Integrated Video Archive Tools

Rune Hjelsvold, Stein Langørgen, Roger Midtstraum, Olav Sandstå
Department of Computer Systems and Telematics
The Norwegian Institute of Technology
N-7034 Trondheim, Norway
{rhj, steinl, roger, olavsa}@idt.unit.no

ABSTRACT
In traditional video archives, video data are stored on analogue video tapes while meta-data, such as textual descriptions of the contents of the video tapes, are stored and handled digitally by computers. In a fully digital video archive, both video data and meta-data are managed by computers and, thus, more powerful tools can be developed.

In this paper, we discuss what kind of tools a digital video archive should offer its users, and we describe an experimental video archive system which consists of tools for playing, browsing, searching and indexing video information. All tools in the system are based on a generic database platform called VideoSTAR (Video STorage And Retrieval) and they share video and meta-data via the common database. The tools are managed by a video archive tool manager which provides mechanisms for communication and cooperation between different tools. The system has been demonstrated to professional archivists and librarians who have given positive response, and as the next step we will have the system tested in a real video archive environment.

KEYWORDS
Video databases, applications, content-based retrieval, browsing, digital libraries

1 INTRODUCTION

The most active research area for video databases today is Video-on-Demand (VOD). VOD services which will allow users to search for movies and videos stored on a digital video server [5, 13, 14, 18]. Today, many libraries and archives – e.g., television archives – have huge amounts of videos and/or films. These libraries and archives have a variety of users running different applications and asking for different services, not only VOD services. Thus, there is a need for an integrated video archive environment allowing the users to run different applications towards a common database of video data and meta-data.

In previous papers [9, 10, 12] we have presented database functionality for supporting video information sharing between different users and applications, and for querying and browsing the contents in a video archive/library. We have developed a number of video archive tools that illustrate the usability of our video database functionality. The purpose of this paper is to give the motivation for each of these experimental tools, describe their functionality, and discuss how these tools are integrated into a common environment.

The paper is organized as follows: In Section 2, we discuss some of the tools that an archive/library should offer. Section 3 describes the experimental video database framework that is used as the basis for these tools. In Section 4, the individual tools are presented and we describe how these tools may interoperate. Section 5 discusses the functionality of the video archive environment and presents some experiences from using these tools in practice, while Section 6 concludes the paper.

2 VIDEO ARCHIVES/LIBRARIES

Archives and libraries are service institutions that manages stored information and that enables users to access the stored information. To understand what services a digital video archive or library should offer its users, one should examine who these users are and what they are requesting from the archive. In a project called LAVA, we are working together with several Norwegian research institutes, the Norwegian Broadcasting Corporation (NRK), TV2 Norway, and the Norwegian Folk Museum to identify what information and services a digital television archive should offer [6]. In the following subsections, we review some of the functionality that should be offered by a digital video archive.

2.1 Searching

The primary service offered by a television archive is as-

Multimedia '95, San Francisco, CA USA
©1995 ACM 0-89791-751-0/95/11..$3.50

sisting reporters and directors in finding pieces of video from the archive. Two different types of video information are present in a television archive: The "raw" footage that is the outcome of video recording, and the final video "document" as it is broadcast. In most cases, users are looking for pieces of (raw) video that can be used to visualize a specific problem or issue; they are searching for video with a certain image or audio content - e.g., pieces of video showing the person Nelson Mandela. In other cases, reporters are doing more thorough programme research where the target may be old documentaries or news related to a specific topic - e.g., news items related to Nobel Peace Prize ceremonies. Thus, video archive search tools should be capable of performing queries on video document topics as well as image/audio contents and should allow the user determine the type of video to retrieve.

The context into which a piece of video belongs, has a strong influence on how this piece of video is interpreted. Thus, when a piece of video is retrieved from the archive, it is often necessary to retrieve parts of its context to interpret it. For instance, assume that a query on the person Nelson Mandela returns a piece of video showing Nelson Mandela applauding in a church. Before a reporter would use this piece of video in a television programme, he/she would probably want to know what event Nelson Mandela participated in and, especially, why was he applauding. The surrounding pieces of video shows the performing gospel choir. Video archive search tools should support users in retrieving such surrounding pieces of video.

2.2 Structure Browsing

One of the key problems with analogue video is the sequential access. In the LAVA project, we have experienced that directors sometimes use more than half of the editing time loading/unloading video tapes and scanning sequentially through the video tapes to find the interesting pieces. If the archive could offer tools which give direct access to individual sequences, scenes, or shots[1], the time could be spent more efficiently on inherent problems of editing. The structure does also represent an important context for a given piece of video within a video document.

2.3 Contents Browsing

A piece of video may show a person or an object without giving their names and/or specific descriptions. Often, this information is known by the production team during recording and may be stored as descriptions in the archive. The archive could store this meta-data together with the video data, and the information could either be entered during production, or it could be registered afterwards. A reporter using the archive for background research will usually be interested in having as much content information as possible about the video that he/she is viewing.

[1] These concepts are defined in [8].

2.4 Video Annotation

The librarians try to classify, describe, and annotate the contents of video documents and recordings to make searching and browsing possible in an efficient manner. Because of the lack of good tools, these descriptions are usually coarse grained and describe long pieces of video – e.g., complete news items. As a result, the user can only retrieve these long pieces of video even when he/she is looking for smaller pieces.

3 A PLATFORM FOR INTEGRATING VIDEO TOOLS

Other researchers have addressed some of the problems and issues discussed in the previous section. None of these have yet described integrated solutions for video archives which support a range of users and applications.

Researchers at MIT have developed ideas and tools for supporting video editing (e.g., Algebraic Video System [23]) and annotation (e.g., EVA [15] and Stratosphere [20]). From a general video archive point of view, the problems with these tools are the lack of support for managing video document structures. In addition, the proposed annotation tools are based on unformatted, free text descriptions of the contents. A digital video archive serving different categories of users should offer a more structured way of describing video contents.

Some researchers have tried to develop tools for automatic segmentation of video material [7, 21] – i.e., identifying cuts in a video document. These tools are interesting preprocessing tools in a video archive for segmenting video documents, especially when the edit decision list is not explicitly available.

Some tools have also been proposed for temporal browsing of video [1, 4, 21]. These tools are best applicable when the structure of the video follow a stringent, predefined syntax; i.e., television news. The disadvantage with these tools is that, when applied to pieces of video with complex or vague structure, they do not manage to group frames and shots into semantically meaningful entities such as scenes and sequences.

The Virtual Video Browser [13], the Video Database Browser [18], and OVID [16] are experimental video database systems that provide content-based querying. Neither of these systems offers tools for structure and contents browsing in the way described in Section 2, and efficient annotation tools have not been proposed.

Our work is aimed at developing a more complete video archive environment including a video database frame-

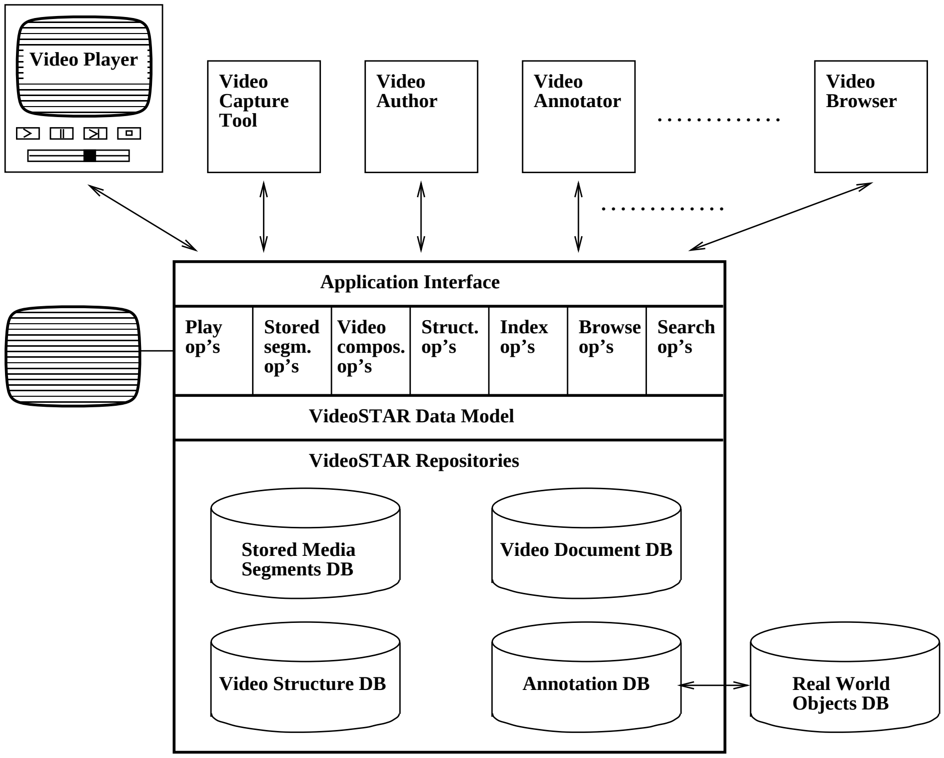

Figure 1: The VideoSTAR Architecture

work and video archive tools. The video database framework is called VideoSTAR (Video STorage And Retrieval) [10, 12] and its elements are (see Figure 1):

- Database repositories for storing video documents and video footage (recordings). In addition, two different types of meta-data are related to video data: document structures and content indexes/annotations.
- A generic data model which provides mechanisms for representing the contents of the four repositories and the relations between entities in these repositories.
- A video window which can be used by an application to play back video documents and recordings.
- A video API that provides a set of operations for creating a video window, loading a video document into the window, access any frame directly, and to control playing (forward and backward playback at different speeds). This API provides a higher level interface than the one proposed by Schnorf [19].
- An API for searching, browsing, accessing, and manipulating the contents of the four repositories.

The main reason for developing this framework is to provide a platform for development of video applications that share video information. To be able to share video information in a satisfactory manner, applications and users need to share a common view of video and meta-data. The generic data model provides such a common view. Video data have special requirements related to the data formats, delivery, and synchronization issues. Since VideoSTAR provides the video window abstraction, applications don't need to know how to tackle these problems and, still, be able to control how the video is played back. The purpose of the API for managing the contents of the repositories, is to encourage application developers to use the framework by providing a rich interface that hides some of the details of a video database. This makes it easier to develop video database applications.

4 VIDEO TOOLS

Section 2 gave an introduction to digital video archives. A video archive consists of video data and meta-data related to these. In this section, we will present the experimental video tools we have developed for working with a digital video archive.

To be able to display video on a computer screen we have implemented a video player. This gives us the basic functionality to play back digital video. A video player does not, however, provide a means for managing the meta-data we want to relate to the video. To be able to do so, we have developed several experimental tools that interact with the video player and give the user access to the meta-data.

We have identified three basic operations a user may want to perform against the meta-data in a video database [10]. The first operation is browsing of the meta-data related to a particular video document. As a second operation, the user might want to search after video documents covering a certain topic. The third operation is the need to be able to register meta-data about what is shown or heard in the video.

First we give an introduction to the video player, then we present the three main tools we have developed. Each of these performs one of the main operations presented above. After presenting the tools, we show how these are integrated and made to interoperate with each other and the video player to become an integrated environment for a video archive.

In addition to these tools, we have also been studying how video editing tools can be integrated into the environment [3], but no such tool has been integrated into the environment yet.

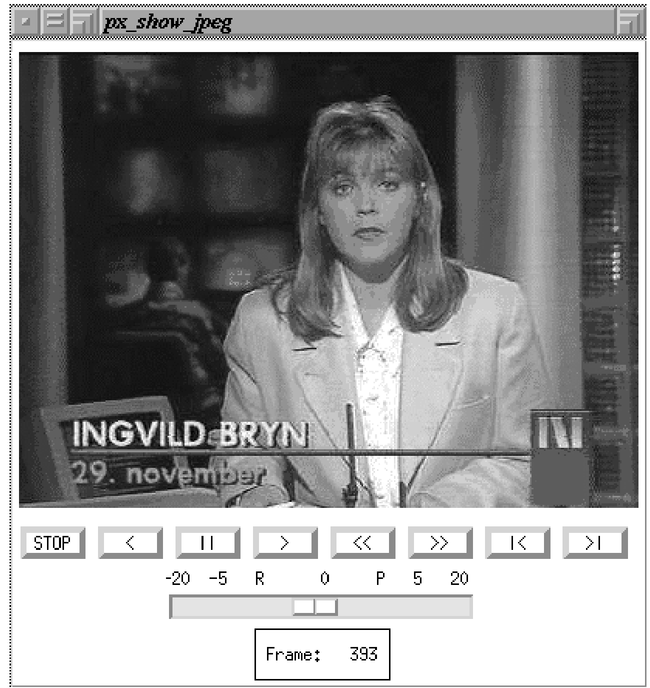

Figure 2: The Video Player

4.1 The Video Player

Figure 2 shows the video player we have implemented. This video player gives us the possibility to display digital video on a computer display. As can be seen from the figure, it has the basic functionality to be expected from a video player such as *Play*, *Stop*, *Pause*, *Fast Forward (Variable Speed)*, *Fast Backward (Variable Speed)*, and *Single Frame Stepping*.

In addition to the user interface shown in Figure 2, the video player has a programming interface. This programming interface allows other tools to control the video player by sending commands to it. It is also possible for the tools to instruct the video player to report status information regularly – e.g., sending the frame number of the current frame at regular intervals.

The video player is implemented on a Sun workstation. In the current version the videos are recorded as JPEG-compressed files [22]. To get an acceptable performance, we use a Parallax XVideo board [17] for video compression/decompression.

4.2 The Video Document Browser

We have two main types of meta-data related to a video document: content annotations, and structure information. Both of these have temporal aspects – i.e., they are only applicable to a specific part of a video document. If we are playing a video document, the set of annotations that are valid will change as we play the video document.

The main task for the video browser is to continually present to the user the meta-data related to the current part of 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 browser informs the video player about which video document to play. While the player displays the video on the screen, it regularly informs the browser application about which part of the video document it is displaying. This is done by sending the current frame number[2] to the video browser. The browser uses this frame number for retrieving and displaying information that is relevant to this part of the video.

In Figure 3, the user interface of the browser is shown. 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 3 is a snapshot of a video document that is related to four different people. The

[2] In our implementation we use frame numbers as a simplification of time codes.

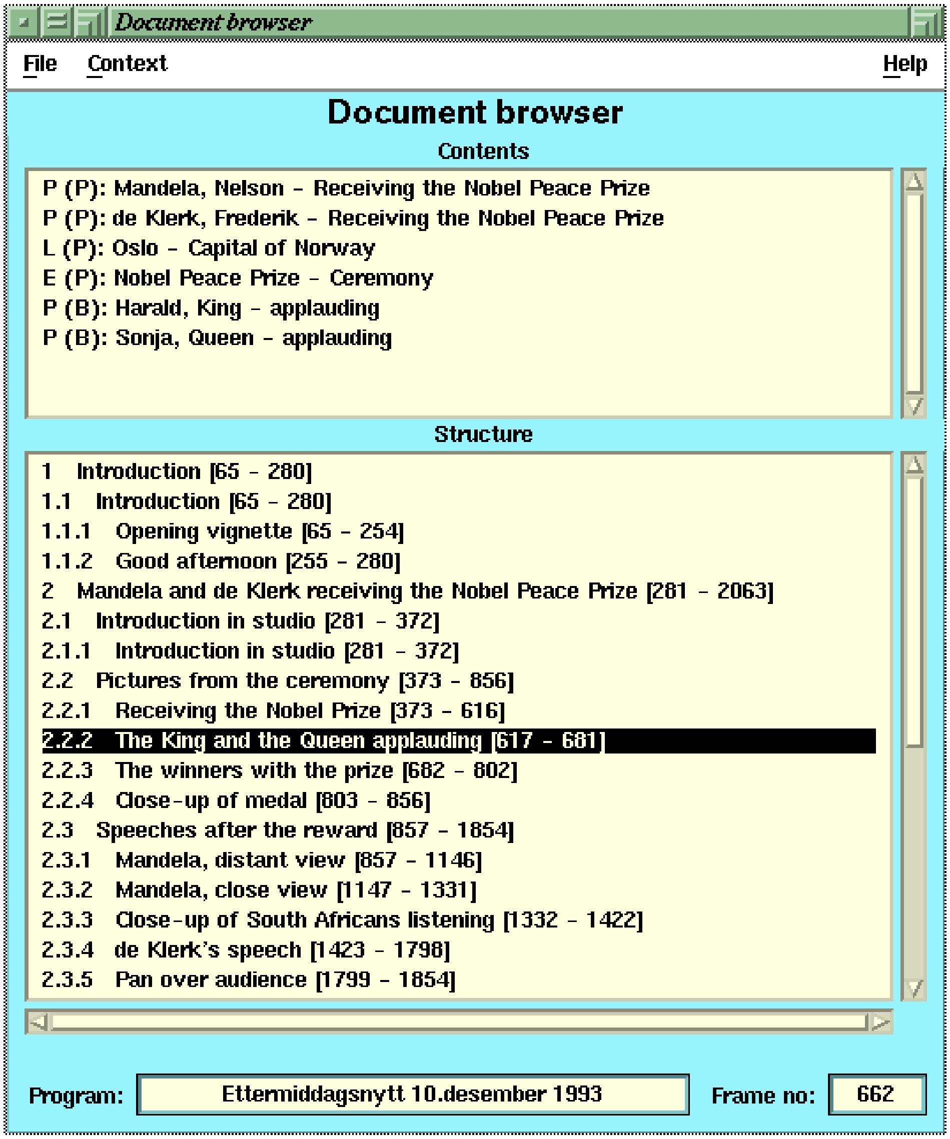

Figure 3: The Video Document Browser

event (preceded by an E) is the Nobel Peace Prize Ceremony, and the location (preceded by an L) is Oslo. We have distinguished between annotations related to the topic in question (marked with "P"), and annotations describing what is actually seen on the video (marked with "B").

Each time the browser gets a new frame number from the video player, the window is updated. Content annotations that are no longer valid are removed from the window and new annotations are displayed.

Structure Information

Structure information differs from content annotations. Each frame in the video belongs to exactly one structure component at each level. In the current implementation, we have three levels of structure: *sequences*, *scenes*, and *shots*. A document consists of one or several sequences, which again consists of one or several scenes. Finally, a scene consists of one or several shots. At each time during the playback of a video document, exactly one sequence, one scene, and one shot is defined.

The structure window of the browser is implemented in the same way as the table of contents in a book. The highlighted line shows which part of the video document currently being played. From Figure 3, we can get the following information about the structure of what is shown at the moment:

- **Sequence:** 2 Mandela and de Klerk receiving the Nobel Peace Prize

- **Scene:** 2.2 Pictures from the ceremony
- **Shot:** 2.2.2 The King and the Queen applauding

The highlighted line scrolls down the "table of contents" as the document is being played back.

The structure window can also be used for interacting with the video player. By selecting a line in the structure window, the browser will cause 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.

4.3 The Video Query Tool

Content annotations can be used for video querying, as well as for browsing. The user can select any combination of items based on the content annotations registered on the material. These items will be sent to the database, which will retrieve all pieces of video fulfilling the selection criteria.

Some possible queries related to the example in the previous section, are:

1. Retrieve all pieces of video where Nelson Mandela is shown in the picture
2. Retrieve all pieces of video about the Nobel Peace Prize
3. Retrieve all pieces of video showing both the Norwegian King and Nelson Mandela
4. Retrieve all pieces of video with Nelson Mandela in connection with the Nobel Peace Prize

The user interface of the video query tool is shown in Figure 4. The interface is divided into several windows. The three windows at the top lists all persons, events, and locations registered in the database in alphabetical order. The user can scroll these lists and select the desired items. When the user selects entries from these lists, the entries are copied to the corresponding windows below, which show the selected persons, events, and locations.

When the user presses the "Search" button at the lower right of the window, the database retrieves all pieces of video related to the selected items.

Figure 4 shows the result after querying the database for "Nelson Mandela". Four matching pieces of video are listed, and all of them are related to the Nobel Peace Prize awarded to Nelson Mandela and Frederik de Klerk in December 1993.

The user can select any item in the result list. As a response, the query tool will cause the video player to play back the corresponding piece of video. In Figure 4, the last item is selected and played. This piece of video is from the actual Nobel Peace Prize ceremony, and its structure is shown in Figure 3.

Besides the Query tool shown in Figure 4, we are also experimenting with more powerful query capabilities [12].

4.4 The Video Annotator

Registration of meta-data related to pieces of a video document is an important, but time consuming task. Different users need to have registration tools tailored to their particular way of doing registration. A news archive will probably do most of the registration of meta-data during the production process, while a museum may want to add additional information to old productions. Our registration tool has mainly been developed to show how registration tools can benefit from having direct access to the video document and from having control over the video player.

Figure 5 shows the user interface of the registration tool. Registration of annotations is done by a close cooperation between the registration tool and the video player. The user starts by playing a video document. Every time he/she wants to register that an annotation starts or ends, he/she pauses the video player which responds by sending the frame number to the registration tool. As can be seen from the figure, start and end frame numbers are associated with every annotation.

The browser divides meta-data into two categories, content annotations, and structure information. The annotation tool uses the same grouping for registering meta-data related to a video.

Content Annotations

When the user pauses the video player, he/she has the possibility to register the start of a new annotation. As mentioned previously, we have chosen to implement three different categories of content annotations in this version of the video database. 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 presses the "Location" button and gets a new window where he/she can register Oslo as the name of the location together with an explanatory text. He/she has now created what we call an *open* annotation – i.e., an annotation with a start time, but no end time. This annotation is shown in the upper left window in Figure 5. When the user later gets to the point in the video where Oslo no longer is valid as the location, the user selects its entry from the window of open annotations. The annotation is given an end time and becomes a *closed* annotation and is consequently moved to the closed annotation's window.

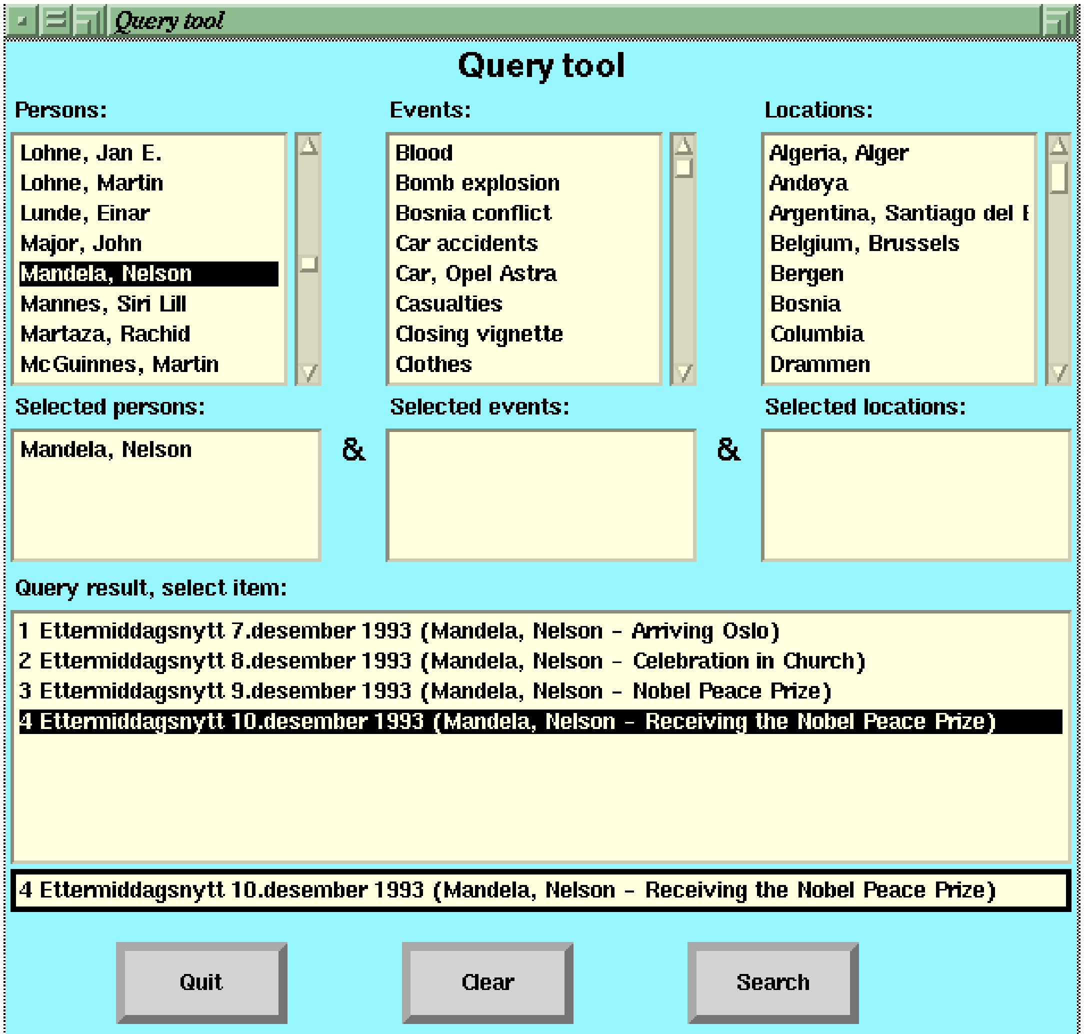

Figure 4: The Video Query Interface

Structure Information
The nature of structure components is different from content annotations. Every time a structure component, – e.g., a scene – ends, a new structure component of the same kind starts. An end of a structure component implicitly results in the end of all structure components of finer granularity – e.g., the end of a sequence also ends the current scene and shot at the same point. The tool makes use both these properties. When the user registers the end of a structure component, a new one is automatically created at the same level. For instance, if the user defines the end of a sequence, the tool automatically also ends the current scene and shot, and creates the start of a new sequence, a new scene, and a new shot.

Often, content annotations are valid in the exact same time period as one of the structure components. As a means of speeding up the annotation process, we have introduced the functionality to add content annotations that will have the same time interval as a structure component. For instance, assume that a reporter is visible during a whole sequence and we want to register a person annotation related to him/her. We can now select the sequence and press the "Person" button shown to the left of the structure window. A new person annotation will be created that will have the same start and end time as the selected sequence. This saves the user from the time consuming job of having to find the exact start and end times for the person annotation.

4.5 Tool Integration

Each of these tools can be used as stand-alone applications together with the video player (and even without the video player), and was actually first implemented as stand-alone applications. To get better support for working with video documents, we have combined them into an *integrated video tool environment*.

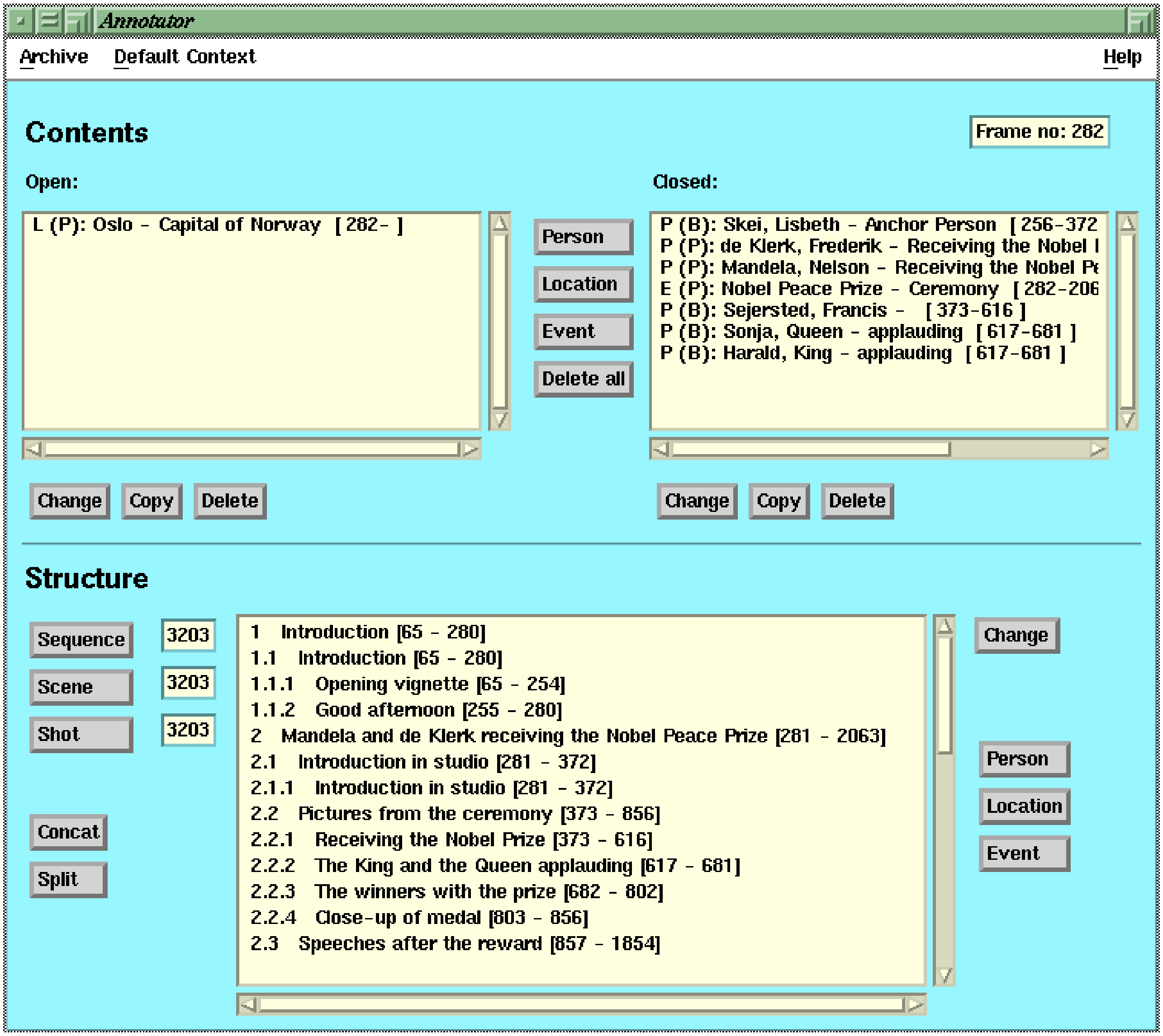

Figure 5: The Video Annotator

The *Tool Manager* shown in Figure 6 is the integrating component. From this, the user can choose which of the tools he/she needs for doing the work. The Tool Manager is also responsible for sharing information between the different tools. It acts as an *broadcast message server* and distributes messages from one tool to the other active tools.

Take Figure 4 as an example: The user has just done a search for Nelson Mandela. Assume that the user selects the last item from the result list. The query tool informs the Tool Manager about which piece of video to display for the user. The Tool Manager then sends this information to all active tools. The video player is informed what video document to play and where to start. If the browser is active, the Tool Manager informs it about what document to browse, and which part of the document the video player is displaying. The browser will respond by displaying the corresponding content annotations and structure information (see Figure 3). During the playback, the browser is regularly updated with information about which part of the video document is being displayed on the screen.

5 EXPERIENCES AND DISCUSSION

To gain experience with 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 demonstra-

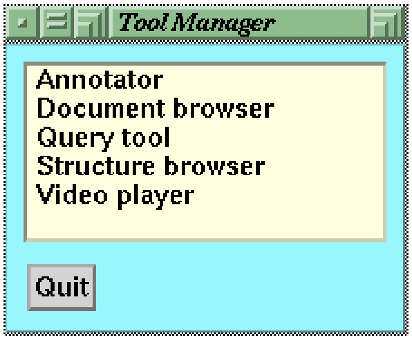

Figure 6: The Tool Manager

tions and discussions with archivists from the two national broadcasting companies, as well as with conservators from museums.

5.1 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 interface, particularly the way structure information is registered, to bring it in accordance with personal preferences and their registration practice. The Norwegian Folk Museum has, for instance, expressed the need of six different types of content annotations, 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.

A benefit of storing digital video is the possibility to provide remote access to television archives and museums of the same quality as provided for local users. The users have communicated interest in performing medium scale experiments based on our models and tools, and we are going to carry out at least one such project during 1995.

5.2 Meta-data and the Registration Process

Even with the support from the video annotator, the registration process is quite time consuming and there is a clear trade-off between this effort and the improved knowledge and searchability of the material. Users report that increased time spent on registration might be acceptable if this significantly improves and speeds up the retrieval process.

The main courses for the extensive use of time in the registration process, is the inherent temporal nature of video which makes it quite time consuming to comprehend video material, and the need of frame accuracy in the definition of video intervals. Parts of this problem could be reduced if the system had access to shot boundaries from editing tools like Avid [2], or if shot detecting tools [7, 21] were employed as preprocessing tools.

From the use of the video annotator, we have experienced that in most cases it is sensible to divide the registration process into two separate tasks: The first task consists of determining the structure components. When structural information is registered for the complete document, the document is played back a second time where the task is to register content annotations. Often, content annotations are valid in the exact same time period as one of the structure components. The mechanism described in Section 4.4 can be used to create such annotations in an efficient manner.

5.3 Video Browser Provides Context

Most video material is produced with the underlying assumption that the video is to be presented sequentially to the user such that a local context is gradually established during playback. This assumption is largely broken in an environment where the user can search for arbitrary pieces of video which have a set of desired properties. The video browser, which shows both structure and annotations during playback, provides important support in establishment of context, and reduces the need to consult surrounding video material. We believe this function of the browser to be of greatest importance when the archive contains a huge amount of video and when the user is unfamiliar with the material in the archive.

Assume that the user is interested in retrieving all pieces of video showing the Norwegian King from the database. The user will then choose "Harald, King" from the list of persons, and press the "Search" button. One of the pieces of video returned is from the news on December 10th 1993. When the user selects this item for playback, he/she will get a three second shot showing the King and the Queen applauding for something. If the browser is not active, the user will have no idea what the occasion is, and why the King and Queen are applauding. If, however, the browser is active, the browser will provide the context for this shot as shown in Figure 3. Then the user will immediately see that the Royals are applauding for Nelson Mandela and Frederik de Klerk, and that the occasion is the Nobel Peace Prize Ceremony.

5.4 Integration of Tools

In early versions of the system, the different tools were not as closely integrated as in the version described in this paper. We have experienced synergetic effects from the integration. Integration of player and annotator is invaluable to facilitate the registration process, and the close interaction between searching tools and browsing tools is important in interpretation of video material as discussed in the previous section.

5.5 Missing Features

The tools are not developed to be the ultimate solution to every user's needs in a video archive but rather as a means to exploit user needs and facilitate discussions, creativity, and further progress. It would, at this point, be fair to discuss some of the weaknesses as we see them at the time of writing:

- First and most important we would like to have much more powerful query capabilities as we have thoroughly discussed in [11, 12].
- We do not have any good means of showing the relative size of pieces of video material, for instance in query results and during browsing.
- Users may want to perform a free text search in structure component descriptions in addition to searching the content annotations.
- The database system is very simple and cannot be used in a large scale experiment.
- The mini-world model we have used to describe content annotations and structure information is hard-coded into the applications. Because of this, it will be quite a lot of work to adjust the applications to other users needs and preferences.

6 CONCLUSION AND FURTHER WORK

In this paper, we have proposed services and tools that should be offered by digital video archives/libraries. We have also given a short presentation of the VideoSTAR database framework which can be used as a platform for developing integrated video archive tools, and we have described the VideoSTAR prototype tools. The system contains a digital video player that can be controlled by the archive tools and a searching, a browsing, and an indexing/annotation tool. The system demonstrates how video data can be searched and browsed, and how the different tools can be combined to give the user powerful means for accessing a digital video archive.

We have demonstrated the system to professional archivists/librarians who have given a positive response to the concepts that the tools are based on. In the near future, we will perform a medium scale experiment together with at least one group of users to evaluate how appropriate the models and tools are in practice. We are continuously improving the tools and the framework, currently with a special focus on developing more powerful querying possibilities and on extending the video database system.

ACKNOWLEDGMENTS

The LAVA project is funded by the Norwegian Research Council, UNINETT and Telenor. We would like to thank Tormod Holte, Petter Merok, Sigurd Skeide, Arnt Sæther, and other master students at NTH which have been working 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.

This and other papers are available from World-Wide Web on URL = "`http://www.idt.unit.no/IDT/grupper/DB-grp/tech_papers/tech_papers.html`".

REFERENCES

1. Arman et al. Content-Based Browsing of Video Sequences. In *Proceedings of ACM Multimedia '94*, San Francisco, USA, October 1994, pp. 97–103.

2. Avid Technology Inc. *Avid Media Composer User's Guide*, 1993.

3. A. Bekkadal. Editing Tool for Digital Video. Master's thesis, Norwegian Institute of Technology, 1994. In Norwegian.

4. G. Cruz and R. Hill. Capturing and Playing Multimedia Events with STREAMS. In *Proceedings of ACM Multimedia 94*, San Francisco, USA, October 1994, pp. 193-200.

5. D. Deloddere, W. Verbiest, and H. Verhille. Interactive Video On Demand. *IEEE Communications Magazine*, 32(5):82–88, May 1994.

6. T. Dybå, T. Holte, and A. Sæter. LAVA Report: Analysis of Television Production. Technical report, SINTEF DELAB, December 1994. In Norwegian.

7. A. Hampapur, R. Jain, and T. Weymouth. Digital Video Segmentation. In *Proceedings of ACM Multimedia '94*, San Francisco, USA, October 1994, pp. 357–364.

8. R. Hjelsvold. Video Information Contents and Architecture. In *Proceedings of the 4th International Conference on Extending Database Technology*, Cambridge, UK, March 1994. pp. 259–272.

9. R. Hjelsvold and R. Midtstraum. Modelling and Querying Video Data. In *Proceedings of the 20th VLDB Conference*, Santiago, Chile, September 1994, pp. 686–694.

10. 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*, San Jose, CA, February 1995, pp. 268–279.

11. R. Hjelsvold, R. Midtstraum, and O. Sandstå. A Temporal Foundation of Video Databases. To appear in *Proceedings of the International Workshop on Temporal Databases*, Zürich, Switzerland, September 1995.

12. R. Hjelsvold, R. Midtstraum, and O. Sandstå. Searching and Browsing a Shared Video Database. To appear in *Proceedings of the First International Workshop on Multimedia Database Management Systems*, Blue Mountain Lake, NY, August 1995.

13. T.D.C. Little et al. A Digital On-Demand Video Service Supporting Content-Based Queries. In *Proceedings of ACM Multimedia 93*, Anaheim, USA, August 1993, pp. 427–436.

14. T.D.C. Little and D. Venkatesh. Prospects for Interactive Video-on-Demand. *IEEE Multimedia*, 1(3):14–24, Fall 1994.

15. W.E. Mackay and G. Davenport. Virtual Video Editing In Interactive Multimedia Applications. *Communications of the ACM*, 32(7):802–810, 1989.

16. 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.

17. Parallax Graphics, Inc. *XVideo User's Guide*, 1991.

18. 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.

19. P. Schnorf. Integrating Video into an Application Framework. In *Proceedings of ACM Multimedia 93*, Anaheim, USA, August 1993, pp. 411-417.

20. T.G.A. Smith and N.C. Pincever. Parsing Movies in Context. In *Proceedings of the 1991 Summer USENIX Conference*, Nashville, USA, 1991, pp. 157–167.

21. S.W. Smoliar and H. Zhang. Content-Based Video Indexing and Retrieval. *IEEE Multimedia*, 1(2):62–72, Summer 1994.

22. G.K. Wallace. The JPEG Still Picture Compression Standard. *Communications of the ACM*, 34(4):30–44, 1991.

23. R. Weiss, A. Duda, and D.K. Gifford. Composition and Search with a Video Algebra. *IEEE MultiMedia*, 2(1):12–25, Spring 1995.