Known-Item Retrieval on Broadcast TV

J.A. List, A.R. van Ballegooij, A.P. de Vries

Information Systems (INS)

INS-R0104 April 30, 2001

Report INS-R0104
ISSN 1386-3681

CWI
P.O. Box 94079
1090 GB Amsterdam
The Netherlands

Known-Item Retrieval on Broadcast TV

Johan List, Alex van Ballegooij, Arjen de Vries
{j.a.list,alex.van.ballegooij,arjen.de.vries}@cwi.nl

CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

ABSTRACT

Many content-based, multimedia retrieval systems are based on a feature-oriented approach to querying, mostly exposing a fixed set of features (introduced at design time) for querying purposes. This restriction to a limited set of features is problematic for two reasons: it restricts the expressiveness at the semantic level, and it seems unfeasible to obtain (a-priori) a sufficiently powerful set of features for all possible queries.

We describe an alternative approach where users specify precisely the distinguishable characteristics of the desired result set. In this query process, the user first describes a representation of the content (based on a feature or collection of features) and then tells the system how to apply the representation in the search.

Our prototype video retrieval system allows the expression of such queries as a sequence of operations, on MPEG video and audio streams, that can be executed on our database system. While the low-level decompression stage is implemented in an imperative programming language, the actual retrieval approach is expressed in declarative database queries. We assessed this system with a case study in known-item retrieval on broadcast video streams: detecting news bulletins in the stream, with the help of both audio and video information.

2000 Mathematics Subject Classification: 68P20, 68U99
1998 ACM Computing Classification System: Information Search and Retrieval (H.3.3), Multimedia Information Systems (H.5.1)
Keywords and Phrases: multimedia retrieval systems, multimedia database systems, known-item retrieval
Note: Work carried out under INS1 MIA project and the DRUID project and the report has been submitted as article to CBMI 2001

1. INTRODUCTION

The most challenging area of multimedia retrieval research is closing the semantic gap, i.e. combining and mapping low level media-specific features in an intelligent way to high-level concepts. Many multimedia retrieval systems focus on a feature-oriented approach to querying. A single representation of the content of multimedia objects is decided upon a-priori. Because an optimal representation for all queries does not seem to exist, researchers attempted to use a series of representations (termed a 'society of models' in [1]), and select a good representation based on user feedback, see e.g. [1, 2].

Choosing the right representation automatically is very important for naive users with generic queries. But, guessing the right query representation from user feedback is a very difficult problem, and has not been solved to a satisfactory level. The question arises whether the user can play a larger role in the 'articulation' of the query than simply saying yes or no to retrieved objects.

This paper describes an approach in which users define their information needs precisely in terms of the distinguishing characteristics of desired responses. The query does not necessarily depend on a single representation of content: it can be constructed from several representations, possibly containing information from different media types. The analogy in image retrieval is our work on the 'Image-Spotter' [3], allowing its users to designate the regions of interest in an image.

Consider for example a user searching for video segments with a space shuttle in orbit over the Earth. Looking at the characteristics of such segments, the user may come up with the following distinguishing characteristics:

Figure 1: Example Space Shuttle Search Image

- the narration track (if there is any present) is likely to contain words such as ‘space shuttle’, ‘orbit’, ‘earth’;
- there will probably be a large amount of dark colors present in such a segment;
- the space shuttle is mostly white;
- Earth itself is characterized as a collection of blue, green, brown and white colors;
- there is a certain spatial relation between the space shuttle and the Earth.

The user specifies the representations for (several of) the characteristics mentioned above, and instructs the system on how to use these representations to answer the query. Some of these characteristics may be complex and difficult to specify by hand, such as the shape of the space shuttle or the variety of colors present in the Earth. Then, an example image such as the one shown in Figure 1 serves as a reference for the system, to determine these characteristics automatically.

We present a work in progress to support the query process sketched in the (hypothetical) example above. We designed and implemented a prototype video retrieval system, and tested its merits with a case study in known-item retrieval. This case study comprised finding the delimiting segments of Dutch news bulletins, from broadcast video streams containing commercials before the start and after the ending of the news bulletins.

The structure of the rest of the paper is as follows. Section 2 discusses relevant work in the field of multimedia retrieval systems, followed by Section 3 containing our approach. Sections 4 and 5 describe our prototype system architecture and the case study respectively. Sections 6 and 7 present our experimental results, conclusions and directions for future research.

2. Multimedia Retrieval Systems

For the answering of queries, many multimedia retrieval systems focus on content-based retrieval: the analysis and extraction of low-level, media-specific features, followed by a similarity search in the feature spaces available. We have looked at two specific cases of multimedia retrieval systems: video and audio retrieval systems.

The first video retrieval systems considered video retrieval as a special case of image retrieval. An example of such a system is QBIC [4]. In QBIC, video data is first segmented at the shot level, after which key frames are chosen or generated (mosaic frames in the case of shots with panning motion). Object detection was achieved through analyzing the layered representation of video: the detection of regions with coherent motion.

VideoQ [5] introduced the concept of video objects. Video objects are sets of regions which display some amount of consistency, under certain criteria, across several frames. As in QBIC, video data is first segmented at the shot level after which the global background motion of each shot is determined. Color-, edge-, and motion-information of regions within the shot is then extracted to track possible video objects across several frames.

Query interfaces for multimedia retrieval can be distinguished into textual or visual query interfaces. Visual query interfaces mostly comprise query by feature or feature combinations (local or global features) and query by sketch or example. The VideoQ system allows the sketch of a motion trajectory for a query object as well, to capture the motion aspect present in video data, and a time duration for the searched video segment (either an absolute duration or an intuitive measure such as ’long’, ’medium’ or ’short’).

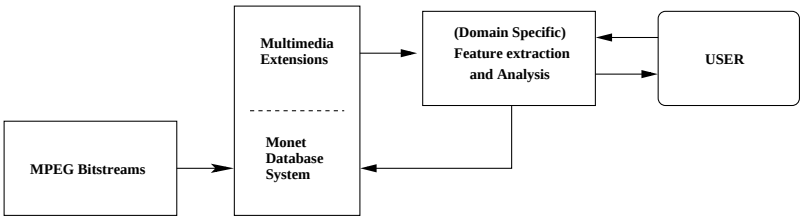


Figure 2: System Architecture

An interesting approach to querying video material, which comes closest to what we want to achieve, is described in [6] and is built on top of the VideoQ system. Querying is performed with semantic visual templates, i.e. a set of icons or example objects that form a representation of the semantic the user is searching for. An icon is an animated sketch, composed of graphical objects which resemble the real-world objects. For searching the system, features are extracted from the icons or example objects and a similarity search is performed.

A similar content-based retrieval approach has been followed in audio retrieval research [7]. Muscle-fish [8] extracts features such as loudness, pitch and brightness. In querying the audio database, the user can specify an example sound from which features are extracted and used during query processing.

SpeechSkimmer [9] is a system developed for interactively browsing or skimming of speech. The system uses time compression processing to enable users to listen to speech several times the speed of real-time, based on the notion that human speech can be understood much quicker than the normal speech rate. Speech sounds can be segmented by using information about the speaker's pitch or the use of speaker identification.

An important feature of audio retrieval systems is the query by example feature, where an example sound segment is provided to the system for querying. Instead of using example sound segments, the system developed at the University of Waikato [10] uses acoustic input for querying: the user can hum a tune. Audio files present in the system are then analyzed for tunes similar to the hummed tune, using pitch extraction or string-matching algorithms (in the case of the Waikato system).

3. Our Approach

A problem is that in many content-based retrieval systems, a standard set of features is present (introduced at design time) for the similarity searches. We deem it impossible to determine a-priori a standard set of features which can be used for answering all queries possible. Moreover, there are only so many objects, concepts or relations to describe as the used feature space (combinations) allow to express. Most importantly, it is impossible to a-priori determine which feature space best captures higher-level concepts for all possible domains, a standard feature-set may not be suitable to exploit available domain knowledge. Since domain specific search strategies generally result in better answers than generic approaches, this is an important issue.

Based on the above, we also conjure that, in the case of a layered or integrated approach to multimedia retrieval (examples include [11, 12, 13]), the restrictiveness in feature spaces present causes restrictiveness at the higher levels.

In our eyes, users can contribute to crossing the semantic gap with a top-down oriented approach, by taking advantage of the domain knowledge of the (expert) users themselves. A multimedia retrieval system should therefore offer the users the possibility to define their information need in terms of distinguishable characteristics of desired responses.

Note that we do not completely dismiss content-based retrieval based on media features. At some point during the query process, media features and the accompanying similarity search will be necessary. We only question the manner in which features are used for retrieval in many systems today.

To test and illustrate our approach, we built a prototype video retrieval system, of which the architecture is shown in Figure 2. The lowest level of the system consists of the raw bit streams (such as video streams,

audio streams and other media) that are stored in a multimedia database system.

The information that is effectively stored in the database system is the information needed to both decompress and decode the audio and video data of the MPEG bit stream. The next abstraction level consists of media-specific analysis or feature extraction elements (such as frames from a video stream or sample data from an audio stream). We propose that the construction of the elements, in this second level of abstraction, the frames and sample data in itself can be regarded as a view on the low-level bit stream.

The video frames and sample data form the basis for feature extraction and analysis, which in our system is written in a higher-level database interface language and consist of sequences of operations on database tables. This is what allows the exact feature extraction algorithms to be chosen at query-time, thus allowing for the most appropriate ones with respect to the domain to be used.

The system as described here is not a complete video retrieval system. It should be considered as a framework consisting of basic operations that requires an actual retrieval model to be built by means of a query-definition for each separate retrieval problem. The abstract interface that is a result of the use of a database system allows such systems to be build relatively easily and quickly. Nevertheless it does place a larger burden on the users. In order to reduce this burden we plan to construct an intermediate layer between the system and the user. This layer will act like a domain-independent thesaurus. This thesaurus component can offer high level primitives (such as 'list all speech segments in a certain stream' together with a description of the appropriate media-representations needed for answering this sub-query).

The prototype system as it stands now is aimed at expert users with knowledge of both video- and audio processing. Our intent is to extend the system further for making it usable by ordinary users (see Section 7).

4. System Architecture

For storage of the information needed for both decompression and decoding of the video- and audio streams (see Figure 2), we used the Monet database system [14]. The main advantage of using database systems for information retrieval task is that they "offer flexible declarative specification means and provide efficiency in execution, through algebraic optimization in the mapping process from specification to query execution plan" [15].

4.1 The Monet Database System

The Monet database system is a main-memory, parallel database system developed at CWI [14]. Its data model is based on Binary Association Tables (BATs), which are two-column tables consisting of a head and tail value. Monet offers a flexible algebra to manipulate the BATs present.

The design goals of Monet included high performance and extensibility. High performance is gained through the execution of operations in main memory, the exploitation of inter-operation parallelism, the frequent use of bulk operations which optimizes cache usage and a simple data model. Extensibility is offered through modules, in which application programmers can define new data types (called atoms), commands, operators, and procedures. Procedures are written in MIL (Monet Interface Language) which is a higher-level language defined for user-level interaction with the database.

At this point, MIL plays the role of a higher-level specification language. We plan to introduce a layer on top of MIL with a query language more suited for the multimodal access and analysis. MIL will then be used as an intermediate layer for communication between user interface components and the retrieval system as a whole.

4.2 Multimedia Extensions

For our experiments an MPEG video and an MPEG audio module were implemented in C. Decoding of the video and audio streams were implemented as commands whereas much of the feature extraction and analysis operations were written in MIL.

The video module consists of a set of database commands that allow both the *decompression* and if needed *decoding* of a given range of frames from an MPEG file. This is needed since we want users to have access to *all* data in the video stream. We consider storing complete decompressed video streams unfeasible because of the enormous amount of data present in such a stream. A rough calculation indicated that 10 hours of video (approximately 5 GB of MPEG I data) requires 500GB for storage of the decompressed video, audio and basic indexing information.

The database commands allow users to access both compressed-domain[1] and image-domain data to search in the video stream.

Analogous to the video module, the audio module primarily consists of a set of database commands that allow both the *decompression* and *decoding* of a given range of audio frames from the audio stream in an MPEG file. Even though the size of audio-data is only a fraction of the video data present in a stream many of the same database-size restrictions hold and the storage of all decompressed data is unfeasible. In addition to these decompression and decoding functions the audio module contains a set of standard signal processing functions that users can use to extract interesting features from the audio data.

The nature of the system encourages (expert) users to define custom algorithms, but we assume certain operations to be of such importance that optimized implementations are provided in the form of database operations. Examples of such optimized operations are color space conversion and histogram construction, which are relatively low-level and computationally expensive operations.

5. Case Study

In the case study used to illustrate our approach, we focused on a specific known-item retrieval problem in broadcast news, namely to find the starting and ending time indices of news broadcasts. This specific case study was chosen to complement an automated news recording system that is in use for the construction of a data warehouse of news broadcasts [16]. Until now, the determination of the exact beginning and end of each recorded news broadcast had to be done by hand.

5.1 Characteristics of News Broadcasts

News broadcasts are relatively structured and are often characterized by the following properties:

Specific "delimiter" parts Broadcast news programs often start with a specific frame or set of frames with a constant layout, see Figure 3. In our case the first frame of the opening sequence shows the NOS[2] logo on a black background. The last frame of the end sequence shows a globe surrounded by some graphics and a caption that announces the NOS website [17]. A similar argument can be made for the audio part of the broadcast. Recognizable tunes are played during the start and end sequences. Additionally, the start and end sequences are accompanied by short periods of silence.

Speech-richness of broadcast news News broadcasts often contains mostly speech, whereas the surrounding commercials are characterized by a more rapid mix of both speech and music. Also, commercial segments appear louder to the human ear in comparison to speech signals.

High cut frequency of commercial material Commercial material is often allocated a certain slot of time during which a certain amount of commercials is shown. The average duration of a commercial is 30 seconds in which all product-related information must be placed. Hence, we can expect a higher level of action (or cut frequency) during commercials.

Note that these characteristics can be collected by watching a collection of news broadcasts.

5.2 Audio and Video Feature Extraction

Given the characteristics of our domain as described in Section 5.1, we chose to use a number of suitable feature extraction and matching algorithms. Note that we chose these specific features because we found them to be applicable to this specific problem, the system in no way prescribes or limits us to the use of the features presented here. In choosing suitable feature extraction and matching algorithms, we focused on the the specific delimiter parts of Dutch news broadcast. Also note that two media types (video and audio) are used in analyzing the streams. The main idea is that, given the problem, simple, media specific features are sufficient for solving the problem when combined during retrieval.

[1] What is usually referred to as 'compressed domain' data is in reality already decompressed. MPEG video data is conceptually decoded in two stages. Firstly Huffman decompression of the bit stream results in motion vectors, and DCT domain pixel descriptions, this is the data usually referred to as *compressed-domain*. Secondly this data can be used to construct the actual images contained in the video stream.

[2] NOS stands for "Nederlandse Omroep Stichting", the Dutch broadcast organization.

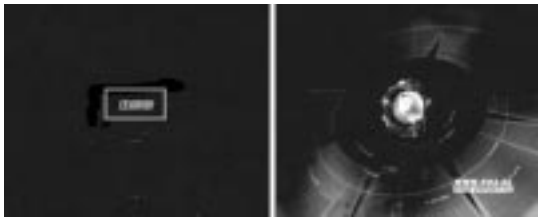


Figure 3: The begin* and end frames of a news broadcast
*Note that the box depicted in the begin image indicates the bounding box used in the image matching step and is *not* part of the actual image.

The first step of the algorithm is the analysis of the audio stream, and specifically the occurrence of high energy parts in the signal. The delimiter parts of news broadcasts are accompanied by specific tunes that are played. Both these tunes show up in the analysis as periods of high energy.

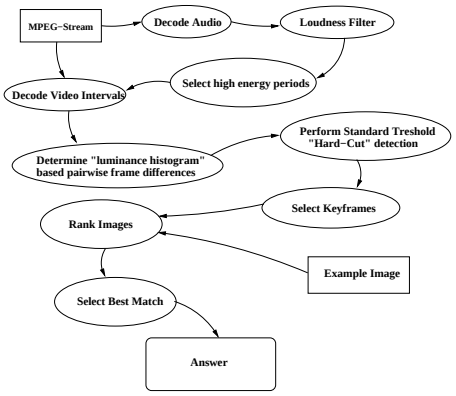
The time index generated during analysis of the audio stream is used by the video module to prune the video stream search space. The frame segments surrounding the high energy periods are decompressed and decoded and analyzed for the occurrence of the specific delimiter frames.

Audio Features Feature extraction of the audio signal uses compressed MPEG audio data [18]. Using compressed MPEG audio data for audio analysis has proven to be an efficient method giving good results compared with analysis on time-domain waveform data [19, 20]. MPEG audio compression is based on perceptual audio coding, which uses the characteristics of the human auditory system in order to gain higher compression ratios.

In order to analyze the audio signal, the approach in [20] was followed where a mean sub band vector is calculated for each frame. A frame is a collection of sub band vectors, each specifying the spectral content of 32 raw input samples. In Layer II MPEG audio a frame consists of 3 sub frames, each containing 12 groups of 32-element sub band sample vectors. A mean sub band vector can then be calculated as follows:

$$M[i] = \sqrt{\frac{\sum_{i=1}^{3*12} S_t[i]^2}{3*12}}, i = 1..32$$

The collection of mean sub band vectors is then further analyzed on energy content, which indicates the loudness of the frame, and is used in detecting periods with high energy in the audio signal. Note that the concept of high energy can indicate a variety of possibilities. A high energy segment can include music, loud speech, or loud background sounds. So care must be taken not to assume too much on the basis




```
# fname contains filename of MPEG stream to be examined
var shots := bat(int, int);
var decodedStream := decode_audio_stream(fname);
var rmsValues := [rms](mean_vectors(decodedStream);
var intervals := high_energy_thresh(high_energy(rmsValues), int(10), dbl(0.85));
intervals@batloop    { shots.insert(getShotsForTimeSlice(fname, $h, $t); }
var beginImages := shots.getKeyFrames_First(fname).
                                    rankImages("beginImage_2001");
var begin           := beginImages.reverse.fetch(0);
var endImages   := shots.getKeyFrames_Last(fname).
                                    rankImages("endImage_2001");
var end             := endImages.reverse.fetch(0);
printf("The news starts at frame %d and ends at frame %d\n", begin, end);
```

Figure 4:
(a) Query-Graph for start and end detection. (b) A snippet of the corresponding MIL query.

of high energy information, als analysis windows must be chosen with care to ensure the best possible classification. Analogous to the high energy variant, periods with low energy in the audio stream can be detected. Low energy periods can indicate silence segments.

Video Features The video data consists of MPEG-I video streams [21]. The smallest unit in a video stream is a frame, in other words a video stream is a sequence of images. In order to provide some structure in a large video stream it is useful to segment the stream into meaningful sections. There are two levels of video segmentation, the first level is the shot level, this essentially constitutes a piece of video shot by a camera in one consecutive go. A higher level of grouping could be managed in the form of scenes, scenes are groups of shots constituting a semantic unit, for example a distinct part of a story.

A simple algorithm for finding the specific delimiter parts (see Subsection 5.1) in the video stream is as follows. First the video stream is segmented at shot level. The collection of detected shots is then analyzed further to give us key frames. The key frames are then given to an image comparison operation, where each key frame is compared with a given image to find the one that matches best.

Indexing the videos at shot level is good enough for rough video segmentation. Shot level segmentation can be achieved in a number of ways, amongst which a pair-wise image comparison of consecutive frames based on color histograms [22]. Given the fact that in our problem domain, we are only looking for the hard cuts (the delimiter frames are sufficiently different from surrounding frames) the computationally relatively inexpensive approach of comparing image color histograms suffices. Histogram based cut-detection performs equally well to other more advanced cut-detection algorithms when hard-cuts are the subject of detection [23]. Although certain color spaces perform better than others empirical examination of the problem domain has shown that using only the luminance component of the images suffices to detect the fast majority of cuts in the data [24]. More importantly, this relatively simple approach has shown to have near-zero miss rate in our data-set. Of course, there is a trade-off against a higher false-hit rate.

The second step comprises key frame selection. Once again, for this problem the simplest approach is best, since we are either looking for the beginning of the news broadcast (which always starts at the beginning of a shot) or the end of the news broadcast (which ends at the end of a shot). Simply selecting either the first or last frame of a shot as the key frame is exactly what we want.

The third step is image comparison. Since we exactly know what we are looking for and the start and end sequence is always the same computer generated animation, a basic pixel-wise image comparison function is adequate. The exact algorithm measures the difference between corresponding pixels on a macro block level. For the ranking of the possible start frames only the macro blocks contained in the bounding box depicted in Figure 3 are used essentially realizing a spatial constraint. For the ranking of the possible end frames the complete image is used. The difference between macro blocks is calculated as follows:

$$MB_{diff}(a,b) = \sum_{i=1}^{256} min(\frac{(a_i - b_i)^2}{255}, 100)$$

The differences are combined in a vector, one dimension per macro-block. The length of this vector is used to express the difference between two images.

6. Results

The test-set used for the experiments was composed of several news broadcasts recorded at different times during the day over a period of a few days, stored in MPEG-1 format. The video content was encoded in standard PAL format, 25 frames of 352x288 pixels per second. The audio content was encoded with MPEG Layer II compression at 44.1KHz 16bit sampling.

The first algorithm we used was segmentation of the video data at shot level, followed by selection of key frames and a full-frame comparison of the key frames and example images. Although this approach has high precision, it is also very computationally expensive.

We attempted to realize a speed-up in searching the video stream by using the low energy information from the audio stream. Assuming that we would be able to find the silences that are present at the beginning and end of respectively the start and end sequence of the news broadcast. The introduction of the energy content analysis of the audio stream cut the execution time with roughly a factor of 2. Although we were able to identify the silences we are interested in, we noticed that a large number of short silences is present in the test set. Thus, the percentage of video material that has to be examined even after only

Method	Start Frame	End Frame	Execution Time
Video full-frames	80%	100%	$2 * D$
Video full-frames and low-energy Audio selection	80%	100%	$(40\% * 2 * D) + (\frac{1}{10}D)$
Video full-frames and high-energy Audio selection	80%	100%	$(5\% * 2 * D) + (\frac{1}{10}D)$
Video spatial and high-energy Audio selection	100%	100%	$(5\% * 2 * D) + (\frac{1}{10}D)$

Figure 5: Results of the experiments

selecting low energy periods is relatively large, about 40%. Limiting the number of low energy periods, by changing the selection of the duration of those low energy periods, resulted in unsuccessful detections. After examination of the test set material, we discovered that the silence periods accompanying the start of the news broadcasts were not of a constant duration.

Selecting high energy periods proved to be a better approach. We used the high energy information to mark possible musical segments such as those played during the start and end of the news broadcast. We discovered that this feature prunes the video search space significantly better than the low energy variant. Another advantage is that the duration of the start and end musical segments are constant. This allowed us to reduce the collection of detected high energy periods with a selection based on duration, reducing the amount of video data that needs to be examined to approximately 5% of the total duration. As a result we realized a speed-up of roughly a factor 10.

Using the high energy information, all end frames were identified correctly, but we still experienced problem with identifying the start frames in a few cases. We discovered that the misses were a result of the used image ranking algorithm. To remedy this, we added a spatial constraint to the image comparison for the start frame, by limiting the comparison to the bounding box of the logo. This counteracted the domination of the background color on the image comparison results, solving the mis-identification problem.

The fact that we obtained a 100% success rate over our test set is easily explained:

- the test set was relatively small, consisting of 10 items;
- this retrieval problem is a relatively simple problem because the characteristics of the data set are clearly defined;
- the computer generated animations from the start and end sequences are easily distinguished from other natural images present in such a video stream.

Another important characteristic of the data present in the test set is that the image quality is rather high. When lower quality recordings are to be tested it is likely that the algorithms will perform less well.

A more generic measure for gained performance by using the combined audio- and video information could be expressed as follows. Let D be the duration of the video stream. We regard D to be the same for the audio stream since we are regarding entire video streams. The decompression, decoding and analysis of the entire video stream has a duration of $2 * D$. The low and high energy analysis of the audio-stream can be performed in approximately $\frac{1}{10} * D$ seconds. Therefore if we know the fraction S of video information that needs to be examined after the audio analysis, we can approximate the total execution time as $(S * 2 * D) + (\frac{1}{10}D)$. Table 5 shows the analysis information and execution times from our different attempts.

To improve on the performance of the retrieval algorithm, other feature extraction algorithms could be introduced. Specifically introducing extraction algorithms for the other characteristics mentioned in Section 5.1 may prove to be useful. A possibility is to implement a music/speech classifier (as in [25]), which provides a more robust classification of music and speech segments and hence a more robust marking of possible start and end points of the news broadcast. Additionally a speech/music classification, combined with the fact that large parts of news broadcasts are often characterized by speech, may contribute to marking possible starting and ending points in the stream more precisely.

A second category of improvements is the optimization of the system itself. The code base used for this system is a naive implementation of MPEG decoding. We have observed optimized MPEG-decoder programs to decode the same streams at least 10 times faster. Therefore, efficiency gains can be expected in that area as well. The possibilities the Monet database system offers for parallelism can provide significant speed-up as well.

7. Conclusions and Future Research

Our experiments show that the approach we discuss is viable although our implementation still lacks performance.

The developed prototype system allowed the articulation of the query graph presented in Figure 4 in higher-level scripts and in our eyes proved to be an efficient method for expressing video retrieval queries. Admittedly, the solution for this specific problem is not generic; however, the framework we propose *is* generic. We feel that this shows that the ability to make use of domain knowledge allows the usage of simple algorithms, while obtaining a high precision. However decent user support in query formulation will have to be provided to ensure usability of the system. Furthermore the system is most useful for semi-interactive querying, depending on users to refine their query in a few iterations in order to ensure the desired precision and efficiency.

The combination of feature information from multiple media types enabled us to implement simple feature extraction algorithms to solve the problem at hand. Multimodal access also proved to be a very efficient way to prune the larger search space, the video stream. Cross-modal correlation is certainly an area to study further.

A direction for future research and extension of system are the development of a domain-independent thesaurus layer between the DBMS and user interfaces.

Another topic to study further is the research and development of caching strategies. Since no caching strategies are applied at this moment, the content representations specified at query time need to be recalculated for each query issued. A nice feature of our approach is that this can be solved as a database problem, independent of the multimedia retrieval system. Also if Monet is improved in other ways, we gain performance automatically without any direct changes to the multimedia retrieval system itself.

We also think of developing a visual query interface for ordinary users. As the system stands now, it is primarily aimed at expert users with knowledge of video- and audio processing. A visual query could use visual query graphs instead of scripts for query articulation. The query graphs are built from both a selection of available high-level media-type specific operations or feature extraction and analysis operations or user defined operations. Figure 4 presents an example of such a graph. We also propose that the construction of user-defined nodes may feed the thesaurus mentioned earlier.

References

1. T.P. Minka and R.W. Picard. Interactive learning using a "society of models". *Pattern Recognition*, 30(4):565–581, April 1997.
2. M.G.L.M. van Doorn and A.P. de Vries. The psychology of multimedia databases. In *Proceedings of the 5th ACM Digital Libraries Conference (DL'00)*, pages 1–9, San Antonio, Texas, USA, June 2000.
3. H.G.P. Bosch, A.P. de Vries, N. Nes, and M.L. Kersten. A case for Image Querying through Image Spots. In *Storage and Retrieval for Media Databases 2001, Proc. SPIE volume 4315*, pages 20–30, 2001.
4. M. Flickner, H. Sawhney, and W. Niblack. Query by Image and Video Content: the QBIC System. *IEEE Computer*, 28(9):23 – 32, 1995.
5. S.F. Chang, W. Chen, H.J. Meng, H. Sundaram, and D. Zhong. VideoQ: An Automated Content Based Video Search and Retrieval System. In *Proceedings of ACM Multimedia*, 1997.
6. S.F. Chang, W. Chen, and H. Sundaram. Semantic Visual Templates: Linking Features to Semantics. In *Proceedings of the 5th IEEE International Conference on Image Processing*, volume 3, pages 531 – 535, 1998.
7. J. Foote. An Overview of Audio Information Retrieval. *Multimedia Systems*, 7(1):2 – 10, 1999.
8. E. Wold, T. Blum, D. Keislar, and J. Wheaton. Content-Based Classification, Search and Retrieval of Audio. *IEEE Multimedia*, 3(3):27 – 36, 1996.
9. B. Arons. SpeechSkimmer: A System for interactively skimming recorded speech. *ACM Transactions on Computer Human Interaction*, 4(1):3 – 38, 1997.
10. R. McNab, L.A. Smith, I.H. Witten, C.L. Henderson, and S.J. Cunningham. Towards the Digital Music Library: Tune Retrieval From Acoustic Input. In *Proceedings Digital Libraries '96*, pages 11 – 18, 1996.
11. M.S. Hacid, C. Decleir, and J. Kouloumdjian. A Database Approach for Modeling and Querying Video Data. *IEEE Transactions on Knowledge and Data Engineering*, 12(5), 2000.
12. S. Marcus and V. Subrahmanian. Multimedia Database Systems. *http://www.cs.umd.edu/projects/hermes/publications/postscripts/mm1.ps*, 1993.
13. M. Petkovic and W. Jonker. An Overview of Data Models and Query Languages for Video Retrieval. *International Conference on Advances in Infrastructure for Electronic Business, Science and Education on the Internet, l'Aquila, Italy, July 2000 (Invited paper)*, 2000.
14. M.L. Kersten and P. Boncz. Monet: An Impressionist Sketch of an Advanced Database System. In *Proceedings Basque International Workshop on Information Technology*, 1995.

15. A. P. de Vries. Challenging Ubiquitous Inverted Files. In *Proceedings of the 1st DELOS Network of Excellence Workshop on Information Seeking, Searching and Querying in Digital Libraries, volume 01/W001 of ERCIM Workshop Reports*, pages 71–75, 2000.
16. Multimedia Indexing, Retrieval on the basis of Image Processing & Language, and Speech Technology. The DRUID Project. *http://dis.tpd.tno.nl/druid*.
17. NOS. Nederlandse Omroep Stichting. *http://www.nos.nl/*.
18. MPEG. Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1,5 Mbit/s – Part 3: Audio. Standard ISO/IEC 11172-3:1993.
19. M.A. Broadhead and C.B. Owen. Direct Manipulation of MPEG Compressed Digital Audio. In *Proceedings of Multimedia95 (ACM)*, pages 499–509, 1995.
20. G. Tzanetakis and P. Cook. Sound Analysis using MPEG Compressed Audio. In *Proceedings of the Internal Conference on Audio, Speech and Signal Processing (ICASSP00)*, 2000.
21. MPEG. Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1,5 Mbit/s – Part 2: Video. Standard ISO/IEC 11172-2:1993.
22. A. Kankanhalli H. Zhang and S. W. Smoliar. Automatic Partitioning of Full-Motion Video. *Multimedia Systems*, 1:10 – 28, 1993.
23. R. Lienhart. Comparison of Automatic Shot Boundary Detection Algorithms. In *Image and Video Processing VII 1999, Proc. SPIE 3656-29, Jan. 1999*, 1999.
24. U. Gargi and R. Kasturi. An Evaluation of Color Histogram Based Methods in Video Indexing. In *Image Databases and Multi-Media Search, Proc. IDB-MMS'96, Aug 1996*, 1996.
25. E. Scheirer and M. Slaney. Construction and Evaluation of a Robust Multifeature Speech/Music Discriminator. In *Proceedings ICASSP 1997*, pages 1331–1334, Munich, Germany, 1997.