moment much research is in the field of trying to access and annotate video objects within non-interactive digital videos. Whilst tv-DbMS has demonstrated the required functionalities to support video retrieval, an ideal interactive-tv would combine the features of video retrieval, automatic content based analysis and on top the users choice to change or modify a particular video clip to viewers desires. The proposal of such a system can be used as a starting point to realise such an ideal video information system.

[11] Big Brother was on air for 40 days on Channel 4 during July / August 2000

A.1 MPEG Overview

The Moving Picture Coding Experts Group (MPEG) is a working group of ISO/IEC, in charge of the development of international standards for compression, decompression, processing and coded representation of moving pictures, audio and their combination.

The purpose of MPEG is to produce standards. The first three standards produced by MPEG were:

MPEG-1, as standard for storage and retrieval of moving pictures and audio on digital storage media (officially designated as ISO/IEC 111172, in 5 parts). The picture and audio quality is similar to VHS devices.

MPEG-2 is a standard for digital television (officially designated as ISO/IEC 13818, in 9 parts). The applications include High Definition Television, DVD, digital satellite and cable transmissions.

MPEG-4 is a standard for multimedia applications that supports the creation of rich, reusable, and interactive multimedia content that can be used by different distribution networks (broadcasting, internet, CDs, etc.) and terminals (PCs with web browsers, TV sets, Set-Top Boxes, DVD players, etc). MPEG-4 is the first real multimedia representation standard, allowing interactivity and a combination of natural and synthetic materials, coded in form of objects that are integrated to compose multimedia presentations (WG11/N1901).

MPEG-7 offers a comprehensive set of audiovisual description tools to create descriptions, which will form the basis for applications enabling the needed quality access to content, which implies good storage solutions, high-performance content identification, proprietary assignation, and fast, ergonomic, accurate and personalized filtering, searching and retrieval. The scope of this standard is a broad spectrum of

multimedia applications, and the broad number of audiovisual features of importance in such context. The question of identifying and managing content is not just restricted to database retrieval applications such as digital libraries, but extends to areas like broadcast channel selection, multimedia editing and multimedia directory services.

A.1.1 MPEG-7

MPEG-7, also known as "Multimedia Content Description Interface", will extend the limited capabilities of proprietary solutions in identifying content that exist today, notably by including more data types. MPEG-7 will specify a standard set of descriptors that can be used to describe various types of multimedia information. MPEG-7 will also standardise ways to define other descriptors as well as structures (Description Schemes) (WG11/N3545) for the descriptors and their relationships. MPEG-7 will also standardise a language to specify description schemes, i.e. a Description Definition Language (DDL). AV material that has MPEG-7 data associated with it can be indexed and searched for. This 'material' may include: still pictures, graphics, 3D models, audio, speech, video, and information about how these elements are combined in a multimedia presentation ('scenarios', composition information). Special cases of these general data types may include facial expressions and personal characteristics.

A.1.1.1 MPEG-7 Objectives

The MPEG-7 standard aims at providing standardized core technologies allowing description of audiovisual data content in multimedia environments. It will extend the limited capabilities of proprietary solutions in identifying content that exist today, notably by including more data types. Audiovisual data content that has MPEG-7 data associated with it, may include: still pictures, graphics, 3D models, audio, speech, video, and composition information about how these elements are combined in a multimedia presentation (scenarios). Special cases of these general data types may include facial expressions and personal characteristics. MPEG-7 description tools do, however, not

depend on the ways the described content is coded or stored. It is possible to create an MPEG-7 description of an analogue movie or of a picture that is printed on paper, in the same way as of digitised content.

MPEG-7 Description tools allow to create descriptions (the result of using the MPEG-7 description tools at the users will) of content that may include:

- Information describing the creation and production processes of the content (director, title, short feature movie)
- Information related to the usage of the content (copyright pointers, usage history, broadcast schedule)
- Information of the storage features of the content (storage format, encoding)
- Structural information on spatial, temporal or spatio-temporal components of the content (scene cuts, segmentation in regions, region motion tracking)
- Information about low level features in the content (colours, textures, sound timbres, melody description)
- Conceptual information of the reality captured by the content (objects and events, interactions among objects) (WG11/N3545)

All these descriptions are of course coded in an efficient way for searching, filtering, etc.

To accommodate this variety of complementary content descriptions, MPEG-7 approaches the description of content from several viewpoints. Currently five viewpoints are defined: Creation & Production, Media, Usage, Structural aspects and Conceptual aspects. The five sets of description elements developed on those viewpoints are presented here as separate entities. However, they are interrelated and can be combined in many ways. Depending on the application, some will present and others can be absent or only partly present.

A description generated using MPEG-7 description tools will be associated with the content itself, to allow fast and efficient searching for, and filtering of material that is of interest to the user. The type of content and the query do not have to be the same; for example, visual material may be queried using visual content, music, speech, etc. It is the responsibility of the search engine and filter agent to match the query data to the MPEG-7 description.

A.1.1.2 MPEG-7 finale

MPEG-7, like the other members of the MPEG family, is a standard representation of audio-visual information satisfying particular requirements. The MPEG-7 standard builds on other (standard) representations such as analogue, PCM, MPEG-1, -2 and-4. One functionality of the standard is to provide references to suitable portions of them. For example, perhaps a shape descriptor used in MPEG-4 is useful in an MPEG-7 context as well, and the same may apply to motion vector fields used in MPEG-1 and MPEG-2.

MPEG-7 descriptors do, however, not depend on the ways the described content is coded or stored. It is possible to attach an MPEG-7 description to an analogue movie or to a picture that is printed on paper. Even though the MPEG-7 description does not depend on the (coded) representation of the material, the standard in a way builds on MPEG-4, which provides the means to encode audio-visual material as objects having certain relations in time (synchronisation) and space (on the screen for video, or in the room for audio). Using MPEG-4 encoding, it will be possible to attach descriptions to elements (objects) *within* the scene, such as audio and visual objects. MPEG-7 will allow different granularity in its descriptions, offering the possibility to have different levels of discrimination.
Because the descriptive features must be meaningful in the context of the application, they will be different for different user domains and different applications.

The elements that MPEG-7 standardizes will support a broad range of applications (for example, multimedia digital libraries, broadcast media selection, multimedia editing, home entertainment devices, etc.). MPEG-7 will also make the web as searchable for multimedia content as it is searchable for text today. This would apply especially to large content archives, which are being made accessible to the public, as well as to multimedia catalogues enabling people to identify content for purchase. The information used for content retrieval may also be used by agents, for the selection and filtering of broadcasted "push" material or for personalized advertising. Additionally,

MPEG-7 descriptions will allow fast and cost-effective usage of the underlying data, by enabling semi-automatic multimedia presentation and editing.
All domains making use of multimedia will benefit from MPEG-7. Following are some of the practical applications that can use MPEG-7 effectively:

- Digital libraries, Education (image catalogue, musical dictionary, Bio-medical imaging catalogues,),
- Multimedia editing (personalised electronic news service, media authoring)
- Cultural services (history museums, art galleries, etc.),
- Multimedia directory services (e.g. yellow pages, Tourist information, Geographical information systems)
- Broadcast media selection (radio channel, TV channel, internet channel),
- Journalism (e.g. searching speeches of a certain politician using his name, his voice or his face),
- E-Commerce (personalised advertising, on-line catalogues, directories of e-shops),
- Surveillance (traffic control, surface transportation, non-destructive testing in hostile environments, etc.),
- Investigation services (human characteristics recognition, forensics),
- Home Entertainment (systems for the management of personal multimedia collections, including manipulation of content, e.g. home video editing, searching a game, karaoke, etc.),
- Social (e.g. dating services, chatting services, internet pals, etc.).

A.1.1.3: MPEG-7 and tv-DbMS compatibility

MPEG-7 is a multimedia content description standard, which closely addresses how humans expect to interact with computer systems because it develops rich descriptions that reflect those expectations. MPEG-7 is about the future of media in the 21st century.

MPEG-7 provides a comprehensive and flexible framework for describing the content of multimedia. To describe content implies knowledge of elements it consists of, as well as, knowledge of interrelations between those elements. This interrelation can be termed as 'concept' which tv-DbMS defines clearly in its data model. Along with tv-DbMS database structure was also made by keeping in view the requirements of MPEG-7 standards, and thus can be easily incorporated with any MPEG-7 application.

tv-DbMS can be defined as a system with comparable functionalities as of MPEG-7, or one of the very early applications incorporating MPEG-7 standard. The Following features of tv-DbMS make it compatible with MPEG-7.

While creating the database structure of tv-DbMS, it was made sure that all the tables, rows and columns in tv-DbMS database structure should define MPEG-7 data structures. All the properties defined in AV (audio video) object of MPEG-7 are also made the part of tv-DbMS media object. For example, while defining a segment, the components like camera motion, motion speed, motion trajectory, parametric motion, motion activity were made MPEG-7 compatibles. Along with, an XML tree generator filter is also created, so that all the video metadata can be converted into XML, so that it can be used further for MPEG-7 applications.

Since MPEG-7 is still in its developing stage and the first ISO standard for MPEG-7 will be available in September 2001, so a lot of things are been altering during the development phase. The final draft for Description Definition Language (DDL) is not available yet, but care was taken while developing the query logic of tv-DbMS, that the query language and query filters used in tv-DbMS should be compatible to the DDL of MPEG-7.

Since DDL is not itself a query module. It is just a language to define the description about an AV object or a concept, so any query made on these objects or concepts that can handle the metadata defined for these objects, can be considered as compatible. This

hold true for the tv-DbMS query module, as it incorporates the compatible metadata for MPEG-7 AV objects.

tv-DbMS can play, edit and annotate MPEG-4 (.asf) formatted videos. As mentioned in section A.2.4, MPEG-4 standard has the capability to encode audio-visual material as objects having certain relation in time (synchronisation) and space (spatial aspect). Along with MPEG-7 allows different granularity in its descriptions, offering the possibility of different levels of discrimination. This is very similar to tv-DbMS hierarchical model for media objects, defined in chapter 4.

Appendix A.2

RDF DTD for tv-DbMS:

```
<rdf: RDF
                xmlns:rdf = "http://www.w3.org/TR/WD-rdf-syntax#"
                xmlns:rdfs = "http://www.w3.org/TR/WD-rdf-schema#"
                xmlns:dc = "http://purl.org/metadata/dublin_core#"
>

<rdfs:Class ID="tv-DbMS_Video_Document">
<rdfs:comment> Class to represent a tv-DbMS database document </rdfs:comment>
</rdfs:Class>

<rdfs:comment> Define all the database entities for tv-DbMS in RDF </rdfs:comment>
<rdfs:comment> Developed on Nov 25, 2000 </rdfs:comment>
<rdfs:comment> IAM labs, ECS Dept: </rdfs:comment>
<rdfs:comment> University of Southampton </rdfs:comment>

<rdf:PropertyType ID ="Title">
<rdfs:comment> Define the title of the video document </rdfs:comment>
<rdfs:Domain rdf:resource="#Video_Document">
<rdfs:range rdf:resource="http://purl.org/metadata/dublin_core#Title"/>
</rdfs:PropertyType>

<rdf:PropertyType ID ="Description">
<rdfs:comment> Define the description of the video document </rdfs:comment>
<rdfs:Domain rdf:resource="#Video_Document">
<rdfs:range rdf:resource="http://purl.org/metadata/dublin_core#Description"/>
</rdfs:PropertyType>

<rdf:PropertyType ID ="Subject">
<rdfs:comment> Define the Subject of the video document </rdfs:comment>
```

<rdfs:Domain rdf:resource="#Video_Document">
<rdfs:range rdf:resource="http://purl.org/metadata/dublin_core#Subject"/>
</rdfs:PropertyType>

<rdf:PropertyType ID ="Publisher">
<rdfs:comment> Define the Publisher of the video document </rdfs:comment>
<rdfs:Domain rdf:resource="#Video_Document">
<rdfs:range rdf:resource="http://purl.org/metadata/dublin_core#Publisher"/>
</rdfs:PropertyType>

<rdf:PropertyType ID ="Date">
<rdfs:comment> Define the Date of the video document </rdfs:comment>
<rdfs:Domain rdf:resource="#Video_Document">
<rdfs:range rdf:resource="http://purl.org/metadata/dublin_core#Date"/>
</rdfs:PropertyType>

<rdf:PropertyType ID ="Format">
<rdfs:comment> Define the Format of the video document </rdfs:comment>
<rdfs:Domain rdf:resource="#Video_Document">
<rdfs:range rdf:resource="http://purl.org/metadata/dublin_core#Format"/>
</rdfs:PropertyType>

<rdf:PropertyType ID ="Language">
<rdfs:comment> Define the Language of the video document </rdfs:comment>
<rdfs:Domain rdf:resource="#Video_Document">
<rdfs:range rdf:resource="http://purl.org/metadata/dublin_core#Language"/>
</rdfs:PropertyType>

<rdf:PropertyType ID ="Relation">
<rdfs:comment> Define the relation of the video document </rdfs:comment>
<rdfs:Domain rdf:resource="#Video_Document">
<rdfs:range rdf:resource="http://purl.org/metadata/dublin_core#Relation"/>
</rdfs:PropertyType>

<rdfs:coment>Define the segment class and its properties </rdfs:comment>

<rdfs:Class ID="Segment">
<rdfs:Comment>Class for representing a Segment Class </rdfs:comment>
<rdfs:subClassOf rdf:resource="Video_Document"/>
<rdfs:Class>

<rdf:PropertyType ID="Segment_ID"
<rdfs:comment> Entity related to a video segment </rdfs:comment>
<rdfs:domain rdf:resource="Video_Document">
<rdfs:range rdf:resource="#Segment">
</rdfs:PropertyType>

<rdf:PropertyType ID="Start_Time"
<rdfs:comment> Entity related to a video segment </rdfs:comment>
<rdfs:domain rdf:resource="Video_Document">
<rdfs:range rdf:resource="#Segment">
</rdfs:PropertyType>

<rdf:PropertyType ID="End_Time"
<rdfs:comment> Entity related to a video segment </rdfs:comment>
<rdfs:domain rdf:resource="Video_Document">
<rdfs:range rdf:resource="#Segment">
</rdfs:PropertyType>

<rdf:PropertyType ID="Person"
<rdfs:comment> Entity related to a video segment </rdfs:comment>
<rdfs:domain rdf:resource="Video_Document">
<rdfs:range rdf:resource="#Segment">
</rdfs:PropertyType>

<rdf:PropertyType ID="Location"
<rdfs:comment> Entity related to a video segment </rdfs:comment>
<rdfs:domain rdf:resource="Video_Document">
<rdfs:range rdf:resource="#Segment">
</rdfs:PropertyType>

<rdf:PropertyType ID="Object"
<rdfs:comment> Entity related to a video segment </rdfs:comment>
<rdfs:domain rdf:resource="Video_Document">
<rdfs:range rdf:resource="#Segment">
</rdfs:PropertyType>

<rdf:PropertyType ID="Concept"
<rdfs:comment> Entity related to a video segment </rdfs:comment>
<rdfs:domain rdf:resource="Video_Document">
<rdfs:range rdf:resource="#Segment">
</rdfs:PropertyType>

<rdf:PropertyType ID="Motion_Type"
<rdfs:comment> Entity related to a video segment </rdfs:comment>
<rdfs:domain rdf:resource="Video_Document">
<rdfs:range rdf:resource="#Segment">
</rdfs:PropertyType>

<rdf:PropertyType ID="Co_Ord_System"
<rdfs:comment> Entity related to a video segment </rdfs:comment>
<rdfs:domain rdf:resource="Video_Document">
<rdfs:range rdf:resource="#Segment">

</rdfs:PropertyType>

<rdf:PropertyType ID="Spatial_Units"
<rdfs:comment> Entity related to a video segment </rdfs:comment>
<rdfs:domain rdf:resource="Video_Document">
<rdfs:range rdf:resource="#Segment">
</rdfs:PropertyType>

<rdf:PropertyType ID="Camera_Follows"
<rdfs:comment> Entity related to a video segment </rdfs:comment>
<rdfs:domain rdf:resource="Video_Document">
<rdfs:range rdf:resource="#Segment">
</rdfs:PropertyType>

<rdf:PropertyType ID="Co_Ord_Are_3D"
<rdfs:comment> Entity related to a video segment </rdfs:comment>
<rdfs:domain rdf:resource="Video_Document">
<rdfs:range rdf:resource="#Segment">
</rdfs:PropertyType>

<rdf:PropertyType ID="Key_Points"
<rdfs:comment> Entity related to a video segment </rdfs:comment>
<rdfs:domain rdf:resource="Video_Document">
<rdfs:range rdf:resource="#Segment">
</rdfs:PropertyType>

<rdf:PropertyType ID="2D_3D_Spatio_Temporal"
<rdfs:comment> Entity related to a video segment </rdfs:comment>
<rdfs:domain rdf:resource="Video_Document">
<rdfs:range rdf:resource="#Segment">
</rdfs:PropertyType>

<rdf:PropertyType ID="Interval_3D"
<rdfs:comment> Entity related to a video segment </rdfs:comment>
<rdfs:domain rdf:resource="Video_Document">
<rdfs:range rdf:resource="#Segment">
</rdfs:PropertyType>

<rdf:PropertyType ID="Interval_2D"
<rdfs:comment> Entity related to a video segment </rdfs:comment>
<rdfs:domain rdf:resource="Video_Document">
<rdfs:range rdf:resource="#Segment">
</rdfs:PropertyType>

<rdf:PropertyType ID="Intensiy_Of_Action"
<rdfs:comment> Entity related to a video segment </rdfs:comment>
<rdfs:domain rdf:resource="Video_Document">

<rdfs:range rdf:resource="#Segment">
</rdfs:PropertyType>

<rdf:PropertyType ID="Direction_Of_Activity"
<rdfs:comment> Entity related to a video segment </rdfs:comment>
<rdfs:domain rdf:resource="Video_Document">
<rdfs:range rdf:resource="#Segment">
</rdfs:PropertyType>

<rdf:PropertyType ID="Temporal_Disturb_Of_Activity"
<rdfs:comment> Entity related to a video segment </rdfs:comment>
<rdfs:domain rdf:resource="Video_Document">
<rdfs:range rdf:resource="#Segment">
</rdfs:PropertyType>

<rdf:PropertyType ID="Spatial_Disturb_Of_Activity"
<rdfs:comment> Entity related to a video segment </rdfs:comment>
<rdfs:domain rdf:resource="Video_Document">
<rdfs:range rdf:resource="#Segment">
</rdfs:PropertyType>

<rdfs:coment>Define the event class and its properties </rdfs:comment>

<rdfs:Class ID="Event">
<rdfs:Comment>Class for representing a Event Class </rdfs:comment>
<rdfs:subClassOf rdf:resource="Video_Document"/>
<rdfs:Class>

<rdf:PropertyType ID="Event_Id"
<rdfs:comment> Entity related to an Event </rdfs:comment>
<rdfs:domain rdf:resource="Video_Document">
<rdfs:range rdf:resource="#Event">
</rdfs:PropertyType>

<rdf:PropertyType ID="Event_Description"
<rdfs:comment> Entity related to an Event </rdfs:comment>
<rdfs:domain rdf:resource="Video_Document">
<rdfs:range rdf:resource="#Event">
</rdfs:PropertyType>

<rdf:PropertyType ID="Event_Concept"
<rdfs:comment> Entity related to an Event </rdfs:comment>
<rdfs:domain rdf:resource="Video_Document">
<rdfs:range rdf:resource="#Event">
</rdfs:PropertyType>

<rdf:PropertyType ID="Event_Time"
<rdfs:comment> Entity related to an Event </rdfs:comment>
<rdfs:domain rdf:resource="Video_Document">
<rdfs:range rdf:resource="#Event">
</rdfs:PropertyType>

<rdf:PropertyType ID="Event_Date"
<rdfs:comment> Entity related to an Event </rdfs:comment>
<rdfs:domain rdf:resource="Video_Document">
<rdfs:range rdf:resource="#Event">
</rdfs:PropertyType>

<rdf:PropertyType ID="Event_Added_Info"
<rdfs:comment> Entity related to an Event </rdfs:comment>
<rdfs:domain rdf:resource="Video_Document">
<rdfs:range rdf:resource="#Event">
</rdfs:PropertyType>

Glossary

ANNOTATION

Some data stored separately, but associated with a segment. Typically an annotation provides some description of some aspect of the content of its associated segment.

AUDIO The audio component of any video document consisting of frequencies corresponding to a normally audible sound wave (20 Hz to 20,000 Hz).

AUDIO BANDWIDTH

The range of audio frequencies which directly influence the fidelity of a sound. The higher the audio bandwidth, the better the sound fidelity. The highest practical frequency that the human ear can normally hear is 20 kHz.

AUDIO EDITING

Similar to video editing. Various portions of audio material are combined and recorded onto the videotape in one continuous form.

AVI Audio-Video Interleave, which is a video format for Microsoft Windows.

BANDWIDTH

(See AUDIO BANDWIDTH and VIDEO BANDWIDTH)

BITRATE

The rate at which the compressed bit stream is delivered from the storage medium to the input of a decoder.

CCD (Charge Coupled Device)

A semiconductor device (IC) that converts optical images to electronic signals. CCDs are the most commonly found type of image sensor in consumer camcorders and video cameras.

CHROMINANCE

Portion of video signal that carries hue and saturation colour information. Also see luminance.

CODEC

Component of a video system that encodes video data into its compressed format, and decodes data from its compressed format. Video data is stored in a compressed format. When an effect is applied to one or more tracks of video, the video must be uncompressed in order to compute (render) the effect. Historically, codecs have been implemented on specialized hardware. Software codecs are now more prominent because desktop computers are now fast enough to support video processing.

COMPRESSION

The process of electronically processing a video picture to make it use less storage or to allow more video to be sent down a transmission channel.

CLIP A sequence of frames regarded as a logical unit.

DC Dublin Core

DCT Discrete Cosine Transform

DDL Data Description Language

DLS Dynamic Link Service

DS Description Scheme

FRAME

One complete still image of video media. Video media is made up of a series of frames. Each video frame has two interlaced fields. A frame contains lines of spatial information of a video signal. For progressive video, these lines contain samples starting from one time instant and continuing through successive lines to the bottom of the frame. For interlaced video a frame consists of two fields, a top field and a bottom field. One of these fields will commence one field period later than the other.

HUE

Often used synonymously with the term tint. It is the dominant wavelength which distinguishes a colour such as red, yellow, etc. Most commonly, video hue is influenced by:

- A camera's white balance
- Scene lighting

INDEX

A special database object that makes finding information in a database table much faster by storing pointers to where needed data resides on-disk. Indexes are essential for good database performance.

JPEG (Joint Photographic Experts Group)

JPEG is a digital compression standard for still video images that allows the image to occupy less memory or disk space. Like the MPEG* standard, it includes options for trading off between storage space and image quality.

KEY FRAME

A frame at which a set of specific parameters is assigned. FX generation automatically calculates differences between key frames in a clip and adjusts the frames accordingly. For example, a title can be instructed to move between different coordinates on the screen, each coordinate is associated with a specified key frame.

LCD (Liquid Crystal Display)

A screen for displaying text/graphics based on a technology called liquid crystal, where minute currents change the reflective ness or transparency of the screen.

LINEAR EDITING

Editing using media like tape, in which material must be accessed in order (e.g., to access scene 5 from the beginning of the tape, one must proceed from scene 1 through scene 4). (See NONLINEAR EDITING)

LUMINANCE

The degree of brightness (black and white portion of the video signal) at any given point in the video image. A video signal is comprised of luminance, chrominance* (colour information) and sync. If luminance is high, the picture is bright and if low the picture is dark. Changing the chrominance does not affect the brightness of the picture.

METADATA

The data about data. In tv-DbMS, it is the additional data provided by the annotator to support video processing.

MODEL In this thesis: abbreviation of video data model.

MPEG (Moving Picture Expert Group)

MPEG is a digital compression standard for moving video images that allows the images to occupy less memory or disk space. Like the JPEG standard, it includes options for trading off between storage space and image quality.

NONLINEAR EDITING

The process of editing using rapid retrieval (random access) computer controlled media such as hard disks, CD-ROMs and laser discs. Its main advantages are:

- Allows you to reorganize clips or make changes to sections without having to redo the entire production.
- Very fast random access to any point on the hard disk (typically 20-40 ms).

NTSC (National Television Standards Committee)

Standard of colour TV broadcasting used mainly in the United States, Canada, Mexico and Japan, featuring 525 lines per frame and 30 frames per second. (See PAL and SECAM)

OD Object Descriptor

ODBC

Open Database Connectivity; a Microsoft standard defining how Windows applications access database data.

OHS Open Hypermedia Systems

OSI

The Open Systems Interconnection Reference Model was formally initiated by the International Organization for Standardization (ISO) in March, 1977.

PAL (Phase Alternate Line)

The European colour TV broadcasting standard featuring 625 lines per frame and 25 frames per second. (See NTSC and SECAM)

PAN

When used in reference to video, it is the sweeping movement of a camera across a scene or the appearance.

PLAYBACK

The process whereby a videotape is displayed on a monitor. During playback, use of a video processor such as the Video Equalizer can be used to alter, enhance, correct or restore a signal.

PIXEL

A single picture element. The smallest element in a graphic image. Pixels are combined with other pixels to make up a graphic image. Picture quality increases as the number of pixels increase in a measured area of an image.

PRE-PRODUCTION

In film making, the generation of scripts and storyboards for the film to be produced.

POST-PRODUCTION

All production work done after the raw video footage and audio elements have been captured. Editing, titling, special effects insertion, image enhancement, audio mixing and other production work is done during post-production.

QUICKTIME

System software from Apple Computer, Inc. that enables the storage, editing, and playing of digitised video and audio media on a computer.

RASTER

The pattern of parallel horizontal scanning lines, traced by a video monitor's electron beam, making up a video image.

RDF Resource Description Framework

RGB (Red/Green/Blue)

The basic components of a colour video signal. Using a colour encoder, in conjunction with sync information, a complete composite video signal comprising luminance, chrominance and sync can be generated from RGB.

SCENE

A continuous block of storytelling either set in a single location or following a particular character. The end of a scene is typically marked by a change in location, style, or time.

SCRIPT

A general term for a written work, detailing story, setting, and dialogue. A script may take the form of a screenplay, shooting script, lined script, continuity script, or a spec script.

SECAM (Sequential Couleur A'memorie)

The video standard used in some European and surrounding countries. In countries using the SECAM standard, most video production is done using PAL and converted to SECAM prior to transmission. (See NTSC and PAL)

SHOT A single uninterrupted sequence of frames from one camera.

SQL

Structured query language; an industry-standard language for inserting, updating, deleting, and retrieving data from a relational database.

STORYBOARD

A sequence of pictures created by a production illustrator to communicate the desired general visual appearance on camera of a scene or movie.

TIME INTERVAL

The time between two instants. In a system that models a time domain using granules (defined in the addendum), an interval may be represented by a set of contiguous granules.

TIMELINE The graphic representation of a program displayed in the Sequencer window.

VHS (Video Home System)

Consumer videocassette record/playback tape format using half-inch wide magnetic tape.

VIDEO BANDWIDTH

The range between the lowest and highest signal frequency of a given video signal. In general, the higher the video bandwidth, the better the quality of the picture. Video bandwidths used in studio work typically vary between 3 and 12 MHz. Consumer VCRs are generally capable of 3-5.5 MHz.

VIDEO CONTENT

An interpretation of the meaning of the moving image presented to the viewer as the video is played back.

VIDEO DOCUMENT

The largest logical unit of information in a video; it might be the whole video recording itself.

VIDEO EDITING

A procedure for combining selected portions of video footage in order to create a new combined version. During video editing, special effects such as wipes, dissolves, inserts, etc. can be added. Professional editing is done using time code recorded on every frame of the magnetic tape allowing single frame accuracy. Audio editing is often carried out simultaneously with video editing.

W3C World Wide Web Consortium

WWW World Wide Web

ZOOM

On a camera, to change the focal length to/from wide-angle and telephoto. In post-production, an editing filter that simulates the effect of having a camera move in very close to the subject, objects, or areas in a frame; or move away from the subject and display a wide view of the entire frame.

Bibliography

Acharya, S., Smith, B., Parnes, P., (2000) **Characterizing user access to videos on the World Wide Web,** *Multimedia Computing and Networking, 2000*, San Jose, CA, USA,

Ahanger, G., Benson, D., and Little, T.D.C., (1995) **Video Query Formulation,** in proceedings of *SPIE on storage and retrieval for image and video databases III*, vol 2420, 9-10 Feb 1995, San Jose, California, USA.

Ahanger, G., Little T.D.C, (1997a), **A system for customised News Delivery from Video Archives,** *MCL Technical Report no 06-06-1997*, Multimedia Communications Laboratory, Dept. of Electrical and Computing Engineering, Boston University Boston, USA

Ahanger, G., Little, T.D.C. (1997b), **Easy ED: An Integration of technologies from Multimedia Education**, in proceedings of *WebNet '97,* October 1997, Toranto, Canada.

Alshuth, P., Hermes, T., Voigt, L. and Herzog, O., (1997), **On Video Retrieval: Content Analysis by ImageMiner**, in proceedings of *SPIE Photonics West '97,* Volume no. 3312, pp. 236 – 247…

Anker, T., Dolev, D., Keidar, I., (1999) **Fault tolerant video on demand services,** *The 1999 19th IEEE International Conference on Distributed Computing Systems (ICDCS'99)*, Austin, TX, USA, 05/31-06/04/99

Akutsu, A., Tonomura Y., Hashimoto, H., and Ohba, Y., (1992) **Video indexing using motion vectors,** in proceedings of *SPIE Visual Communications and Image Processing '92,* vol. 1818, pp. 1522 – 1530, SPIE, USA.

Blackburn, S., DeRoue, D., (1998), **Amphion: Open Hypermedia Applied to Temporal Media,** in proceedings of *4th Workshop on Open Hypermedia Systems*, Hypertext '98 Pittsburgh, PA, June 20-24, 1998

Boris V.D, Loukachevitch, N.V., Yudina, T.N., (1997), **Conceptual Indexing Using Thematic Representations,** Centre for Information Research, Moscow State University, Russia, 1997.

Brazilay, R., Elhadad, M., (1997), **Using lexical chains for text summarisation,** in proceedings of *ACL/ EACL workshop for Intelligent Scalable Text Summarisation, 1997,* Madrid, Spain.

Brondmo, H.P. and Davenport, G. (1990) **Creating and viewing the Elastic Charles – a hypermedia Journal.** In Mc. Aleese et. Al. (eds.) *Hypertext: State of the Art*, pp. 43-51, Oxford: Intellect Limited

Bryan-Kinns, N., (2000), **VCMF: A Framework for Video Content Modelling,** *Multimedia Tools and Applications,* No. 10 vol. 1, pp. 23-45, January 2000, Kluwer Academic Press, Netherlands.

Bryan-Kinns, N., (1998), **A Framework for Modelling Video Content,** *Ph.D. Thesis,* Queen Marry and Westfield College, University of London, August 1988.

Caetano, A., Guimaraes, N., (1998) **A Model for Content Representation of Multimedia Information.** Presented at a workshop *The Challenge of Image Retrieval* organised by the British Computer Society 1998.

Carr, L., Davis, H., Hall, W., and Hey, J., (1996) **Using the world wide web as an electronic library,** in proceedings of *First ACM International Conference on Digital Libraries, DL'96,* ACM

Carrer, M., Ligresti, L., Ahanger, G., Little, T.D.C., (1997), **An Annotation Engine for Supporting Video Database Population**, *Multimedia Tools and Applications, an International Journal*, November 1997, Volume 5, Number 3, Kluwer Academic Publishers Netherlands.

Carriera, M., Casebolt, J., Desrosiers, G., and Little, T.D.C., (1995), **Capture-time indexing paradigm, authoring tool, and browsing environment for digital broadcast video,** in proceedings of *SPIE on Multimedia Computing and Networking 95,* 6 – 8 Feb. 1995, vol 2417, San Jose, CA, USA.

Cascia M.L, Ardizzone, A., (1996), **JACOB: Just a Content-based query system for video databases**, in *IEEE International conference on Acoustics, Speech and Signal Processing,* May 7-10, 1996, Atlanta, GA, USA.

Chang, S.K. and Hsu, A. (1992) **Image Information Systems: Where do we go from here?** *IEEE Transactions on Knowledge and Data Engineering*, October 1992, Vol. 4, No. 5, pp. 431 – 442

Christel, M., Kanade, T., Mauldin, M., Reddy, R., Sirbu, M., Stevens, S., and Wactlar, H., (1994), **Informedia Digital Video Library**. *Communications of the ACM* , Volume 4 April 1994 pp. 57-58, ACM Press USA.

Christel, M., Martin D.J., (1998), **Information Visualisation within a Digital Video Library,** *Journal of Intelligent Information Systems,* Volume 11, No. 3, Nov/Dec, 1998, pp. 235-257.

Christel, M., Olligschlaeger. A., (1999), **Interactive Maps for a Digital Video Library,** proceedings of *IEEE International Conference and Systems,* June 7 – 11, 1999, Florence, Italy.

Christel, M., (1999), **Visual Digests for News Video Libraries,** proceedings of *ACM Multimedia 1999,* Oct 30 – Nov 5, pp. 303 – 311, Orlando, FL, USA.

Codd, E., (1970), **A relational model for large shared data banks**, *Communications of ACM*, vol 13, no. 6, pp. 377-387, July 1970

Colombo, C., Bimbo, A.D., Pala, P., (1999), **Semantics in Visual Information Retrieval,** *IEEE Multimedia*, July – September 1999, pp 38 – 53, IEEE, USA.

Davenport, G., Murtaugh, M., (1997) **Automatist storyteller systems and the shifting sands of story,** *IBM Systems Journal*, vol 36, no 3, 1997 pp446-456

Davis, H.C, Hall, W., Heath, I., Hill, G., and Wilkins, R., (1992) **Towards an integrated information environment with open hypermedia systems,** in proceedings of *Fourth ACM Conference on Hypertext, ECHT'92,* Milan Italy, December 1992, pp 181 – 190, ACM Press

Deardorff, E., Little, T.D.C., Marshall, J.D., Venkatesh, D., and Walzer, R., (1994), **Video Scene Decomposition with the Motion Picture Parser,** in proceedings of *SPIE on storage and Retrieval for image and video databases II,* vol. 2187, Feb. 1994, San Jose, CA, USA.

Devadiga, S., Koshiba, D.A., Gargi, U., Oswald, S., Kasturi R., (1995), **A semi automatic video database system,** in proceedings, *SPIE Conference on storage and retrieval in Image and video databases.* 1995.

Dumais, S.T., Furnas, G.W., Landauer, T.K., (1988), **Using latent semantic analysis to improve access to textual information,** *ACM Computer – Human Interaction '88 proceedings,* pp 281-283, ACM press, USA.

Elliott, E L. (1993), **WATCH - GRAB - ARRANGE - SEE: thinking with motion images via streams and collages**, *M. S. V. S. thesis*, Massachusetts Institute of Technology Media Laboratory, 1993.

Elmagarmid A.K. and Jiang. H., (1997), **Video Database Systems: Issues, Products and Applications**, Kluwer Academic Publishers, AH Dordrecht, Netherlands.

Fountain, A., Hall, W., Heath, I., Davis, H., (1990), **Microcosm: An open model with Dynamic Linking**, in proceedings of *European Conference on Hypertext*, pp. 298 – 311, Cambridge University Press.

Galvez, P., Newman, H., (1997) **Networking, videoconferencing and collaborative environments,** *The 1997 International Conference on Computing in High Energy Physics*, CHEP, Berlin, Germany, 04/07-11/97

Garzotto, F., Mainetli, L., Paolini, P., (1995), **Hypermedia Design Analysis and Evaluation Issues,** *Communications of ACM, pp. 78 – 86, Vol 38.*

Gauch, S., Li, W., Gauch, J. (1996), **The VISION Digital Video Library**, *ACM Digital Libraries*, 1996, ACM Press USA.

Gibbs S., Breiteneder, C., Tsichritzis, D., (1993), **Audio Video Data Model, An Object-Oriented Approach**, *9th International Conference on Data Engineering*, Proceedings, Sponsored by IEEE USA, ch-74, pp381-390.

Gibbs, S., Breiteneder, C., Tsichritzis, D., (1992), **Modelling of Audio/ Video Data,** Entity-Relation Approach, *ER-92*, ch-25, pp 323-339.

Gibbs, S., Tsichritzis, D., (1995), **Multimedia Programming, Objects, Environments and Frameworks**, Addison Wesley / ACM Press, UK.

Goose, S., Hall, W., (1995), **The development of a sound viewer for an open hypermedia system,** *The new review of Hypermedia and Multimedia, Applications and research,* vol. 1, No. 5, pp. 213-231.

Grønbæk, K Wiil. U.K., (1997), **Towards a common reference architecture for open hypermedia systems,** *Journal of Digital Information,* 1997.

Grosky, W.I (1994) **Multimedia Information Systems**, *IEEE Multimedia,* Spring 1994, pp. 12-24

Grosky, W.I (1999) **Managing Multimedia Information in Database Systems**, *Communications of the ACM,* December 1997, Vol. 40, No. 12, pp. 73-80

Hampapur, A., Jain, R., and Weymouth, T.E., (1995), **Indexing in Video Databases,** in proceedings of *SPIE on storage and retrieval of image and video databases III,* vol. 2420, 9-10 Feb, 1995, San Jose, CA, USA.

Hall, W., (1994) **Ending the Tyranny of the Button,** *IEEE Multimedia,* 1(1), pp. 60 –68, 1994, IEEE press USA.

Hall, W., Davis, H., Hutchings, G., (1996) **Rethinking hypermedia: The Microcosm approach.** Kulwer academic press

Hardman, L., Bulterman, D.C.A and Rossum, G.V., (1993) **The Amsterdam hypermedia model: extending hypertext to support real multimedia,** *Hypermedia,* Vol. 5, No. 1, pp. 47 –69, Taylor Graham publishing.

Hauptmann, A.G., Witbrock, M.J., (1998) **Story segmentation and detection of commercials in broadcast news video,** *The 1998 IEEE Forum on Research and Technology Advances in Digital Libraries*, IEEE ADL'98, Santa Barbara, CA, USA, pp. 168-179,

Heath, I., (1992) **An Open Model for Hypermedia: Abstracting links from Documents.** *PhD Thesis,* Department of Electronics and Computer Science, University of Southampton.

Hjelsvold R., (1994a), **Video Information Contents and Architecture,** in proceedings of the *4th international Conference of Extending Database Technology*, Cambridge UK, March 1994, pp 259 - 272

Hjelsvold R., (1994b) **Digital Television Archives - Combining Computer Technology and Video**. Presented at the *IASA/FIAT Annual Conference 1994*, Bogensee, Germany, September 1994.

Hjelsvold R., Midstraum R., (1994), **Modelling and Querying Video Data**, in proceedings of the *20th VLDB Conference*, Santiago, Chile, September 1994, pp 686 – 694.

Hjelsvold, R., Midtstraum R., (1995), **A temporal foundation of video databases,** in proceedings of *the International Workshop on Temporal databases*, Zurich, Switzerland, September 1995.

Hjelsvold, R., Midtstraum, R., Sandsta, O., (1995), **Databases for Video Information Sharing,** in proceedings of *SPIE on Storage and Retrieval for Image and Video Databases III*, Vol. 2420, 9-10 February 1995, San Jose California, pp 268-279

Hunter, J., (1999), **MPEG-7, Behind the Scenes,** *D-Lib Magazine,* Vol. 5, No. 9, September 1999, ISSN: 1082-9873.

Jain, R., (1993), **NSF Workshop on Visual Information Management systems,** *SIGMOD Record,* Vol 22, No. 3, September 1993, pp 57-75

Jain, R., Hampapur, A., (1994), **Metadata in Video Databases**, *SIGMOD Record*, volume 23, No. 4, December 1994, pp 27 -33

John Z Li, Iqbal A. Goralwalla, M.Tamer Ozsu, Duane Szafron (1997), **Modelling Video temporal relationships in an object database management system,** Department of Computing Science, University of Alberta, Canada 1997

John Z Li, M.Tamer Ozsu (1998), **STARS: A spatial attributes retrieval system for images and videos**, Department of Computing Science, University of Alberta Canada

Kalva, H., Eleftheriadis, A., Zamora, J., (1999) **Delivering object-based audio-visual services,** *IEEE Transactions on Consumer Electronics [IEEE Trans Consum Electron],* vol. 45, no. 4, pp. 1108-1111, 1999

Kanda, J., Wakimoto, K., Abe, S., Tanaka, S., (1998), **Video hypermedia authoring using automatic object tracking,** in proceedings of *Storage and Retrieval for Still Image and Video Database VI*, volume 3312, 38-30 January, 1998, San Jose California, pp 106 – 115.

Kelly, P.H., Gupta, A., Jain, R., (1996), **Visual Computing meets data modelling: Defining objects in multi-camera video databases,** in proceedings *of Storage and Retrieval for Still Image and Video Database IV*, volume 2670, 1-2 February 1996, San Jose, California, pp 120 -131

Khoja, S., Hall, W. (1999a), **Thematic Video Indexing to support video database retrieval and query processing,** in proceedings of *SPIE Multimedia Storage and Archiving Systems IV,* September 1999, Boston, MA, pp 371-380

Khoja, S., Hall, W. (1999b), **tv-DbMS - A video database management system incorporating a thematic indexing model,** *ACM Multimedia 99,* Orlando, Florida, USA. November 1999

Klas W, Neuhold E.J., Schrefl M., (1990), **Using an Object – Oriented Approach to Model Multimedia Data.** *Computer Communications,* Vol. 13, No. 4, pp 204 – 216

Kozuch, M., Wolf, W., Wolfe, A., (1996) **Client server architectures for nonlinear video services,** *Integration Issues in Large Commercial Media Delivery Systems,* Philadelphia, PA, USA ,pp. 17-28,

Kunze, J., (1999), **Encoding Dublin Core Metadata in HTML,** *Draft paper, Metadata Initiative,* http://www.ietf.org/rfc/rfc2731.txt

Lagoze, C., (1996), **The Warwick Framework, a container architecture for diverse sets of metadata,** *D-Lib Magazine,* July / August 1996, ISSN: 1082-9873

Legget, J.J., Schanse, J.L., Smith, J.B., and Fox, E.A., (1993) **Final report on the NSF workhshop on hyperbase Systems,** *Technical Report (TAMU-HRL 93-002),* Texas A&M University, USA

Letsche, T.A., Berry, M.W., (1997) **Large scale information retrieval with latent semantic indexing,** *Information Sciences,* 100(1-4), 105 – 137.

Lee, S.Y., Kao, H.M., (1993) T.D.C. **Video Indexing – An approach based on moving object and track,** in proceedings of *SPIE Storage and Retrieval for image and video databases,* vol. 1908, 2-3 Feb. 1993, pp. 25 – 36, San Jose CA, USA.

Lee, T.B., Cailliau, R., Groff, J.F., A. Nielsen, H.F. and Secret A. (1994) **The World-Wide Web**, *Communications of the ACM*, August 1994, Vol. 37, No. 8, pp. 76-82

Lee, T.B., Cailliau, R., Groff, J.F., and Pollermann, B. (1990) **World-Wide Web: The Information Universe**, *Internal Report*, CERN 1211, Geneva, Switzerland.

Levaco, R. (1974), **Kuleshov on film: writings by Lev Kuleshov**. Berkeley, CA, University of California Press, 1974.

Lewis, P.H., Davis, H.C., Griffiths, S.R., Hall, W., Wilkins, R.J., (1996), **Media based navigation with generic links,** in *proceedings of Hypertext 96*, ACM, ACM Press USA.

Liu, J. C., Du, D., Shim, S., Hsieh, J., Lin, M., (1999) **Design and evaluation of a generic software architecture for on-demand video servers,** *IEEE Transactions on Knowledge and Data Engineering [IEEE Trans Knowl Data Eng]*, vol. 11, no. 3, pp. 406-424, 1999

Liusheng, H., Lee, J.C., Lee, Q., Xiong, W., (1996), **An Experimental Video Database Management System Based on Advanced Object Oriented Techniques.** *ISAT/SPIE symposium on storage and retrieval for image and video databases*, Feb 1996, San Jose, USA.

Lowe, D., Hall, W., (1999) **Hypermedia and the web, an engineering approach**¸ *John Wiley and Sons publishing Co*, 1999

Mandler, J.M., (1984), **Stories, Scripts and Scenes: Aspects of Schema Theory,** *Lawrence erlbaum associates, publishers,* London, 1984.

Merlino, A., (1997) **Broad cast news navigation system using story segmentation,** *ACM Multimedia 97*. ACM Press USA.

Miller, G.A., Beckwith, R., Fellbaum, C., Gross, D., Miller, K (1993) **Introduction to WordNet: An on-line lexical database**, Revised edition August 1993. (http://www.cogsci.princeton.edu/~wn)

Mills, M., Cohen, J., and Wong, Y.Y., (1992) **A Magnifier Tool for Video Data,** in *Striking a balance: CHI' 92 conference proceedings,* 3 – 7 May, 1992, Monterey, CA, USA.

Nagasaka, A., Tanaka, Y., (1992) **Automatic Video Indexing and Full-Video Search for Object Appearences,** *Visual Database Systems,* Vol. II, pp. 113 – 127, Elsevier, Amsterdam, The Netherlands.

Nack, F., and Parkes, A., **The Application of Video Semantics and Theme Representation in Automated Video Editing**, *Multimedia Tools and Applications,* Vol.4, No.1, pp 57 – 83, January 1997.

Nielson, J., (1993) **Hypertext and Hypermedia,** Second Edition, USA Academic Press

Nielson, J., (!995) **Multimedia and Hypertext:** The Internet and Beyond. USA Ac Press

Oomato, E., and Tanaka, K., (1993) **OVID, Design and Implementation of a Video Object Database System,** *IEEE Transactions on Knowledge and Data Engineering,* Vol. 5, No.4, pp 629 – 643, August 1993.

Otsuji, K., Tonomura, Y. and Ohba, Y. (1991) **Video browsing using brightness data.** In proceedings *SPIE Visual communications and image processing ' 91: Image Processing,* Vol. 1606, pp. 980-989

Pentland, A., Picard, R.W., Haase, K, Davenport, G., (1995), **Video and image semantics: Advance tools for telecommunications**. Technical Report 283, MIT, Media labs Perceptual Computing Section, 1994.

Picard, R.W., (1995) **Digital libraries: Meeting place for high level and low level vision,** in proceedings, *Asian Conference on Computer Vision,* December 1995, number 1, pp. 1-5, Singapore.

Picard, R.W., (1996) **Society of models for video and image libraries,** *IBM Systems Journal [IBM SYST J]*, vol. 35, no. 3-4, pp. 292-312, IBM, Armonk, NY, (USA)

Powel T.A.,(1998) *HTML : The complete Reference.* USA: Osborne, Mc Graw Hill Publising

Pua P.K., (1993), **Prototyping the VISION Digital Video Library System**, *Maters of Science Thesis* submitted to the Department of Electrical Engineering & Computer Science and the Faculty of Graduate School of the University of Kansas, USA.

Real Works Inc (1998) **Real System G2 Production Guide,** Real System G2, Release 7, operation Manual. http://www.realnetworks.com

Real Works Inc (1999) **Authoring Streaming Media Presentations for Real System G2,** company white papers.
http://www.realnetworks.com/devzone/documentation/wp_authoring.html

Rowe, L.A., Boreczky, J. S., Eads, C.A. (1994) **Indices for User Access to Large Video Database,** in proceedings of *SPIE Conference on Storage and Retrieval for Image and Video Database II,* Vol. 2185, pp 150 – 161, Feb. 1994, SPIE, USA.

Salam, S., (1996), **VidIO: A model for personalised video information management,** Ph.D. thesis, November 1996, Multimedia Research Group, University of Southampton.

Salam, S., Hall, W., (1997), **Design and implementation of an experimental video database system for supporting video retrieval from different perspectives,** *in proceedings of Storage and Retrieval for Image and Video Databases V*, volume 3022, 13-14 February, San Jose California, pp 324 - 339

Segall, R.G. (1990) **Learning Constellations: A multimedia ethnographic research environment using video technology for exploring children's thinking,** PhD thesis, media arts and sciences, MIT, August 1990

Simonnot, B. (1995) **A Cooperation model for video document retrieval,** in proceedings of *SPIE on Multimedia Computing and Networking 1995,* Vol. 2417, 6-8 Feb 1995, San Jose CA, USA.

Smith, T.G.A., and Davenport, G. (1992) **The Stratification System: A Design Environment for Random Access Video,** in proceedings of *3rd International workshop on*

Network and Operating System Support for Digital Audio and Video, Nov, 1992, pp. 250 – 261, New York, USA.

Smith, T.G.A. and Pincever, N.C. (1991) **Parsing movies in context**, in proceedings of *1991 Summer USENIX Conference*, Nashville, TN, pp. 157 – 168

Smoiler S., Zhang, H., (1994), **Content based video indexing and retrieval,** *IEEE Multimedia*, Summer 1994, pp 62 – 72

Southampton Yellow Pages, 2000 – 2001; British Telecommunications Plc.

Srinivasan, U., Gu, L., Tsui, K., Simpson-Young, W.G., (1997), **A Data Model to Support Content-based Search in Digital Video Libraries,** *The Australian Computer Journal,* Vol 29, No. 4, pp. 141-147.

Stevens, S., Christel, M., and Wactlar, H., (1994) **Informedia: Improving Access to Digital Video, I*nteractions***, Volume 1, 4, New York: ACM, October 1994, pp. 67-71, ACM Press USA.

Swanberg, D., Shu, C.F, and Jain, R., (1993) **Video Map and Video Sapce Icon : Tools for Anotomizing Video content,** in proceedings of *INTERCHI ' 93 : Bridges between worlds, 24 – 29* April, 1993, Amsterdam, The Netherlands.

Tansley, R.H., (2000), **The Multimedia Thesaurus: Adding a Semantic Layer to Multimedia Information,** *PhD thesis,* Department of Electronics and Computer Science, University of Southampton.

Tonomura, Y., Akutsu, A., Otsuji, K., and Sadakata, T., (1993) **Video Map and Video Space Icon: Tools for Anotomizing Video Content,** in *INTERCHI ' 93 conference proceedings: Bridges between worlds, 24* – 29 April, 1993, Amsterdam, The Netherlands.

Ueda, H., Miyatake, T., Yoshizawa, S. (1991) **IMPACT: An interactive natural motion picture dedicated multimedia authoring system.** In *INTERCHI ' 93 conference proceedings: Bridges between worlds,* Stacey Ashlund et. Al. (eds.), April 24-29, 1993, Amsterdam, The Netharlands, pp. 137-141, New York: ACM.

Wactlar H.D, (1999), **New Directions in Video Information Extraction and Summarisation,** in proceedings of *10th DELOS Workshop, June 24 – 25, 1999, Sanorini, Greece.*

Weiss, R., Duda A., Gifford, D.K., (1995), **Composition and search with a Video Algebra,** *IEEE Multimedia*, Spring 1995, pp 13-25. IEEE press USA.

Weiss, R., Duda, A., Gifford, D., (1994), **Content-based access to algebraic video,** in *proceedings of IEEE International Conference Multimedia Computing and Systems*, 1994, Los Alamitos, CA, USA.

WG11/N1901, International Standard Organisation ISO/IEC JTC1 / SC29/WG11/ n1901**, Coding of Moving Pictures and Audio, Information Technology – coding of Audio visual Objects: system ISO/ IEC 14496-2** (Committee Draft), October 11, 1997

WG11/N2460 International Standard Organisation ISO/IEC JTC1 / SC29/WG11/n2460, **MPEG-7 Context and Objectives** (version 10), Atlantic City, October 1998.

WG11/N3545 International Standard Organisation ISO/IEC JTC1 / SC29/WG11/n3545, **Introduction to MPEG-7** (version 1), Beijing, July 2000

WG11/N3545 International Standard Organisation ISO/IEC JTC1 / SC29/WG11/ M4996, **Proposal for a new MPEG-7 Description Definition Language Grammar,** (J. Hunter), October, 1999

Wiil, U.K., (1997) **Open hypermedia systems, interoperability and standards,** *Special Issue on Open Hypermedia,* Journal of Digital Information ,1,2 December 1997.

Wilson, K.S. (1988) **PALENQUE : An interactive multimedia digital video interactive prototype for children**, in E.Soloway, D. Frye and S.B. Sheppard (eds.), *CHI'88: Proceedings of the fifth ACM-SIGCHI Conference on human factors in computing systems*, May 15-19, 1988, Washington D.C.

Woods, W. A. (1995) **Content-Based Information Access using Structured Conceptual Taxonomies**, slides presented at the Fourth International World Wide Web Conference, Boston, MA, December, 1995.

Woods, W. A. (1997) **Conceptual Indexing: A Better Way to Organize Knowledge,** SUN Technical Report, 1997.

XingTech Inc. (1998) **Multicast Requirements: Xing Stream works and the Internet Group Management Protocol,** company white papers. http://www.xingtech.com/files/docs/streamworks/igmp/index.html

Xiong, W., Lee, C.M., Ip, C.M., (1995), **Net Comparison: A fast and effective method for classifying Image Sequence.** *ISAT/SPIE symposium on storage and retrieval for image and video databases*, Feb 1995, San Jose, USA.

Yeung, M.M., Yoe. B.L., Wolf, W., Lieu, B. (1995) **Video Browsing using Clustering and Scene Transitions on Compressed Sequences,** in proceedings, *SPIE on Multimedia Computing and Networking 1995,* Vol. 2417, 6-8 Feb 1995, San Jose CA, USA.

Yeo, B.L., Yeung, M.M., (1997), **Retrieving and visualizing Video,** *ACM Communications*, vol. 40, December 1997, pp 43 – 52, ACM Press USA.

Zhang H.J., Wu, J.H., Low C.Y., Smoiler, S.W. (1995), **A video parsing, indexing and retrieval system**, in proceedings of *ACM Multimedia ' 95,* 7 – 9 November, 1995, San Francisco, CA, USA.

Zhang H.J., Smoliar, S.W., (1995) **Content based video browsing tools,** in proceedings of *SPIE on Multimedia Computing and Networking 1995,* vol 2417, pp. 389 – 398, Feb 6 –8, 1995, San Jose, CA, USA.