

---

# SEARCHING AND BROWSING A SHARED VIDEO DATABASE

Rune Hjelsvold, Roger Midtstraum,  
Olav Sandstå

*Department of Computer Systems and Telematics  
The Norwegian Institute of Technology  
Trondheim, Norway*

In this chapter we will discuss issues related to searching and browsing a shared video database. The discussion will be founded on a review of characteristics of video information and video database applications, and we will discuss requirements to be fulfilled by video databases in shared environments. The issues that will be discussed include video database architectures, video algebra operations, video querying, and video browsing. We have developed an experimental environment called *VideoSTAR (Video Storage And Retrieval)* that will be used to illustrate the issues being discussed.

## 1 INTRODUCTION

There is a strong technology push behind the development of video databases, but there is also a real end-user demand for video database technology. Such end-user demands should be taken into considerations when developing video database systems. There is a variety of users and types of applications and this makes it difficult to put up one set of requirements that can be unanimously agreed upon. Thus, if video database functionality for searching and browsing is based on requirements from one specific application only - e.g., video-on-demand - one may end up with a video database system that does not fulfill the needs for other users.

Television archives can serve as an example where video information sharing is important. The primary purpose of a television archive is helping reporters and directors in finding pieces of video with a specific image content - e.g., pieces of video showing a specific person, object, or location. The television archive is

also used by reporters for programme research - i.e., when they are searching for background information that might shed light on a specific issue.

A television archive may also have users outside the television company. Historians and media researchers have started going to the television archives because these archives are a rich source of documentary information from the last half of the 20th century. The television company might also make use of digital video technology to overcome some of the limitations with the broadcasting concept: Television programmes might be stored on video-on-demand servers so that the viewers can decide *when* to watch a given programme, or more interactive services might be provided - e.g., news-on-demand.

## 2 VIDEO INFORMATION IN A SHARED ENVIRONMENT

Databases are especially developed for managing data in shared environments. By shared environment we mean an environment where data are to be shared between different users. Also, in a shared environment different computer tools - e.g., planning tools, editors, and query tools - may access a common data repository for sharing and exchanging data. To develop video databases for shared environments, therefore, one should consider characteristics of video information and video database applications.

### 2.1 Video Information

*Video information* is not a very stringently defined term. To make the following discussion on video information sharing clearer, we will define our interpretation more precisely. We will use *video document* as the term denoting video compositions, such as movies and television programmes. A video document is often composed of still images, audio, and other types of media in addition to video data. We will use *video information* as a collective term that includes both *media data* and *meta-data* to be associated with the media data. The types of data that may be present in a video database include:

- *Media Data*. This category includes audio and video data *as recorded*, audio and video data generated during editing (e.g., when effects such as wipes and dissolves are applied to video data), and other media data.

- *Media-specific Data.* Some meta-data is required to control playback and rendering of media data - e.g., video format, frame rate, and size of the video window.
- *Compositional Data.* A composition defines the relations between a video document and the media data that are used. Today, compositional data are often used to generate complete copies of the document on video tape. The resulting video plays a dual role: It can be considered as one contiguous audio and video segment, and, at the same time, it represents a possibly complex video document.
- *Bibliographic Data.* This category describes the video document as a whole and includes information such as title, date of issue, production team, and actors/reporters contributing in the video document.
- *Structural Data.* Video documents are often well-structured into a structure hierarchy in similar ways as books are organized into chapters, sections, and subsections.
- *Content Annotations.* Content annotations are textual descriptions of the sensory content in a video. These annotations are manually entered by users and serves as indexes to the content of a piece of video.
- *Content Feature Data.* This category includes features that are automatically extracted from video and audio data. Such features can be used instead of - or in addition to - content annotations to provide content-based retrieval.
- *Topic Annotations.* The topics that are presented or elucidated in a video document are determined by the contents of the individual pieces of audio and video used in the document *and* by the way the individual pieces are combined. Content annotations can be used to describe or classify the issues being raised in video documents.
- *Supplementary Information.* Content and topic annotations serve as indexes to the content and topics in video. The user may want to associate other types of information to a piece of audio/video - e.g., for making personal remarks.

One important question for the users of a video database is how to acquire meta-data. There are several approaches that can be taken. The most ambitious one, *feature extraction*, assumes that characteristic features can be automatically extracted from media data. A less ambitious approach, *in-production capturing*, assumes that the tools used for recording and/or video editing (semi)

Meta-data	Feature extraction	In-production	Post-production
Compositional	Cut-detection [10, 30]	Yes	Yes
Bibliographic	No	Some	Yes
Structural	Simple structures	Yes [12]	Yes
Sensory content	Limited domains	Little	Yes
Topic content	Hardly	Some	Yes
Supplementary	No	Yes	Yes

Table 1 Usability of Different Meta-data Capturing Methods

automatically collects meta-data - e.g., when recording tools automatically register date and time for recording, or when compositional data are generated by authoring tools. The approach that is chosen by most users today (though this is a time-consuming task), *post-production capturing*, assumes that users - e.g., librarians - manually enters meta-data when the production process is completed. Table 1 indicates how the different methods apply to different types of meta-data.

## 2.2 A Variety of Video Applications

A shared video database should support different types of applications in sharing of video information. In this section we will take a closer look at some classes of applications that might use video databases. The purpose of this presentation is to illustrate that a shared video database must give consideration to the needs of more than one specific application. For this purpose, we have chosen five different classes of applications:

- *Video-On-Demand services.* VOD services allow users to search for videos and movies stored on a digital video server [6, 19, 20, 27]. A typical VOD service is aimed at offering the user flexibility in choosing movies from a large set of available titles. The entire movie is presumed to be the unit of interest and, thus, selection is mainly based on bibliographic data such as title, genre, or director. When a video has been selected, it is assumed that the movie will be viewed from beginning to end with little user interaction: Users may start, stop, pause, or playback a part of the video, but they do not need functionality to skip parts of the video, change the sequence of scenes, or search for parts of the video having a specific content.

- *Interactive video applications.* Most videos and movies on VOD servers can be classified as *linear* [21]. As opposite to linear video, interactive video assumes that the users may access scenes in any order. An interactive *News-on-demand* service [24] is an example of an interactive video application. The users of a news-on-demand service may want to select news items based on topic - and possibly on image contents - and may request the ability to decide in which sequence the news items should be played. Structural and compositional data are a prerequisite for offering this kind of functionality.
- *Shot-stock applications* are applications that access information related to recorded audio and video. A *shot* is a film theoretical term for a piece of film or video that has been recorded continuously. Shot-stock applications are especially used by film [33] or television directors and television reporters to retrieve video with a certain image content - e.g., to find video shots showing a specific person or object.
- *Programme research applications* are applications that assist users - e.g., television reporters, mass media researchers, or historians - in finding video documents related to a specific, but not necessarily precisely defined, topic. "People's Century" is considered to be the biggest and most ambitious historical documentation series that BBC has ever undertaken [35]. During the work on the series BBC has gained substantial experience with this kind of research: Topic data are needed to retrieve a collection of television programmes that can serve as the starting point for the research. A comprehensive breakdown of the individual programmes, identifying each shot *and* its source, is needed because BBC has to acquire permission from owners and authors in order to reuse archival material. In such a large project, it is also necessary to add personal remarks to interesting shots in the archive for use in later stages of the process [35].
- *Video documentation applications* support users in using video to document aspects of the real-world. This includes, for instance, anthropology [29], hand craft documentation [22], and user requirements analysis [18]. The main difference between this class of application and the others is the strong emphasis on supplementary data; users want to attach their detailed remarks and analysis to the video data.

Table 2 summarizes the importance of different types of meta-data for these classes of applications.

Meta-data	VOD	I-video	Shot-stock	Research	Docum.
Compositional	No	High	No	High	Low
Bibliographic	High	High	Medium	High	Medium
Structural	Low	High	No	High	Med./high
Sensory content	No	Medium	High	High	High
Topic content	Low	High	Medium	High	High
Supplementary	No	Medium	Low	Medium	High

Table 2 Importance of Meta-data in Different Applications

## 2.3 Temporal Aspects of Video Information

Video data are inherently temporal in the sense that the content of a video display is dynamically changing during playback. Technically, video data can be considered as a stream of images (called *frames*) displayed to the user at a constant frame rate. Video information is, however, more complex than just a stream of images: The way a user interprets individual pieces of video is influenced by the surrounding parts (i.e., the context into which the piece of video is used). This was evidently shown in a series of experiments by Soviet film makers in the last half of the 1920s [7].

These film makers noticed that temporal composition (in film theory called *montage*) was at least as important as spatial composition - i.e., how the scene space is organized (in film theory called *mise en scène*):

*Pudovkin offers a sort of formula: Film creation equals (1) what is shown in the shots, (2) the order in which they appear, and (3) how long each is held on the screen. [7]*

Temporal composition gives the director a means for creating contexts that give the user the ability to correctly interpret the contents of a video document.

Video documents and audio/video recordings define their own, discrete time systems because of their temporal characteristics. A specific part of a video defines a *temporal interval* within the time system of this video. This means that audio/video data and related meta-data, in contrast to traditional data types, may have temporal relationships to each other. During composition, the audio/video recordings are bound to the time system of the video document.

Thus, video document and video recording time systems may be related to each other.

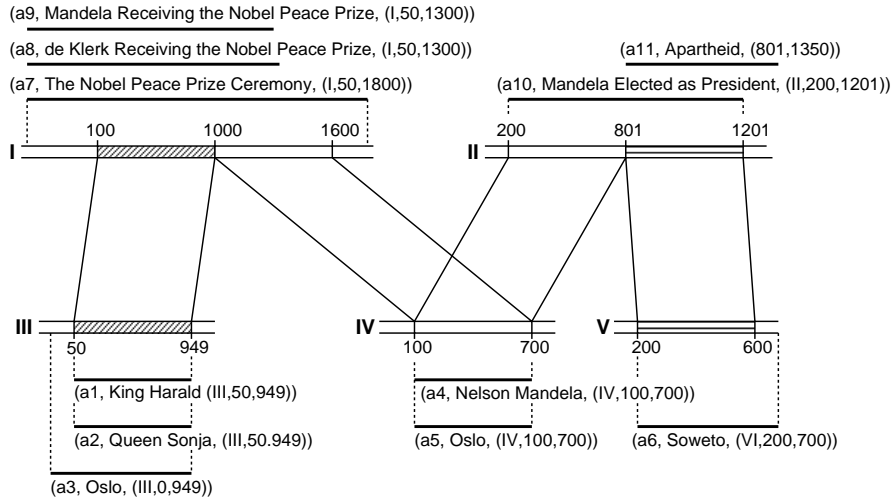
## 2.4 Sharing Video Information

In a shared video environment, the same pieces of media data may be part of several different video documents. Different documents define different contexts and, since the interpretation of video data is strongly dependent on its context, context handling is an important feature of a shared video database. A key question for context handling is how meta-data, especially structural and content data, can be shared in a consistent way when media data are shared or parts of video documents are reused in other documents.

Some researchers [26, 34] have proposed to support reuse of video information by allowing parts of a video document to be a component of other video documents and, thus, introducing an unconstrained hierarchy of video documents. We recommend to have only two levels in a video database: media recordings and video documents. This allows us to organize meta-data into three different classes with different relevance for a given document:

- The *primary context* contains meta-data that are specifically valid for the given document. If a part of the video document is reused in another video document, primary context meta-data may no longer be valid.
- The *basic context* for a piece of video contains meta-data that are valid independent of which primary context it is being used in - e.g., the name of a person shown on the video and the time and location for the recording.
- The *secondary context* for a given video document exists if the video document uses media data that are also being used in other video documents. These other document's primary contexts comprise the secondary context.

Figure 1 illustrates these concepts using a sample database consisting of two video documents (*I* and *II*) and three video recordings (*III*, *IV*, and *V*). The two documents share an interval from recording *IV*. The database also contains six image content annotations (*a1* through *a6*) and five topic content annotations (*a7* through *a11*). (The annotations have the format (*AnnotationID*, *Title*, *StreamInterval*) where *StreamInterval* is an interval from a video document or video recording having the format (*StreamID*, *StartTime*, *EndTime*).



**Figure 1** Example Video Data and Meta-data

Take the stream interval  $(I, 1000, 1600)$  as an example. The *primary context* for this piece of video consists of video document  $I$  and three annotations ( $a7$  through  $a9$ ) which say that the topic of the news item is the Nobel Peace Prize Ceremony for Mandela and de Klerk. The *basic context* consists of video recording  $IV$  and two sensory content annotations  $a4$  and  $a5$  which say that this video was recorded in Oslo, Norway, and that it shows Mr. Nelson Mandela. The *secondary context* for the stream interval, in the current state of the database, consists of video document  $II$  and one topic annotation,  $a10$ , associated with it. It is important to note that the secondary context is a virtual concept which identifies a collection of (other) primary contexts sharing a piece of video.

The reason we have defined the basic, primary and secondary contexts is to provide better control of sharing and visibility of the descriptive data. Annotations related to an audio/video recording that will be valid in any context, are represented in the basic context and can be "seen" and shared by all video documents using the recording. Annotations that are specific to a video document are represented in the primary context of the document and will not be intermixed with annotations specific to other documents, even when the two documents share some piece of video. When needed, the secondary context can be searched to find other video documents using the same piece of video

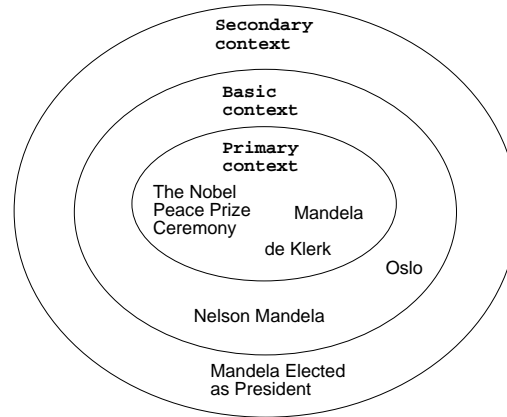


Figure 2 Annotations Grouped by Context

and the annotations defined in this context. Figure 2 illustrates the relation between different contexts for the example discussed above.

## 2.5 Querying and Browsing

In this paper we will focus on querying and browsing features that allow the user to retrieve media or meta-data from a video database. For querying, we will focus on how to retrieve pieces of video from video documents or from recordings by specifying meta-data properties. For browsing, we will discuss operations that allow the user to either browse meta-data or to browse the media data by using structural data. From the discussion in the previous subsections, we conclude that a video database supporting video information sharing should include the following features:

- *Generic architecture/data model.* Users and applications that need to share video information must share a common interpretation and understanding of video and meta-data. A generic video database architecture, including a generic data model, would reduce the effort needed to obtain such a common view of video and meta-data. This is further discussed in Section 3.
- *Content indexing.* Content-based retrieval is an important task for most video databases. This can be done by using advanced feature extrac-

tion/matching tools or by providing tools and methods that can enhance manual indexing (see Section 5). Feature extraction/matching is a challenging research area but will not be further discussed in this chapter.

- *Temporal relations.* It should be possible for users to exploit the temporal nature of video information in video querying - e.g., by specifying temporal relationships between pieces of video.
- *Controlling scope of interest.* It should be possible for users to control the degree of sharing and, thereby, the scope of interest. Temporal and search scope operations are further discussed in Section 4 and their use in querying and browsing are discussed in Sections 6 and 7, respectively.

### 3 FRAMEWORKS FOR VIDEO DATABASES

As discussed in the Section 2, video databases might be rather complex with a number of different types of media and meta-data. To ease development of video database applications, one may develop video database frameworks that can hide some of this complexity from the applications.

#### 3.1 Architecture

The framework presented here, VideoSTAR, provides an overall architecture for storing and management of media *and* meta-data from which video database applications can be developed. Other researchers have proposed generic multimedia information system architectures [9] and multimedia processing architectures [3]. The three-level VideoSTAR architecture, which is shown in Figure 3, is richer than existing architectures in supporting video information management:

- *Specialized repositories.* Four specialized repositories are defined: *Stored Media Segment DB* stores uninterpreted media data together with media-specific and bibliographic data, *Video Document DB* stores compositional and bibliographic data, *Video Structure DB* stores structural data, while *Annotation DB* can be used for any type of content or supplementary data that links real world concepts to a specific piece of video.

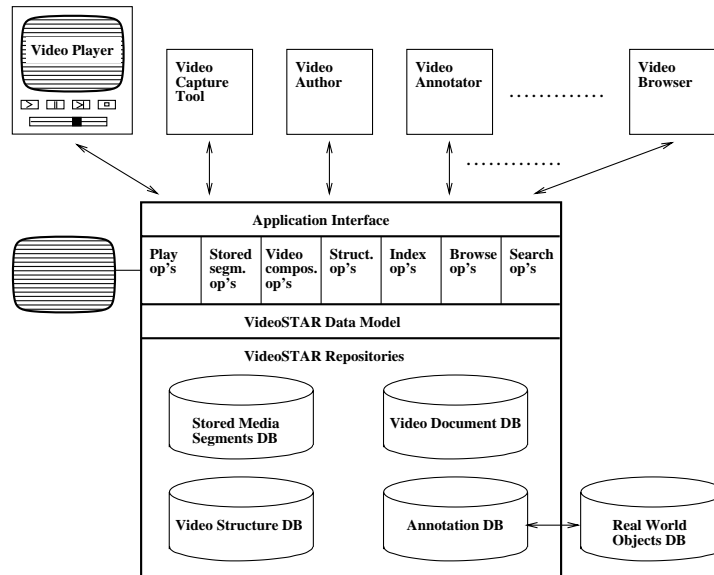


Figure 3 The VideoSTAR Architecture

- *Generic data model.* The framework includes a generic data model which is further discussed in Section 3.2
- *Video database API.* An application’s programming interface offers operations for managing the contents of the repositories, for allowing applications to control video replay, and for querying and browsing.

### 3.2 Data Models

If video information is to be shared by different applications and users, these applications and users have to share a common understanding of how the video data and meta-data are to be interpreted. The generic VideoSTAR data model is specially designed to represent relations between the contents of the four repositories. The data model is discussed in detail elsewhere [16] while Figure 4 highlights the dominant characteristics of the model.

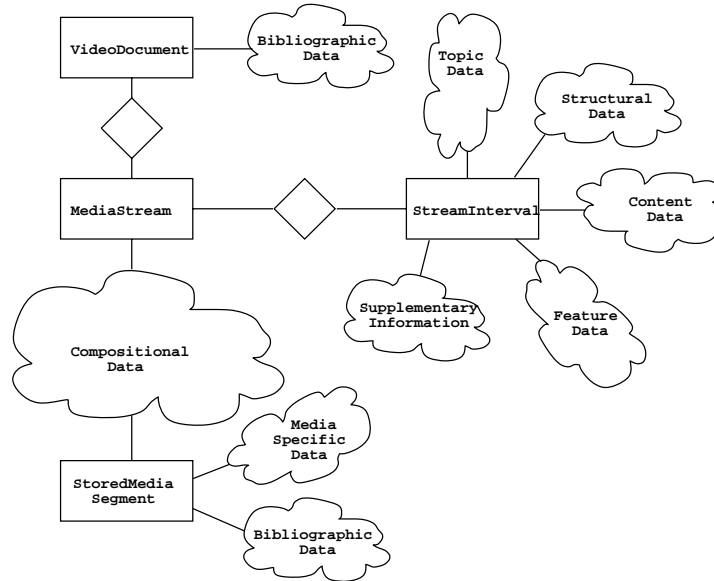


Figure 4 Overview of the Generic VideoSTAR Data Model

The model is a refined successor of earlier models [13, 15]. The core of the model is the **MediaStream**, which captures the most important characteristics of continuous media data like audio and video, and which provides the play operations offered in the API. Each video document is associated with a **MediaStream** that represents the document's video stream. Compositional data represents relations between video documents and **StoredMediaSegments** - i.e., media data - which, in turn, are specializations of **MediaStreams**. Bibliographic data are associated with video documents and media data.

A **StreamInterval** is an arbitrary (temporal) interval from a **MediaStream**. **StreamIntervals** is the means for relating meta-data to specific pieces of a video document or a media segment, as illustrated in the figure. The structure part of our model is inspired from film theory [25] and defines a structural hierarchy consisting of *shots*, *scenes*, *sequences*, and (possibly recursive) *compound units*.

We do not claim that the proposed model is ideal for every application and user. It is rather a kernel model that can be tailored to the needs of specific

application domains by giving structural components an application domain specific interpretation, by refining meta-data definitions, and by linking meta-data to a specific real-world model.

## 4 FUNDAMENTAL QUERYING AND BROWSING OPERATIONS

In [11] Hampapur, Jain and Weymouth write: "*The nature of indexing video is dependent on the problem of modeling video which in turn depends on the nature of the application of the database.*" Querying and browsing are different ways to make use of the results from the indexing efforts, and thus are equally dependent of the model of the video data.

Meta-data models are used in many parts of computer science, such as databases [32], information retrieval [28], knowledge representation [31], and image processing [8, 11]. Models from different domains tend to focus on different types of information and solve different aspects of the problem of modelling and retrieving information.

Considering the different types of data in a shared video database as defined in Section 2.1, database type models are most suited for modelling of *media data*, *media specific data*, *compositional data*, *bibliographic data* and *structural data*. Information retrieval models are most suitable for modelling of *content annotations* and *topic annotations*, image processing models are most suitable for *content feature data*, and knowledge representation models are most suitable for the representation of *supplementary information*.

When used in a video database system, browsing and querying of the different types of meta-models are done by the standard techniques used for that type of model. What is novel in video databases is that some of the descriptive data are related to pieces of video data – stream intervals in the VideoSTAR model – and that it may be necessary to consider both the temporal and the compositional aspects of the video information to provide the necessary functionality.

Take the situation in Figure 1 as an example and consider a media researcher studying the television news related to the last president election in South Africa. She could for instance want to investigate how video related to apartheid was used. One possible query could be to find video data related both to the election of Nelson Mandela and to apartheid. It is straightforward to find the

set of stream intervals related to Mandela and the set of stream intervals related to apartheid, but in order to combine these two sets into the wanted result, temporal operations have to be applied.

In the work with VideoSTAR we have developed a video query algebra [17] that allows us to formulate complex queries based on temporal relationships between stream intervals. This algebra allows the user to define the contexts to be searched. The algebra is defined over a special type of sets that are called *Mapped video object sets*. Mapped video object sets contain tuples (*ObjRef*, *StreamInt*) where *ObjRef* is the object identifier of one of the objects in a VideoSTAR repository (see Figure 3) while *StreamInt* is the stream interval that the object is mapped onto.

The video algebra operations will be briefly described in this section and their use in querying and will be further discussed in Section 6 and Section 7.

## 4.1 Temporal operations

In order to be able to express the necessary relations between sets of mapped video objects we have to extend the standard set operators. The new operators take into account the temporal nature of the mapped video objects.

**Normal Set Operations:** Normal set operations – i.e., intersection (*AND*), union (*OR*), and difference (*NOT*<sup>1</sup>) where two elements are defined to be identical if they refer to the same object over the same stream interval. Assuming that *A* and *B* are sets of person annotations, *A AND B* will contain elements from *A* where an element from *B* is referring the same person annotation over the same interval.

**Temporal Set Operations:** As noted by Clifford and Crocker [4], normal set-theoretic operations of union, intersection and difference produce counter-intuitive results when applied to temporal data. Therefore, temporal variants called *tAND*, *tOR*, and *tNOT* are defined. The elements of the result set contain the stream intervals resulting from intersecting, merging and subtracting stream intervals from the input sets as illustrated in Figure 5. Assuming that *A* is a set of annotations related to the election of Nelson Mandela as president and *B* is a set of annotations related to apartheid, *A tAND B* will have elements

---

<sup>1</sup>*NOT* is used to denote the difference between two sets – i.e.,  $A \text{ NOT } B \equiv A \text{ AND } (\text{NOT } B)$ .

identifying the stream intervals which can be associated with *both* Mr. Mandela and apartheid.

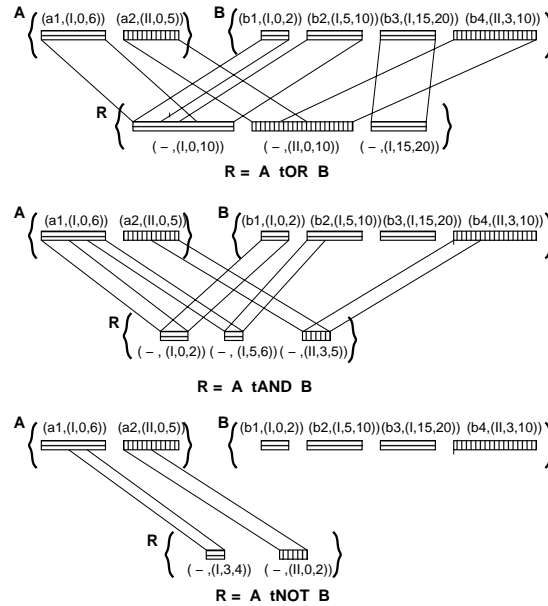


Figure 5 Example Showing the Interval Set Operators

**Filter Operations:** Allen has shown that there are 13 possible (distinct) relationships that can exist between two temporal intervals [1] – e.g., *before*, *overlaps* and *equals*<sup>2</sup>. Filter functions compare two stream intervals to check whether a given temporal relationship exists. We have defined the *tREDUCE* operator to take two input sets and one filter function as arguments. It returns the elements from the first input set that have the given relationship to at least one element from the second input set. Assuming that *A* is a set of scenes and *B* is a set of annotations related to Nelson Mandela, *tREDUCE*(*A*, *B*, *intersects*) will return the scenes from *A* that can be related to Nelson Mandela.

**Macro operations:** To ease the specification of two common tasks we have defined two (macro) operators – *ANNOT* and *STRUCT*. The *ANNOT* operator takes one input set and one annotation type parameter as arguments and retrieves all annotations of the given type that intersects any of the elements

<sup>2</sup>By combining these one may also define other operations such as *intersects*.

in the input set. The *STRUCT* operator takes one input set and one structure type parameter as arguments and retrieves all structural components of the given type that intersects any of the elements in the input set.

## 4.2 Compositional operations

In the example where a researcher wanted to find video related both to the election of Mr. Mandela as president and to apartheid, we could use a temporal *tAND* operator because the annotations were made in the same (primary) context. If the researcher instead had wanted to find video recordings from the Soweto area used in relation to the election of Mr. Mandela as president, a *tAND* operator would not have given the wanted result because the annotations are made in different contexts. In order to process such queries, which require temporal operations on annotations from (possibly) different contexts, one will have to apply compositional operations.

The compositional operations map objects from one time coordinate system onto another time coordinate system. A mapping operation does not affect the *ObjRef* part of a mapped video object, while the *Interval* part after the operation gives the stream interval onto which the object is mapped.

The *Decompose* operator which is illustrated in Figure 6 maps the objects in the input set onto the basic context. If an element in the input set is already related to a part of a basic context, the element will be copied without changes to the output set. If an element in the input set is related to a primary context, the element will be mapped onto the basic context(s) from which the corresponding stream interval is composed. The left part of figure 6 shows an example where the annotation *a10* made in the primary context of video document *II* is decomposed onto the basic contexts of stored media segments *IV* and *V*.

The *MapToComposition* operator maps objects in the opposite direction – i.e., from the time coordinate systems of a basic context to the related primary contexts. If an element in the input set is related to a primary context, the element will be copied without changes to the output set. If, on the other hand, the element is related to a basic context, the element will be mapped onto all primary contexts which uses parts of the basic context in its composition. The right part of figure 6 shows an example where the annotation *ab* made in the basic context *V* is mapped onto the (in this example single) primary context *II*.

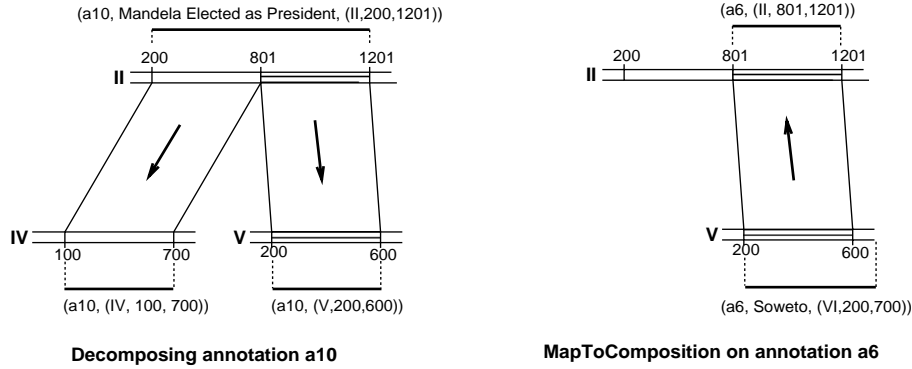


Figure 6 Compositional Operations

This operator can be used to find the result for the query on Mandela and Soweto. Assume that A contains the elements related to the election of Nelson Mandela,  $A = \{(a10, (II, 200, 1201))\}$ . Assume further that B contains the elements related to Soweto,  $B = \{(a6, (V, 200, 700))\}$ . The video related to both Soweto and Mandela can then be obtained by use of  $A \text{ tAND } MapToComposition(B)$ . This will give the result  $\{(-, (II, 801, 1201))\}$ .

## 5 AN EXPERIMENTAL VIDEO ARCHIVE ENVIRONMENT

The previous sections have given a general introduction to the main aspects of searching and browsing in a shared video database. In this and the following sections we will focus on how searching and browsing are supported in the VideoSTAR framework, and how the VideoSTAR applications use this functionality to allow users to search and browse the contents of the video database.

During the VideoSTAR project we have developed several experimental video archive tools. The main reasons for developing these video tools have been to create an experimental environment for working with digital video databases, and to test out the usability of the VideoSTAR framework. The tools have also

been used to get feedback from potential users of digital video databases – e.g., librarians working in television news archives.

The VideoSTAR tools are combined into an *integrated video tool environment* [14] which we present in this section. Tools for searching and browsing are presented in the following sections.

## 5.1 Tool Integration

The VideoSTAR integrated video tool environment consists of a video player, a tool manager, and tools for searching, browsing, and registration of meta-data.

A key feature for video archive tools is to provide interactive access to the video database including stored video documents and recordings. In providing such functionality, the video tools will need the capability to instruct the video player, for instance to load a specific video document and to jump to a specific point within the document. The VideoSTAR video player has the functionality that can be expected from a video player. In addition, it has a programming interface which allows the archive tools to control the playback by sending commands to the video player.

For some operations, the video tools do also need status information from the video player – e.g., to get the identity of the video that it is currently playing and the identity of the current frame – to allow the tools to operate synchronously with the video player. The VideoSTAR player can be instructed to report such status information regularly and the video tools can update their user interfaces according to the new state.

The integrated tool environment has been developed to make it possible for users to have different video archive tools interoperating simultaneously. The *Tool Manager* facilitates such synchronous interoperability: From the Tool Manager interface the user can choose which of the tools he/she needs for doing the work. The tool manager acts as a *broadcast message server* between the video tools and the video player. It broadcasts control commands from every video tool to every other active tool and distributes status information from the video player to every active tool.

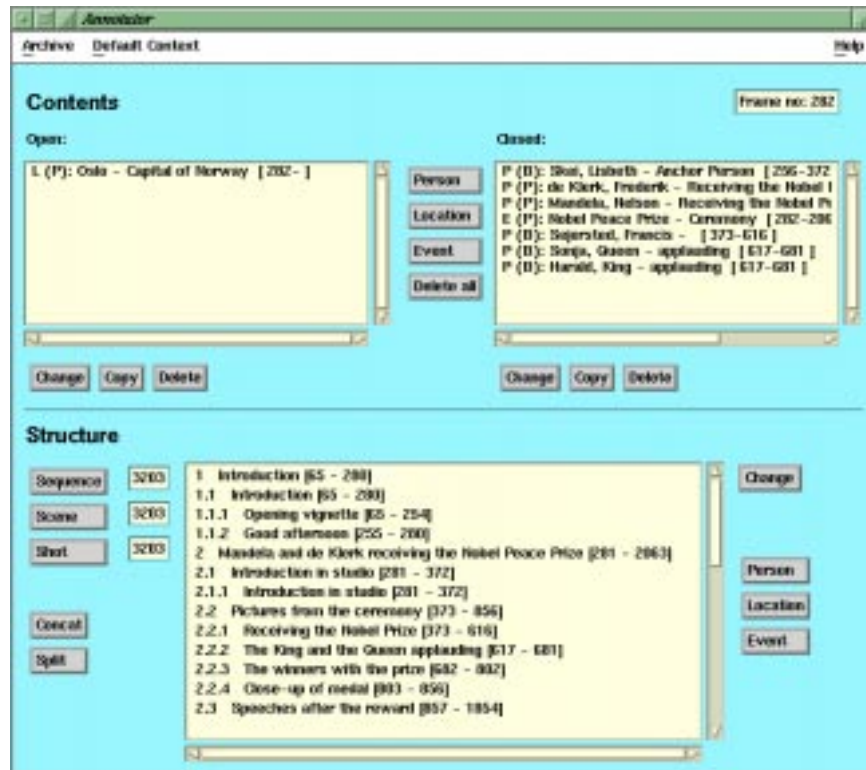


Figure 7 The Video Annotator

## 5.2 The Video Annotator

In Section 3 the VideoSTAR architecture was presented. In addition to storing the media-data, we must be able to register meta-data related to the video data. For this purpose we have developed a registration tool which is used to register both content annotations and structure information. The user interface is shown in Figure 7.

Registration of annotations is done by using the registration tool together with the video player. The user starts the registration process by playing the video document of interest – e.g., an evening news which among other things cover the Nobel Peace Prize Ceremony. Assume the user want to register information about the location of the Nobel Peace Prize Ceremony. When the user

has found the start of the part of the video related to the Nobel Peace Prize Ceremony, he/she pauses the video player and presses the “Location” button. The video player informs the annotator about the current frame number. This frame number<sup>3</sup> is used as the *start time* for the annotation, and the user can register Oslo as the location together with an optional explanatory text. The user then continues the playback until Oslo no longer is the location, pauses the player, positions the player on the last frame where Oslo is the location, and registers the *end time* of the location annotation.

Structure information is registered in much the same way as content annotations. The structure part of the registration tool is seen in the lower part of the figure.

Registration of meta-data is an important, but time consuming task. Our registration tool has mainly been developed to show how registration tools can benefit from having direct access to the video data and from having control over the video player. Different users need to have registration tools tailored to their particular way of doing registrations.

## 6 QUERYING

VideoSTAR contains a video query module that implements the algebra presented in Section 4. In this section we will describe how the algebra operations can be used to formulate video queries with a focus on context handling. The examples used in this section are taken from television news archives. The structure of a news report is represented as a ordered set of *sequences* where each sequence corresponds to one news item. The *primary* context of the news represents the issues being covered in the news items.

### 6.1 Query Processing

Query processing usually involves parsing a query, breaking it into basic (often algebra-based) operations, determining - and possibly optimizing - a query plan defining the sequence of basic operations, and performing this plan. The current version of VideoSTAR offers a pure video query algebra interface that allows us to test the usefulness of the algebra discussed in Section 4 without having to implement a complete query processor. Figure 8 gives an overview of the

---

<sup>3</sup>In our implementation we use frame numbers as a simplification of time codes.

four steps that an application has to go through when requesting VideoSTAR to process video queries.

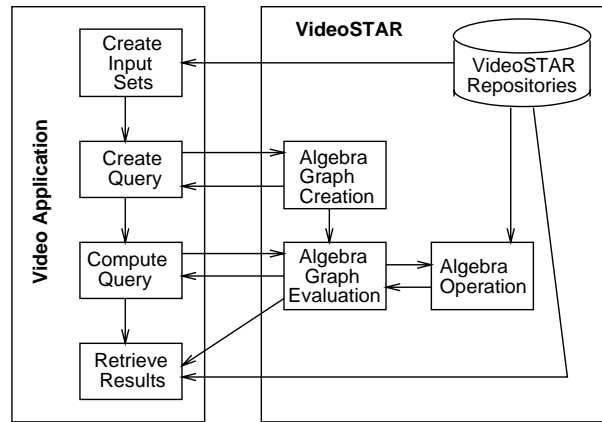


Figure 8 Query Processing Steps

**Step 1:** The application retrieves objects from the VideoSTAR repositories and inserts them into appropriate input sets for the query processing.

**Step 2:** The application instructs VideoSTAR to create the query graph with the corresponding operations. VideoSTAR will return a reference to each node that can be used by the application to access the node – e.g., for modifying the operation or for retrieving its result set.

**Step 3:** The application instructs VideoSTAR to perform the computation. The VideoSTAR *Algebra Operation* module computes the operations one-by-one in the sequence defined by the user application. The intermediate results are explicitly stored in each node and are used as input sets to other operations.

**Step 4:** The application accesses the appropriate node in the query graph and retrieves the corresponding result set.

The VideoSTAR repositories are used in the first step for selecting input sets, they are accessed by the *Algebra Operation* module to perform annotation, structure, and mapping operations, and they are accessed in the last step when the application retrieves the resulting objects themselves.

## 6.2 Experimental Video Query Tool

There are several ways to formulate queries – e.g., graphically such as QBIC [2], by using icons [5], or textually such as SQL [23]. The video query tool shown in Figure 9 offers the VideoSTAR query algebra directly to the user. It is not a query interface that can be offered most (at least infrequent) users but it provides us with an experimental environment for testing the video query algebra.

The user interface of the query tool is divided into three main parts. In the upper part the user can select from the set of content annotations contained in the database. In our implementation we have so far restricted these to be either persons, events, or locations. The middle part is used to create and compute the query, while the lower part is used to present the query results. Each item in the result list can be selected for replay by the video player.

When the user selects a person, event, or location, the query tool creates a mapped video object set which contains one mapped video object for each annotation related to the selected item. A tag is assigned to the set and the user can refer to the input set by using this tag.

The user creates a query step-by-step by selecting one of the operations offered in a list of commands. In addition to the operations defined in Section 4, the user may select an *INPUT* operation that connects input sets to the query. For convenience, most of the operations are also provided in versions that maps the input sets to the context of the operation before performing the operation – e.g., *tmAND* performs a temporal intersection operation after mapping the input set onto primary contexts. Each operation is tagged and these tags can be used to define the the result of one operation as an input argument to other operations.

When the query has been constructed, the user can select the *Compute* button to process the query. The sizes of all intermediate result sets are displayed next to each operation, and the items of the final result set are displayed in the result list.

## 6.3 Searching Primary Context

Assume that the user is searching for news items discussing the relations between Nelson Mandela and Frederik Willem de Klerk – i.e., the user is searching



Figure 9 The VideoSTAR Experimental Query Tool

for pieces of video from the primary context which can be associated with both persons. If we let  $P1$  be the set of annotations from the *primary* context related to Mr. Mandela and  $P2$  the set of annotations related to the Mr. de Klerk, the query can be expressed as:

```
R1 = INPUT P1           // Retrieves Mandela from primary context
R2 = INPUT P2           // Retrieves de Klerk from primary context
```

```
R3 = R1 tAND R2      // Finds intersecting stream interval
R4 = Sequences R3    // Finds the corresponding sequences
```

**R1** and **R2** contain the annotations from the primary context that are related to Mr. Mandela and Mr. de Klerk, respectively. As can be seen from Figure 9, **R1** and **R2** each contains four annotations. **R3** determines the four stream intervals which can be associated with both persons, while **R4**<sup>4</sup> retrieves the corresponding four news items (sequences).

## 6.4 Searching Basic Context

Assume that a news reporter is searching for one piece of video *showing* Mr. Mandela together with Mr. de Klerk. One solution would be to watch the news items resulting from the previous query to check whether such pieces exist. The user would then have to spend some time watching video to identify the relevant pieces. Even worse, the user may not get all pieces of video fulfilling the condition. For instance, assume that a news item related to economical reforms in South-Africa contains a shot from the Parliament showing Mr. Mandela and Mr. de Klerk together. Neither of the two are within the primary context of this news item, so this item would not be retrieved by the query.

The user should explicitly search the basic contexts because sensory information is registered in basic contexts. If **P3** the set of annotations from the *basic* context related to Mr. Mandela and **P4** is the set of annotations related to Mr. de Klerk, the query can be expressed as:

```
R5 = INPUT P3        // Retrieves Mandela from basic context
R6 = INPUT P4        // Retrieves de Klerk from basic context
R7 = R5 tAND R6      // Finds intersecting stream interval
```

As seen from Figure 9, there are four such pieces in the current database. If the user wants to retrieve the news items into which these stream intervals have been used, he/she should retrieve the corresponding news items (sequences):

```
R8 = Sequences R7    // Tries to find corresponding sequences
R9 = Sequences mR7    // Maps to primary context, finds sequences
```

As could be expected the set **R8** is empty since **R7** contains objects mapped to the basic context. By definition, basic context does not have structure and the

---

<sup>4</sup>Sequences **R3** is a more readable form of the operation  $STRUCT(R3, sequence)$

objects have to be mapped to the primary contexts – i.e., onto news documents – before news items can be retrieved. The **mR7** denotes that **R7** is mapped to the primary context before applying the **Sequences** operation. This makes **R9** return the two news items containing the stream intervals held by **R7**.

## 7 BROWSING AND PRESENTATION OF VIDEO INFORMATION

A user of a video database who are watching or working with a video document may wish to get more information about the video material than can actually be seen in the pictures and heard from the sound. The user may, for instance, want to know the names of the persons shown, the name of the locations where the video was recorded, and the time of recording. The user may also wish to get an overview of the video without having to watch the video, or to navigate within the video document without having to use fast forward or fast backward. To address this kind of needs, the database system should support browsing of structure information and presentation of content indexes related to a video document.

In this section we show how the structure and content meta-data can be used for browsing a video document and we present the VideoSTAR document browser.

### 7.1 Contents Browsing

In VideoSTAR, content browsing allows the user to retrieve annotations from the database intersecting a given interval of a video stream. By specifying what kind of annotations the user is interested in, he/she can restrict the browsing along two dimensions; by specifying *context(s)* for browsing and by specifying *conditions* on the annotations.

The different contexts a video document can be interpreted in, were introduced in Section 2.4. These contexts are the primary means for allowing the user to specify the scope of the browsing. VideoSTAR offers three different browsing functions: By browsing the *primary* context, the user gets all annotations related to the topic of a video document – i.e., all annotations related specifically to that document. By browsing the *basic* context, the user gets all annotations related to the stored media segments used in the document’s composition – e.g., annotations related to persons, objects or locations seen in the video recording.

By browsing the *secondary* context, the user gets annotations related to other video documents using some intersecting parts of the video document's stored media segments. The user can also specify that he/she wants information from two of the contexts or from all three contexts.

These browsing functions are implemented by using the fundamental operations defined in Section 4 [17]. To retrieve the annotations valid for the primary context the *ANNOT* operation is used. To retrieve the annotations from the basic (or secondary) context, the video stream interval has to be mapped to the basic (or secondary) contexts before the annotations can be retrieved.

The second dimension a user can restrict the information in, is to specify predicates on the content annotations. Such predicates can be that the user is only interested in person annotations, that he/she is not interested in the name of the photographer, or only want to get annotations registered by a certain user – e.g. the news archive department.

## 7.2 Structure Browsing

Assume that a user of a television news archive wants to browse through a collection of television news to have a quick impression of their contents. Fast forward replay has been the traditional way to do this kind of browsing. In a video database containing structural data, these data can be used for browsing. The purpose of structure browsing of a video document is to give the user an overview of the structure of the video document, and to let the user navigate within the document. The structure information can also be used to give the user a description of the context into which a piece of video has been used – e.g., when pieces of video have been retrieved in content-based queries.

Structure information differs from content annotations. In VideoSTAR, the structural components are organized as a hierarchy consisting of compound units, sequences, scenes and shots. To get structure information for a video document or an interval from a video document, the *STRUCT* operation defined in Section 4 is used. The VideoSTAR API provides operations that can be used to navigate in this hierarchical structure.

By using the structure information, a *table of contents* for the video document can be created. This can be used to give the user information about which part of the document that is currently being shown in the video player. It also provides an easy way for the user to move within the document. It allows, for

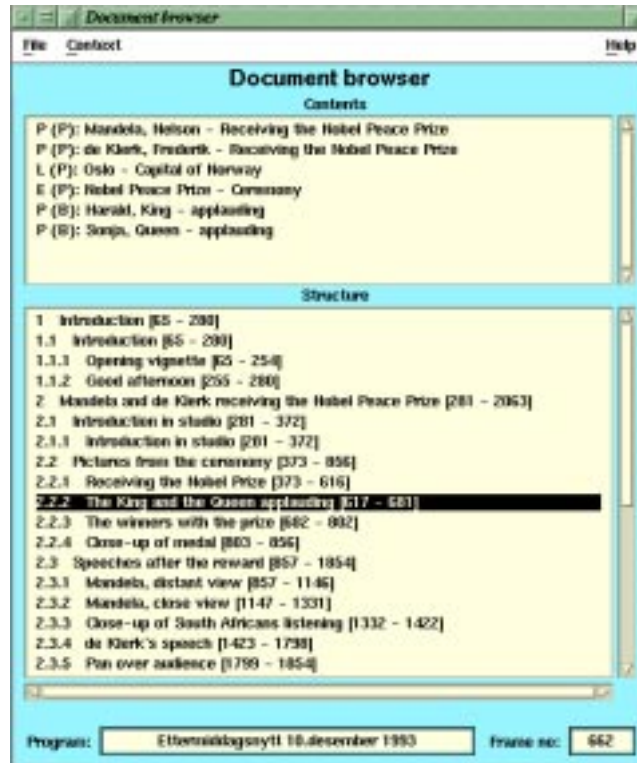


Figure 10 The Video Document Browser

instance, the user to directly jump to one given news item or to one specific scene within a news item.

### 7.3 The VideoSTAR Document Browser

The main task for the video document browser is to present to the user the meta-data related to the video as the video is played back. The temporal aspect of video information requires that the document browser is synchronized with the video playback, and there is a need for a close cooperation between the video player and the browser. The Tool Manager is responsible for keeping the browser synchronized with the video player.

The user interface of the browser is shown in Figure 10. As the figure shows, the browser is divided into two main parts, the upper part shows content annotations while the lower part shows structure information.

### *Content Annotations*

In our experimental database we have chosen to group content annotations into the categories *persons*, *locations*, and *events*. Figure 10 shows a snapshot of a video document containing the evening news from December 10th, 1993. From the figure we can see that four different people are related to the current part of this video document. The event (preceded by an E) is the Nobel Peace Prize Ceremony, and the location (preceded by an L) is Oslo. The browser distinguishes between annotations related to the topic in question (primary context – marked with “P”), and annotations describing what is actually seen on the video (basic context – marked with “B”).

The browse window is updated each time the browser gets a new frame number from the video player. This frame number is used for finding the content annotations which are valid for that frame. Content annotations that are no longer valid are removed from the window and new annotations are added.

### *Structure Information*

The structure window of the video document browser is organized in the same way as the table of contents in a book. The highlighted line shows which part of the video document that is currently being played. As the document is being played back, the highlighted line scrolls down the “table of contents”.

The structure window can also be used for interacting with the video player. By selecting a line in the structure window, the browser causes the video player to jump to this part of the video document and continue playing from that position. This makes it easy for the user to navigate within the video document.

## **8 EXPERIENCES AND DISCUSSION**

To gain experience with VideoSTAR environment and the video tools we have digitized and annotated 15 television news programs from The Norwegian Broadcasting Company (NRK), 15 news programs from TV 2 Norway, and

5 ethnographic films from the collections of The Norwegian Folk Museum. Besides our own experiences with this database, and casual demonstrations, we have arranged extensive demonstrations and discussions with archivists from the two national broadcasting companies, as well as with conservators from museums.

## 8.1 User Responses

The professional users have consistently given a very positive response to the philosophy behind the video tools. Of course, they would like a mini-world model tailored to their own needs and traditions, and they would like to change parts of the user interfaces, to bring them in accordance with personal preferences and practice. The Norwegian Folk Museum has, for instance, expressed the need for six different types of content annotations and supplementary information, in addition to the three types we have defined.

Still, altogether users acclaim to the power and flexibility of the underlying video data model which provides means for both structuring and free annotating of video material. They also appreciate the tool support of the registration process and are especially pleased with the direct connection to the video material during this process.

The query facility, which gives content based access to the video data, is seen as a very productive tool. It will, when used in real archives, do away with a lot of transport of video cassettes between magazines and users, and will significantly reduce the considerable amount of time spent on tedious sequential searching in today's archives.

More experiences related to the integrated video archive environment can be found in [14].

## 8.2 Context Handling

The clear definition of, and distinction between, primary and basic contexts supports sharing of meta-data, in two ways:

- When users classify meta-data as basic context data, they clearly indicate that this information is always relevant to the video. Since the basic con-

text related to a piece of video is common to all video documents using this piece, basic contexts preserve meta-data in well-structured ways. From the cooperation with television companies, we have learned that much sensory content data which have been registered in relation to a specific video document in their systems today - e.g., name of persons shown in the video - are not “inherited” to or visible from other documents reusing the video.

- By classifying some meta-data as part of the primary context of a piece of video, the user indicates that this information should be interpreted in relation to the specific video document, and may not be relevant for contexts. When shared video databases become common, we expect that it will be important to separate such context specific meta-data but in such ways that they are still available for users - e.g., media researchers - that need the possibility to search across several contexts. Librarians that have tested the VideoSTAR tools have responded positively to this possibility of determining the query scope.

By using the VideoSTAR tools, we have seen how useful the context handling is to establish a context for interpreting small fragments of video that are returned by a query. The following example may illustrate this: If we are using the query tool to retrieve all pieces of video showing the Norwegian King from the database, we will, as one of the resulting items, get a very short video fragment showing the King and Queen applauding. The user will have no idea what the occasion is, and why the King and Queen are applauding, by only watching the small fragment. If, however, the browser is active, the browser will provide the primary context for this shot as shown in Figure 10. The user will then immediately see that the royalties are applauding for Nelson Mandela and Frederik de Klerk, and that the occasion is the Nobel Peace Prize Ceremony.

No tools for video editing have been included into the video database environment yet, and, thus, we have not had the opportunity to experiment with secondary contexts. One of the the experiences made by BBC in the “People’s Century” project [35], was that specific video documents were the starting point for a comprehensive browsing of the related documents. The explicit support for browsing secondary contexts can play a significant role in simplifying this kind of research.

### 8.3 Querying Temporal Relationships

The most valuable effect of giving the user the opportunity to specify temporal relations in a video query, is that this reduces the size of the result and the effort needed to get an overview of it: First, because it allows the user to formulate more precise queries which will result in fewer, but more relevant, result items; second, because the items themselves may be smaller because irrelevant parts have been removed; third, small fragments of video that intersects can be merged into fewer and larger result items and, thus, it will be easier to examine the result. The price one has to pay is a more complex query language and a more complex query processor. To fully exploit temporal relationships, one also has to spend a lot of effort on indexing the video to ensure that meta-data have a proper accuracy.

Though less complex mechanisms may be sufficient for many applications - especially when sharing of meta-data is not required - we think that temporal operations are necessary to achieve acceptable accuracy when meta-data are shared. Assume, for instance, that each shot is described as one complete entity [33], and that one specific shot shows a pan over a crowd of people ending up with a zoom on Nelson Mandela. This shot would have Nelson Mandela as part of its basic context. Assume now that the first part of this shot which shows the crowd only, is used in another document. Since Nelson Mandela was part of the basic context, Nelson Mandela will implicitly be related to the sensory contents of the second document even though he is not shown on the part of the video that is actually used in this document. This will reduce the quality of the meta-data and result in highly undesirable effects that may confuse the user.

### Acknowledgements

We are especially grateful to researcher Stein Langørgen who has turned our ideas into running software and to the LAVA project (funded by the Norwegian Research Council, UNINETT and Telenor) for the financial support of our research. We would like to thank all master students at NTH who have contributed to our research on digital video applications and video database environments. We will also thank the Norwegian Broadcasting Corporation, TV2 Norway, and the Norwegian Folk Museum for participating in the work.

## REFERENCES

- [1] J.F. Allen. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, November 1983.
- [2] J. Ashley et al. Automatic and Semi-Automatic Image Retrieval Methods in QBIC. In *Proceedings of Storage and Retrieval for Image and Video Databases III - part of IS&T/SPIE's Symposium on Electronic Imaging: Science and Technology*, San Jose, CA, February 1995.
- [3] J.F.K. Buford. Architectures and Issues for Distributed Multimedia Systems. In J.F.K. Buford, editor, *Multimedia Systems*, chapter 3, pages 45–64. Addison-Wesley Publishing Company, Inc., 1994.
- [4] J. Clifford and A. Crocker. The Historical Relational Data Model (HRDM) Revisited. In A.U. Tansel et al., editors, *Temporal Databases: Theory, Design, and Implementation*, chapter 1. The Benjamin/Cummings Publishing Company, Inc., 1993.
- [5] M. Davis. Media Streams: An Iconic Language for Video Annotation. In *Proceedings of 1993 IEEE Symposium on Visual Languages*, Bergen, Norway, 1993.
- [6] D. Deloddere, W. Verbiest, and H. Verhille. Interactive Video On Demand. *IEEE Communications Magazine*, 32(5):82–88, May 1994.
- [7] J.C. Ellis. *A History of Film*. Prentice Hall, 3rd edition, 1990.
- [8] R.C. Gonzales and R.C. Woods. *Digital Image Processing*. Addison-Wesley Publishing Company, 1992.
- [9] W.I. Grosky. Multimedia Information Systems. *IEEE Multimedia*, Spring 1994.
- [10] A. Hampapur, R. Jain, and T. Weymouth. Digital Video Segmentation. In *Proceedings of ACM Multimedia '94*, pages 357–364, San Francisco, USA, October 1994.
- [11] A. Hampapur, R. Jain, and T. Weymouth. Indexing in Video Databases. In *Proceedings of the IS&T/SPIE Symposium on Electronic Imaging Science and Technology, Conference on Storage and Retrieval for Image and Video Databases III*, pages 292–306, San Jose, CA, February 1995.
- [12] L. Hardman, G. van Rossum, and D.C.A. Bulterman. Structured Multimedia Authoring. In *Proceedings of ACM Multimedia 93*, pages 283–290, Anaheim, CA, August 1993.

- [13] R. Hjelsvold. Video Information Contents and Architecture. In *Proceedings of the 4th International Conference on Extending Database Technology*, pages 259–272, Cambridge, UK, March 1994.
- [14] R. Hjelsvold, S. Langørgen, R. Midtstraum, and O. Sandstå. Integrated Video Archive Tools. In *Proceedings of the ACM Multimedia '95*, San Francisco, California, November 1995.
- [15] R. Hjelsvold and R. Midtstraum. Modelling and Querying Video Data. In *Proceedings of the 20th VLDB Conference*, pages 686–694, Santiago, Chile, September 1994.
- [16] R. Hjelsvold and R. Midtstraum. Databases for Video Information Sharing. In *Proceedings of the IS&T/SPIE Symposium on Electronic Imaging Science and Technology, Conference on Storage and Retrieval for Image and Video Databases III*, pages 268–279, San Jose, CA, February 1995.
- [17] R. Hjelsvold, R. Midtstraum, and O. Sandstå. A Temporal Foundation of Video Databases. In *Proceedings of the International Workshop on Temporal Databases*, Zürich, Switzerland, September 1995.
- [18] M.E. Hodges and R.M. Sasnett. *Multimedia Computing. Case Studies from MIT Project Athena*. Addison-Wesley Publishing Company, Inc., 1990.
- [19] T.D.C. Little et al. A Digital On-Demand Video Service Supporting Content-Based Queries. In *Proceedings of ACM Multimedia 93*, pages 427–436, Anaheim, USA, August 1993.
- [20] T.D.C. Little and D. Venkatesh. Prospects for Interactive Video-on-Demand. *IEEE Multimedia*, 1(3):14–24, Fall 1994.
- [21] C. Locatis, J. Charuhas, and R. Banvard. Hypervideo. *Educational Technology, Research and Development*, 38(2), 1990.
- [22] A.O. Martinussen. Håndverksregistrering på S-VHS. In *Fra idé til virkelighet*. Norske Kunst- og Kulturhistoriske Museer, 1994.
- [23] J. Melton and R. Simon. *Understanding the New SQL: A Complete Guide*. Morgan Kaufmann Publishers, 1993.
- [24] G. Miller, G. Baber, and M. Gilliland. News On-Demand for Multimedia Networks. In *Proceedings of ACM Multimedia 93*, pages 383–392, Anaheim, CA, August 1993.
- [25] J. Monaco. *How to Read a Film. The Art, Technology, Language, History and Theory of Film and Media*. Oxford University Press, 1981.

- [26] E. Oomoto and K. Tanaka. OVID: Design and Implementation of a Video-Object Database System. *IEEE Transactions on Knowledge and Data Engineering*, 5(4):629–643, 1993.
- [27] L.A. Rowe, J.S. Boreczky, and C.A. Eads. Indexes for User Access to Large Video Databases. In *Proceedings of the IS&T/SPIE Symposium on Electronic Imaging Science and Technology, Conference on Storage and Retrieval for Image and Video Databases II*, San Jose, CA, February 1994.
- [28] G. Salton. *Automatic Text Processing - The Transformation Analysis, and Retrieval of Information by Computer*. Addison-Wesley Publishing Company, 1988.
- [29] T.G.A. Smith. If You Could See What I Mean... Descriptions of Video in an Anthropologist's Notebook. Master's thesis, MIT, 1992.
- [30] S.W. Smoliar and H. Zhang. Content-Based Video Indexing and Retrieval. *IEEE Multimedia*, 1(2):62–72, Summer 1994.
- [31] J. Sowa, editor. *Principles of Semantic Networks*. Morgan Kaufmann, 1991.
- [32] D.C. Tsichitzis and F.H. Lochovsky. *Data Models*. Prentice-Hall, 1982.
- [33] J. Turner. Representing and accessing information in the shotstock database at the National Film Board of Canada. *The Canadian Journal of Information Science*, 15(4), December 1990.
- [34] R. Weiss, A. Duda, and D.K. Gifford. Composition and Search with a Video Algebra. *IEEE Multimedia*, 2(1):12–25, Spring 1995.
- [35] C. Whittaker. People's Century - Through the Archives. In *FIAT/IASA Internationaler Kongress 1994*, pages 123–129, Bogensee, Germany, September 1994. Battered Verlag, Baden.