Extending a DBMS to Support Content-Based Video Retrieval: A Formula 1 Case Study

Milan Petković[1], Vojkan Mihajlović[2], and Willem Jonker[1]

[1] Computer Science Department, University of Twente, PO BOX 217, 7500 AE, Enschede, The Netherlands
{milan, jonker}@cs.utwente.nl
http://www.cs.utwente.nl/~milan
[2] Computer Science Department, Faculty of Electrical Engineering, Beogradska 14, 18000 Nish, Yugoslavia
{vojkan.m}@elfak.ni.ac.yu

Abstract. Content-based retrieval has been identified as one of the most challenging problems, requiring a multidisciplinary research among computer vision, information retrieval, artificial intelligence, database, and other fields. In this paper, we address the specific aspect of inferring semantics automatically from raw video data. In particular, we present the Cobra video database management system that supports the integrated use of different knowledge-based methods for mapping low-level features to high-level concepts. We focus on dynamic Bayesian networks and demonstrate how they can be effectively used for fusing the evidence obtained from different media information sources. The approach is validated in the particular domain of Formula 1 race videos. For that specific domain we introduce a robust audio-visual feature extraction scheme and a text recognition and detection method. Based on numerous experiments performed with DBNs, we give some recommendations with respect to the modeling of temporal dependences and different learning algorithms. Finally, we present the experimental results for the detection of excited speech and the extraction of highlights, as well as the advantageous query capabilities of our system. *

1 Introduction

Recent developments in digital television, Internet, and information technology resulted in a demand for techniques that can manipulate the video data based on content. As database management systems do not provide enough facilities for managing and retrieving video contents, this has led to a wide range of research in this field (see [1,2,3,4] for reviews). However, the database research is not limited only to general database problems, such as modeling video as a new data type, new query languages, or spatio-temporal query processing and indexing. Nowadays, researchers meet some new difficult problems. Among them, content-based

* Proceedings of the 2nd Intl. Workshop on Multimedia Data Document Engineering, Prague, Czech Republic, 2002

A.B. Chaudhri et al. (Eds.): EDBT 2002 Workshops, LNCS 2490, pp. 318–341, 2002.

retrieval has been identified as one of the most challenging problems, requiring a multidisciplinary research among computer vision, information retrieval, artificial intelligence, database, and other fields. Only by combining these fields, researchers can find the solution for various problems that have to be solved to enable content-based retrieval. These problems include finding knowledge-based methods for interpreting raw data into semantic content, video processing, object recognition and tracking, video understanding, scalability, flexibility, dealing with unstructured data, etc.

This paper addresses these problems with the emphasis on the automatic recognition of semantic content from raw video data. With respect to this problem, video retrieval approaches presented in literature can be roughly divided into two main classes.

The first class focuses mainly on visual features that characterize colors, shapes, textures, or motion, i.e. the low-level visual content. Although these approaches use automatically extracted features to represent the video content, they do not provide semantics that describe high-level video concepts, which is much more appropriate for users when retrieving video segments.

The second class concerns annotation-based approaches, which use free-text, attribute, or keyword annotation to represent the high-level concepts of the video content. However, this results in many drawbacks. The major limitation of these approaches is that the search process is based solely on the predefined attribute information, which is associated with video segments in the process of annotation. Thus, user is restricted to small number of predefined queries for retrieving useful information from videos. Furthermore, manual annotation is tedious, subjective and time consuming.

Obviously, the main gap lies between low-level media features and high-level concepts. In order to solve this problem, several domain-dependent research efforts have been undertaken. These approaches take an advantage of using domain knowledge to facilitate extraction of high-level concepts directly from features. In particular, they mainly use information on object positions, their transitions over time, etc., and relate them to particular events (high-level concepts). For example, methods have been proposed to detect events in football [5], soccer [6], and hunting [7], etc. Motion (for review see [8]) and audio are, in isolation, very often used for event recognition. In [9] for example, extracting highlights from baseball games is based on audio only. Although these efforts resulted in the mapping from features to high-level concepts, they are essentially restricted to the extent of recognizable events, since it might become difficult to formalize complex actions of non-rigid objects using rules. Furthermore, rules require expert knowledge and have problems when dealing with uncertainty.

On the other hand, some other approaches use probabilistic methods that often exploit automatic learning capabilities to derive knowledge. For example, Naphade et al. [10] used hierarchical Hidden Markov Models (HMMs) to extract events like explosions. Structuring of video using Bayesian networks alone [11] or together with HMMs [12] has been also proposed.

Numerous approaches presented in literature have shown that is now becoming possible to extract high-level semantic events from video. However, the majority of the aforementioned approaches uses the individual visual or audio cues, and is error-prone suffering from robustness problems due to detection errors. Fusing the evidence obtained from different sources should result in more robust and accurate systems. Furthermore, some events are naturally multi-modal demanding the gathering of evidence from different media sources.

On the other hand, the fusion of the multi-modal cues is quite challenging, since it has to deal with indications obtained from different media information sources, which might contradict each other. Only a few attempts to fuse mainly audio and video sources in order to detect and recognize events have appeared recently. In [13] a probabilistic model has been used to combine results of visual and audio event detection in order to identify topics of discussion in a classroom lecture environment. Another Bayesian approach used for topic segmentation and classification in TV programs has been proposed in [14].

However, the aforementioned approaches, which mainly come from the computer vision community, have problems with scalability, because they were only intended for small collections of data. Furthermore, they stick to one technique for semantic extraction, while, as we can see from literature, different techniques are more suitable for extraction of different events. In contrast to these approaches, we propose a database approach that integrates the techniques used in computer vision within a DBMS.

From the database point of view, the contribution of this work is twofold. Firstly, we integrate video processing and feature extraction techniques into a DBMS, which allows incremental and dynamical change of metadata. Furthermore, we integrate into the system a few knowledge-based techniques, namely hidden Markov models, Dynamic Bayesian Networks (DBNs), and a rule-inference engine. We demonstrate how these techniques can be used together to automatically interpret low-level features into semantic content. By coupling these techniques with the DBMS tightly and integrating them in all three layers of the DBMS architecture (not only in one place), we achieve a high degree of scalability, flexibility, and efficiency. Secondly, our approach benefits of using domain knowledge, but at the same time, it provides a general framework that can efficiently use the aforementioned techniques in different domains.

From the computer vision perspective, we contribute by demonstrating how dynamic Bayesian networks can be effectively used for content-based video retrieval by fusing the evidence obtained from different media information sources. We validate our approach in the particular domain of Formula 1 race videos. For that specific domain we introduce a robust audio-visual feature extraction scheme and a text recognition and detection method. Based on numerous experiments performed for fusing extracted features in order to extract highlights, we give some recommendations with respect to the modeling of temporal and atemporal dependences and different learning algorithms used for DBNs. Finally, we present a user interface and some query examples that give an impression of the advantageous capabilities of our system.

2 System Architecture

The architecture of our video DBMS is easy extensible, supporting the use of different knowledge-based techniques for identifying the video contents. The content abstractions, which are stored as metadata, are used to organize, index and retrieve the video source. The metadata is populated off-line most of the time, but can also be extracted on-line in the case of dynamic feature/semantic extractions in the query time.

In order to achieve content independence, we introduce a video data model called Cobra (for a detailed formal description see [15]). The model provides a framework for automatic extraction of high-level concepts (objects and events) from raw video data. It is independent of feature/semantic extractors, providing flexibility in using different video processing and pattern recognition techniques for those purposes. The model is in line with the latest development in MPEG-7, distinguishing four distinct layers within video content: the raw data, the feature, the object and the event layer. The object and event layers are concept layers consisting of entities characterized by prominent spatial and temporal dimensions respectively. By using the Cobra video model, we achieved insulation between applications and feature/semantic extraction techniques on one, and data on the other hand.

The system is flexible in using different knowledge-based techniques for interpreting raw video data into high-level concepts. For that purpose, the system can use different techniques, such as Hidden Markov Models (HMMs), Dynamic Bayesian Networks (DBNs), neural networks, rules, etc. From the implementation point of view, flexibility is achieved by choosing an open and flexible database kernel that is easy to extend with different semantic extraction techniques. In the next section, we elaborate more on that.

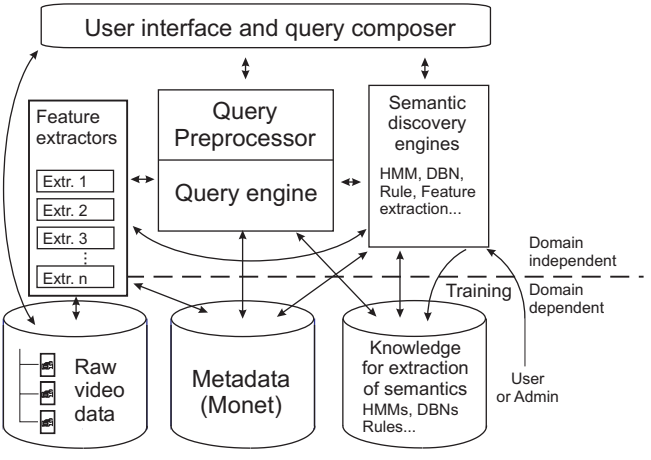

Fig. 1. The conceptual architecture

Dynamic feature/semantic extraction is facilitated by a query pre-processor. It checks the availability of required metadata needed to resolve the query. If metadata is not available it invokes feature/semantic extraction engines to extract it dynamically. The query pre-processor is also responsible for high-level optimisation during the semantic extraction. Depending on the (un)availability of metadata (features/semantics already extracted) and methods for feature/semantic extractions, as well as the cost and quality models of the method, it makes a decision which method and feature set to use to fulfil the query.

As shown in Fig. 1, domain independence is achieved by separating domain knowledge and techniques, which use it. Domain knowledge is stored within the database. Therefore the system can be used in different domains. To provide a user with the ability to query a new domain, knowledge of that domain (HMMs, DBNs, rules, etc.) has to be provided.

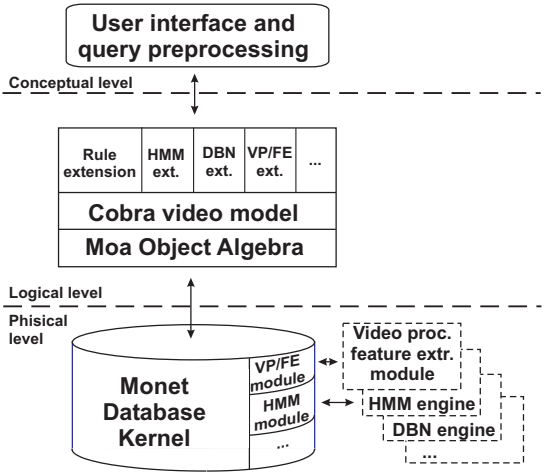

Fig. 2. The Cobra VDBMS

3 Implementation Platform

The architecture presented in the previous section is implemented within our prototype video database system that follows the well-known three-level DBMS architecture (Fig. 2).

At the conceptual level, we use an extension of the object query language. The query preprocessor rewrites a graphical query and performs preprocessing described in the previous section. The Moa object algebra [16], enriched with the Cobra video data model and several extensions, is used at the logical level. The algebra accepts all base types of the underlying physical storage system and

allows their orthogonal combination using the structure primitives: set, tuple, and object. This provides data independence between the logical and physical level, as well as possibilities for extra optimization during query execution.

At the logical level we define structures and operators that support Moa extensions. In the current implementation we have four extensions: Video-processing/feature-extraction, HMM, DBN, and rule-based extension.

The video-processing and feature-extraction extension encapsulates operations used for video segmentation, processing and feature extraction. Operations are implemented using Matlab and its image processing toolbox and as such used through a Matlab server directly by the system. At the moment, we are using the same Matlab server for the DBN extension, since the DBN learning and inference algorithms are implemented in Matlab.

The other two extensions are tightly coupled with the system. In the sequel, we will describe them very briefly (for a detailed description see [17]). The rule-based extension is implemented within the query engine. It is aimed at formalizing the descriptions of high-level concepts, as well as their extraction based on features and spatio-temporal reasoning. The HMM extension implements two basic HMM operations: training and evaluation. Here, we exploit the parallelism of our database kernel and implement the parallel evaluation of different HMMs at the physical level. Figure 3 shows a database server with the HMM extension, which calls remotely six HMM servers performing parallel evaluation. By distributing the HMM evaluation, we speed up the query processing of the very costly inference operation.

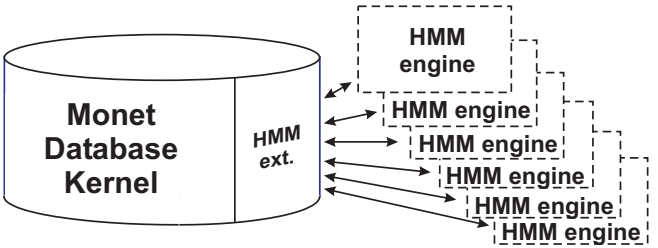

Fig. 3. Parallel HMM inference

For each Moa operation, there is a program written using an interface language understood by the physical layer. In our system, a Moa query is rewritten into Monet Interface Language (MIL), which is understood by Monet [18] - an extensible parallel database kernel that is used at the physical level. Monet supports a binary relational model, main memory query execution, extensibility with Abstract Data Types (ADTs) and new index structures, as well as parallelism.

The Moa extensions are supported at the physical level by the efficient implementation of their operations. Operations are implemented as Monet functions using MIL or as separate modules using Monet Extension Language (MEL).

For example, at the Monet level, the HMM inference operator is implemented as a MIL function, exploiting the parallel execution operator of Monet. In that way the function sends data, starts, and obtains results from 6 HMM engines in parallel (Fig. 4).

```
PROC hmmP(BAT[oid,dbl] f1, BAT[oid,dbl] f2, BAT[oid,dbl] f3,
 BAT[oid,dbl] f4) : str := {

  # preparing a observation sequence
  #(quntization of features)...
  VAR Obs:=new(void,int);
  Obs:=quant1(f1,f2,f3,f4);

  VAR parEval:=new(str,flt);

  # evaluating 6 models in parallel
  VAR BrProcesa:=threadcnt(7);
  {|

    # Service
    VAR vr:=hmmOneCall(Server1, "aMatrixS.bat", "bMatrixS.bat", Obs, num);
    parEval.insert("Service",vr);

    # Forehand
    vr:=hmmOneCall(server2, "aMatrixF.bat", "bMatrixF.bat", Obs, num);
    parEval.insert("Forehand",vr);

    # Smash
    vr:=hmmOneCall(Server3, "aMatrixSm.bat", "bMatrixSm.bat", Obs, num);
    parEval.insert("Smash",vr);

    # Backhand
    vr:=hmmOneCall1(Server4, "aMatrixB.bat", "bMatrixB.bat", Obs, num);
    parEval.insert("Backhand",vr);

    # Volley backhand
    vr:=hmmOneCall1(Server5, "aMatrixVB.bat", "bMatrixVB.bat", Obs, num);
    parEval.insert("VolleyBackhand",vr);

    # Volley forehand
    vr:=hmmOneCall1(Server6, "aMatrixVF.bat", "bMatrixVF.bat", Obs, num);
    parEval.insert("VolleyBackhand",vr);
  |}

  VAR najmanji:=parEval.max;
  VAR ret:=(parEval.reverse).find(najmanji);

  RETURN ret;
}
```

Fig. 4. Parallel evaluation of 6 HMMs

By extending our system at all levels we efficiently integrate several knowledge-based techniques within our VDBMS. This is an important advantage over approaches that implement a video extension at the application level, which results in a much slower system.

4 Dynamic Bayesian Networks

A Bayesian network is a kind of probabilistic network, which is designed for reasoning under uncertainty. Basically, it is a directed acyclic graph that describes dependencies in a probability distribution function defined over a set of variables. The nodes represent variables, while the links between nodes represent

the dependencies between the variables. Therefore, the graph can be seen as a representation of joint probability distribution for all variables.

A dynamic Bayesian network is a probabilistic network, which is able to model stochastic temporal processes. It is a special case of singly connected Bayesian networks specifically aimed at time series modeling. A time-slice of a dynamic Bayesian network is used to represent each snapshot of the evolving temporal process. A DBN satisfies the first order Markov property. So, each state at time t may depend on one or more states at time t-1 and/or some states in the same time instant. The conditional probabilities between time-slices define the state evolution model.

The parameters of a DBN can be learned from a training data set. As we work with DBNs that have hidden states, for this purpose we employ the Expectation Maximization (EM) learning algorithm, which is based on Maximum Likelihood (ML) algorithm. For inference, we use the modified Boyen-Koller algorithm for approximate inference. For a detail description of both algorithms see [19].

At the moment, the DBN extension uses a Matlab server, since the DBN learning and inference algorithms are implemented in Matlab. An operation of the MOA extension (Fig. 5a) is supported at the physical level by the implementation of a MIL procedure (Fig. 5b). The procedure sends a remote call using the TCP/IP module of Monet to the Matlab server. The Matlab server invokes the right function (Fig. 5c), which does all computations and then retrieves results back to Monet.

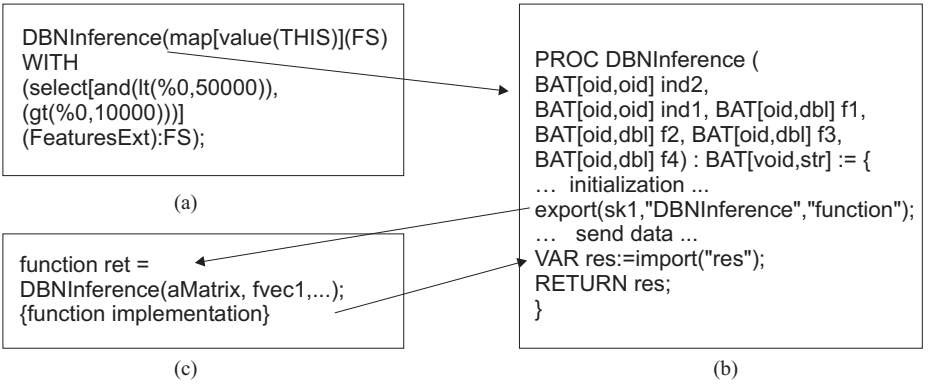

Fig. 5. Implementation of DBN inference; (a) Moa level; (b) Monet level; (c) Matlab

5 Formula 1 Case Study

In this section, we describe the extraction of multi-modal cues obtained from the three different media components of the TV Formula 1 program, as well as their fusion using dynamic Bayesian networks in order to characterize the

highlights. We will present several experiments done to investigate properties of these networks. Finally, we will demonstrate how the obtained results can be used for content-based retrieval.

5.1 Information Sources

As the majority of techniques for event detection, which rely solely on the one-media cues, showed to have robustness problems, we decided to base our analysis on the fusion of the evidence obtained from different information sources. In particular, we concentrate on three different media: audio, video and text.

Audio plays a significant role in the detection and recognition of events in video. Generally, audio information is highly correlated with visual information. In our domain, the importance of the audio signal is even bigger, since it encapsulates the reporter's comment, which can be considered as a kind of the on-line human annotation of a Formula 1 race. Furthermore, the occurrence of important events that can be classified as highlights is most of the time characterized by the commentator very well. Whenever something important happens the announcer raises his voice due to his excitement, which is a good indication for the highlights.

Visual information has been widely used for video characterization. It yields significant and useful information about the video content, but consequently it is the most difficult to automatically understand. Furthermore, the processing of visual information is very time consuming. Therefore, we made a trade-off between the usefulness of the video cues and the cost of their computation.

The third information source we use is the text that is superimposed on the screen. This is another type of on-line annotation done by the TV program producer, which is intended to help viewers to better understand the video content. The superimposed text often brings some additional information that is difficult or even impossible to deduce solely by looking at the video signal. As examples in the Formula 1 program, think of the fastest speed, lap time, order, or the visual difference between the two Ferrari cars of Michael Schumacher and Rubens Barrichello, which are almost the same[1] and can be distinguished only by diverse driver's helmets.

5.2 Audio Characterization

The audio signal of the TV broadcasting Formula 1 program is very complex and ambiguous. It consists of human speech, car noise, and various background noises, such as crowd cheering, horns, etc. Usually, the Formula 1 program involves two or more announcers, pit reports, and on-line reports received from the Formula 1 drivers. Car noise includes roaring of F1 engines, or the car braking noise. Extraction of basic characteristics from these audio recordings, which consist of complex mixtures of frequencies, is demanding and challenging task.

[1] The only difference is a very small yellow mark on Barrichello's camera

Despite this, we decided to use the audio signal to find the segments with announcer's exited speech, as well as the segments in which specific keywords are mentioned, since the audio signal is shown to be very powerful for video characterization and indexing.

Audio Feature Used. Based on a few experiments we made a selection among the variety of features that can be extracted from the audio signal. We chose Short Time Energy (STE), pitch, Mel-Frequency Cepstral Coefficients (MFCCs), and pause rate.

Short time energy represents the average waveform amplitude, defined over a specific time window. Short time energy is usually computed after performing sub-band division of wide range signal. Since indicative bands for speech characterization are lower sub-bands, we use bands below 2.5kHz in our work. Among four filters that are frequently used for the computation of STE (see [19]), we employed Hamming window filter for the calculation of Short time energy, because it brought the best results for speech endpoint detection, and excited speech indication.

Pitch is the fundamental frequency of an audio signal. In general, only speech and harmonic music have well-defined pitch, but still it can be used to characterize any audio form. Among many techniques that have been proposed for pitch estimation and tracking we used the autocorrelation analysis. All techniques for pitch estimation demand appropriate bandwidth of audio signal for accurate estimation of pitch. Since human speech is usually under 1 KHz, we are particularly interested in determining pitch that is under this frequency range.

Mel-Frequency Cepstral Coefficients are widely used for speech recognition. They are based on Mel-scale. Mel-scale is gradually warped linear spectrum, with coarser resolution on higher, and finer resolution on lower frequencies. It is metrically adapted to the human perception system. Based on this division, the Mel-frequency sub-band energy is defined. MFCCs are a simple cosine transform of the Mel-scale energy for different filtered sub-bands.

The pause rate feature is intended to determine the quantity of speech in an audio clip, which can be used as an indication of the emphasized human speech. We calculate it by counting the number of silent audio frames in an audio clip.

Audio Analysis. In order to classify human speech as excited or non-excited, first the speech endpoint detection has to be performed. For that we employ short time energy for filtered audio signal and MFCCs. We use 0-882Hz filtered audio signals in the calculation of short time energy, because this bandwidth diminishes car noises, and various background noises as well. From Mel-Frequency Cepstral Coefficients, we use only first three coefficients of the total number of 12 coefficients, because they are shown to be the most indicative for speech detection. We calculate the values of these two features for each audio frame (10 ms segments), their average values and dynamic range, and maximum values of STE for audio clips (0.1s segments). After setting the appropriate thresholds for these parameters, we were able to perform speech endpoint analysis of our

audio signal. The thresholds we used are 2.2 x 10^{-3} for the weighted sum of the average and maximum values, and dynamic range of STE, and 1.3 for the sum of the average values and dynamic range of first three Mel-frequency cepstral coefficients. As a result we get an indication for each audio clip (0.1s segment) whether it can be considered as a speech or non-speech segment. For the speech endpoint detection we performed some experiments with entropy and zero crossing rate, but they showed powerless when applied in a noisy environment such as ours.

For the detection of emphasized speech we use STE, MFCCs, pitch, and pause rate. For different features we use different frequency bands. For STE we use filtered audio signal, 882Hz - 2205Hz, and for MFCCs and pitch we use low passed audio signal, 0 - 882Hz. We compute average and maximum values in an audio clip for all these features obtained for audio frames. Additionally, we compute dynamic range for STE, and pitch as well. These computations are only performed on speech segments obtained by the speech endpoint detection algorithm. Such calculated features are then used by a probabilistic system to detect excited speech.

For the recognition of specific keywords we used a keyword-spotting tool, which is based on a finite state grammar [20]. We extract a couple of tens of words that can be usually heard when the commentator is excited, or it is a specific part of the race that we are interested in. Two different acoustic models have been tried for this purpose. One was trained for clean speech, and the other was aimed at word recognition in TV news. The latter showed better results. Thus, we employed it for keyword spotting in our system. It resulted in considerably high accuracy, but note that even better results could be obtained using a specific acoustic model for the Formula 1 TV program.

The keyword spotting system calculates the non-normalized probability for each word that is specified, the starting time when the word is recognized, as well as the duration of the recognized word. After the normalization step based on keyword spotting system outputs, these parameters are used as inputs of a probabilistic network.

5.3 Visual Analysis

In the pre-processing step of our visual analysis we segment a race video into shots. A simple histogram based algorithm is modified it the sense that we calculate the histogram difference among several consecutive frames. This algorithm resulted in the accuracy of over 90%, which we considered satisfying. The visual analysis we perform is intended to result in the visual cues that can be used to characterize replay scenes, as well as video content correlated with three different events, namely, the start of a race, passing and fly-out events.

The Formula 1 program usually contains a large amount of replay scenes. They are very important, since they always contain interesting events. The replay scenes in the Formula 1 program are usually neither slowed down, nor marked. Frequently, they begin and conclude with special shot change operations termed Digital Video Effects (DVEs). The problem is that these DVEs vary very often,

even in the same race and consequently must be frequently learned. Therefore, we decide to employ a more general algorithm based on motion flow and pattern matching [19].

Finally, we extract some visual features that indicate the three events we want to find: start, passing and fly-out. Start is defined by two parameters: (1) the amount of motion in the scene, and (2) the semaphore presence in the image. To detect the amount of motion we use pixel color difference between two consecutive frames. The semaphore is described as a rectangular shape, because the distance between red circles is small and they touch each other. This rectangular shape is increasing its horizontal dimension in regular time intervals, i.e. after a constant number of video frames. The rectangular region is detected by filtering the red component of the RGB color representation of a still image. For passing, we calculate the movement properties of several consecutive pictures, based on their motion histogram. This enables us to compute the probability that there is a chance of one car passing another. Note that we employed very general visual feature for passing detection. By applying more powerful techniques for object tracking we could obtain much better results.

Fly outs usually come with a lot of sand and dust. Therefore, we recognize presence of these two characteristics in the picture. We filter the RGB image for these colors and compute the probability, which will be used by a probabilistic network.

5.4 Text Detection and Recognition

The text that appears in a digital video can be broadly divided into two classes: scene text and graphic text. The scene text occurs as a natural part of the actual scene captured by the camera. Examples in our domain include billboards, text on vehicles, writings on human clothes, etc. The graphic (superimposed) text, which is the point of our interest, is mechanically added text to video frames in order to supplement the visual and audio content. It usually brings much more useful information, since it represents additional information for better understanding of a video scene, and is closely related to it.

Since the process of text detection and recognition is complex, we will divide it into three steps, as follows: (1) text detection, (2) refinement of text regions, and (3) text recognition. An example of text detection and recognition is shown in Fig. 6.

As the number of frames in a typical Formula 1 video is large, processing each frame for text recognition is not computationally feasible. Therefore, the first step of the text recognition task will be to find text regions in a still image. Here, we used the property of our domain that the superimposed text is placed in the bottom of the picture, while the background is shaded in order to make characters clearer, sharpened, and easier to read. The characters are usually drawn with high contrast to the dark background (light blue, yellow, or white), on the pre-specified position in each frame. Therefore, to detect whether the superimposed text is present in the picture, we simply need to process the bottom part of the picture.

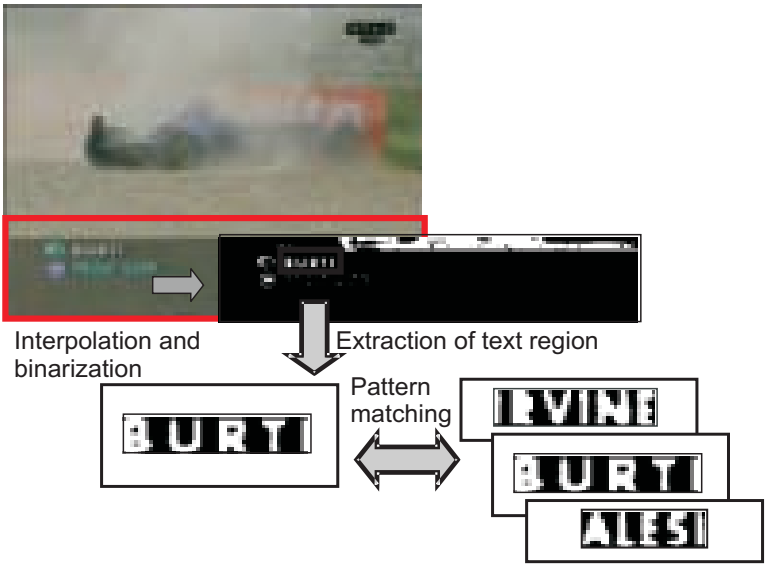

Fig. 6. Text recognition

Our text detection algorithm consists of two steps. In the first step we analyze if the shaded region is present in the bottom part on each image in a video sequence. By computing the number of these shaded regions in consecutive frames, we skip all the short segments that do not satisfy the duration criteria. In the second pass we calculate the duration, number, and variance of bright pixels present in these shaded regions. If computed values satisfy constrains defined for the text detection algorithm then this video sequence is marked as a segment that contains the superimposed text.

Such segments are further processed in the refinement process, which consists of next steps: (1) filtering of text regions, and (2) interpolation of text regions. The text regions have to be filtered in order to enable better separation from the background, as well as for sharpening the edges of characters. The filtering is done through minimizing pixel intensities over several consecutive frames. However, this filtering is not sufficient for text recognition. Therefore, we have to employ an interpolation algorithm to enlarge characters and make them clearer and cleaner. In this interpolation algorithm the text area is magnified four times in both directions. After this refinement, we have magnified text regions with much better character representations. After these actions, the text is ready for the text recognition step.

The algorithm for text recognition is based on pattern matching techniques, mainly because of the uniform structure of a small number of different words superimposed on the screen. These words are names of the Formula 1 drivers, and some informative words, such as pit stop, final lap, classification, winner, etc. Since the processing of a color image is computationally expensive and slow,

we decided to extract reference patterns, and to perform matching with black-white pictures. Black-white text regions are obtained from the color text regions by filtering RGB components. After applying thresholds on the text region, we marked characters as a white space on the black background. For character extraction we used the horizontal and the vertical projection of white pixels. Since characters can have different heights we used a double vertical projection in order to refine the characters better. However, we did not match characters to reference patterns because they are usually irregular and can be occluded or deformed. Thus, we connect characters that belong to one word into a region. This is done based on the pixel distance between characters. Regions that are closed to each other are considered as characters that belong to the same word.

Having the regions containing one word, we perform pattern matching. To speed up the matching algorithm, we separate words into several categories based on their length, and perform the matching procedure only for reference patterns with a similar length. A simple metric of pixel difference is used for pattern matching. By specifying an appropriate threshold, we were able to recognize the superimposed words. Thus, a reference pattern with the largest metric above this threshold is selected as a matched word. A more detailed description of the text detection and recognition algorithm is given in [19].

5.5 Probabilistic Fusion

In this subsection, we demonstrate how dynamic Bayesian networks can be effectively used for fusing the evidence obtained from the audio-visual analysis described above. We performed numerous experiments to compare Bayesian Networks (BNs) versus Dynamic Bayesian Networks (DBNs), different network structures, temporal dependences, and learning algorithms.

We digitalized three Formula 1 races of the 2001 season, namely, the German, Belgian, and USA Grand Prix. The average duration of a Formula 1 race is about 90 minutes or 135,000 frames for a PAL video. Videos were digitized as a quarter of the PAL standard resolution (384x288). Audio was sampled at 22kHz with 16 bits per audio sample.

Feature values, extracted from the audio and video signal, are represented as probabilistic values in range from zero to one. Since the parameters are calculated for each 0.1s, the length of feature vectors is ten times longer than the duration of the video measured in seconds. The features we extracted from a Formula 1 video are: keywords (f_1), pause rate (f_2), average values of short time energy (f^3), dynamic range of short time energy (f_4), maximum values of short time energy (f_5), average values of pitch (f_6), dynamic range of pitch (f_7), maximum values of pitch (f_8), average values of MFCCs (f_9), maximum values of MFCCs (f_{10}), part of the race (f_{11}), replay (f_{12}), color difference (f_{13}), semaphore (f_{14}), dust (f_{15}), sand (f_{16}), and motion (f_{17}). Since we also employed text detection and recognition algorithms, we were also able to extract text from the video. We decide to extract the names of Formula 1 drivers, and the semantic content of superimposed text (for example if it is a pit stop, or driver's classification is shown, etc.).

For the BN/DBN learning and inference we employed the Expectation Maximization learning algorithm and the modified Boyen-Koller algorithm for approximate inference, respectively. A detailed description of these algorithms can be found in [19].

Audio BNs and DBNs. We decided to start our experiments by comparing the results that can be achieved by employing BNs versus DBNs using different network structures. Therefore, we developed three different structures of BNs for processing only audio clues to determine exited speech, and corresponding DBN structures for the same purpose. The intention was to explore how different network structures can influence the inference step in this type of networks. The structures of BNs, which are also used for one time slice of DBNs, are depicted in Fig. 7.

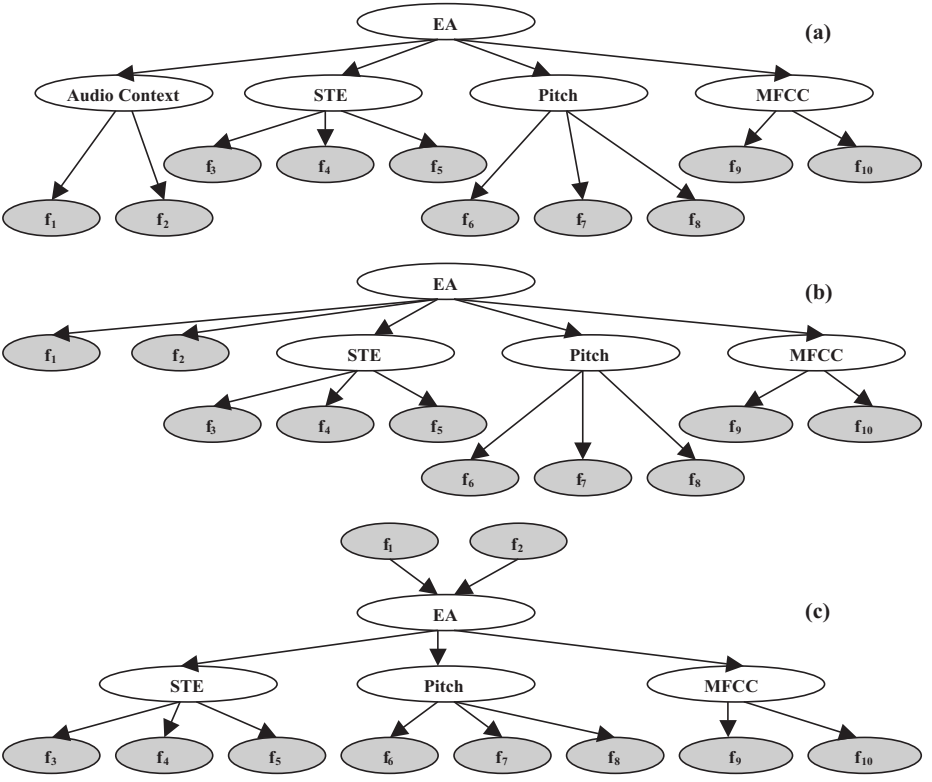

Fig. 7. Different network structures: a) Fully parameterized structure; b) Structure with the direct influence from evidence to query node; c) Input/output BN structure

The query node is Excited Announcer (EA), since we want to determine if the announcer raise his voice due to the interesting event that is taking place in

the race. The shaded nodes represent evidence nodes, which receive their values based on features extracted from the audio signal of the Formula 1 video.

The temporal dependencies between nodes from two consecutive time slices of DBNs were defined as in Fig. 8. For learning and inference algorithms we considered all nodes from one time slice as belonging to the same cluster ("exact" inference end learning).

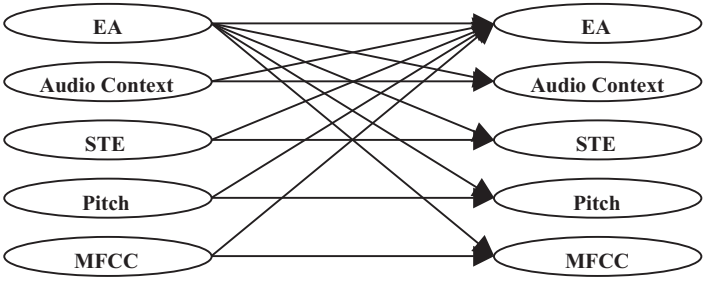

Fig. 8. Temporal dependencies for DBNs

We learned the BN parameters on a sequence of 300s, consisting of 3000 evidence values, extracted from the audio signal. For the DBNs, we used the same video sequence of 300s, which was divided into 12 segments with 25s duration each. The inference was performed on audio evidence extracted from the whole digitalized German Grand Prix. For each network structure we computed precision and recall. Note that we had to process the results obtained from BNs since the output values cannot be directly employed to distinguish the presence and time boundaries of the excited speech, as can be seen in Fig. 9a. Therefore, we accumulated values of a query node over time to make a conclusion whether the announcer is excited.

However, the results obtained from a dynamic Bayesian network are much smoother (see Fig. 9b), and we did not have to process the output. We just employed a threshold to decide whether the announcer is excited. The results from conducted experiments with previously described networks are shown in Table 1.

By comparing different BN structures we can see that there is no significant difference in precision and recall obtained from them. The corresponding DBNs did not perform much better except for the fully parameterized DBN. It gives much better results than other networks (Table 1). To see whether those results are the best that we can obtain from the extracted audio parameters, we conducted more experiments with DBNs that will be described in the sequel.

Next, we investigate the influence that different temporal dependencies have on learning and inference procedures in DBNs. We developed three DBNs with the same structure of one time slice (the fully parameterized DBN), but different temporal dependencies between two consecutive time slices. First one was with

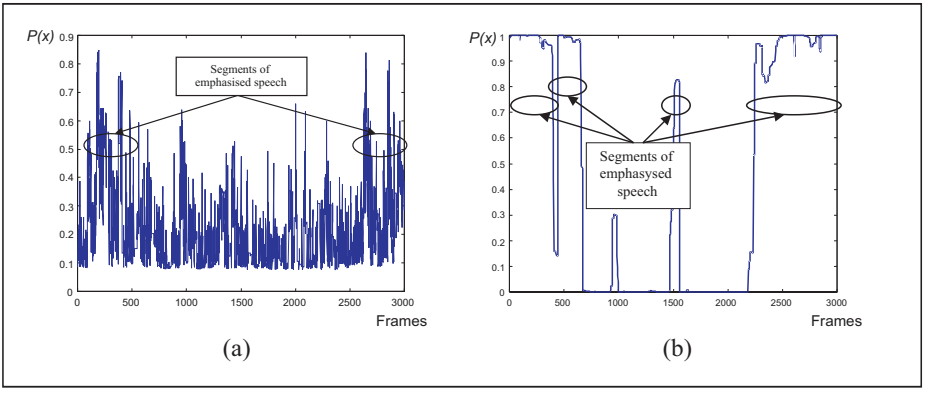

Fig. 9. Results of an audio BN (a) and DBN (b) inference for 300s long "avi" file

Table 1. Comparison of BNs and DBNs for detection of emphasized speech

Used network structure	"Fully parameterized" BN (Fig. 7a)	BN with direct evidence influence (Fig. 7b)	Input/Output BN (Fig. 7c)	"Fully parameterized" DBN (Fig. 8, Fig. 7a)
Precision	60 %	54 %	50 %	85 %
Recall	67 %	62 %	76 %	81 %

temporal dependencies shown in Fig. 8. Next one was the DBN where all non-observable nodes distribute evidence to the query node in the next time slice, and only the query node receives evidence from the previous time slice. The third one was the configuration where the query node does not distribute evidence to all non-observable nodes, but only to the query node in the next time slice. Here, all other non-observable nodes pass their values to the corresponding nodes and the query node in the next time slice. The evaluation showed that the first one significantly outperforms the second and slightly the third structure.

In addition we make experiments with different clusters formed in the fully parameterized DBN. Since our network is relatively simple we made only one experiment with clustering. In this experiment we separate non-observable nodes from the other part of the network, as proposed by Boyen and Coller in [21]. In the original network, all nodes from one time slice are assumed to be in the same cluster. Evaluation showed that the clustering technique did not bring significant changes of the recall parameter, but resulted in a larger number of misclassified sequences.

Conclusions from these experiments are twofold. From the first group of experiments we conclude that the DBN learning and inference procedure depend a lot on the selected DBN structure for one time slice. We can see that this is not the case when we perform inference and learning with BNs. These experi-

ments also showed the advantages of the fully parameterized DBN over the other BN/DBN networks. Secondly, we conclude that chosen temporal dependencies between nodes of two consecutive time slices have strong influence on the results of DBN inference. The best result was obtained from the fully parameterized DBN with temporal dependencies depicted in Fig 8.

Based on results obtained from these experiments, we selected the "fully parameterized" DBN, with one cluster for nodes in same time slice, as the most powerful DBN structure for detection of the emphasized announcer speech. To evaluate the chosen network structure we employed it for detecting the emphasized speech in the audio signal of the Belgian and USA Grand Prix. Table 1 shows recall and precision obtained by employing the DBN inference algorithm for these two races.

Table 2. Evaluation results for the audio DBN

Race	Belgian Grand Prix	USA Grand Prix
Precision	77 %	76 %
Recall	79 %	81 %

Audio-visual DBN. However, the audio DBN can only extract the segments of the Formula 1 race where the announcer raises his voice. Other interesting segments, which were missed by the announcer, could not be extracted. Therefore, the employment of the audio DBN for highlight extraction would lead to high precision, but low recall (if we count replay scenes, recall will be about 50%).

To improve the results obtained solely from audio cues we developed an audio-visual DBN for highlight detection. The structure that represents one time slice of this network is depicted in Fig. 10. The Highlight node was chosen to be the main query node, while we also queried nodes: Start, Fly Out, and Passing, in our experiments. Chosen temporal dependencies between nodes in this network are shown in Fig. 11.

Experiments were done similarly as for the audio DBNs. We employed the learning algorithm on 6 sequences with 50s duration each. The results obtained by applying the audio-visual DBN to the German Grand Prix are shown in Table 3. The precision and recall for highlights are calculated based on the probability threshold of 0.5, and minimal time duration of 6s.

The values of the other query nodes are calculated based on the value of the main query node. We calculated the most probable candidates during each "highlight" segment, and pronounce it as a start, fly out, or passing based on values of corresponding nodes. For segments longer than 15s we performed this operation every 5s to enable multiple selections.

The supplemental query nodes are incorporated in the scheme in order to classify different interesting events that takes place in the Formula 1 race. We

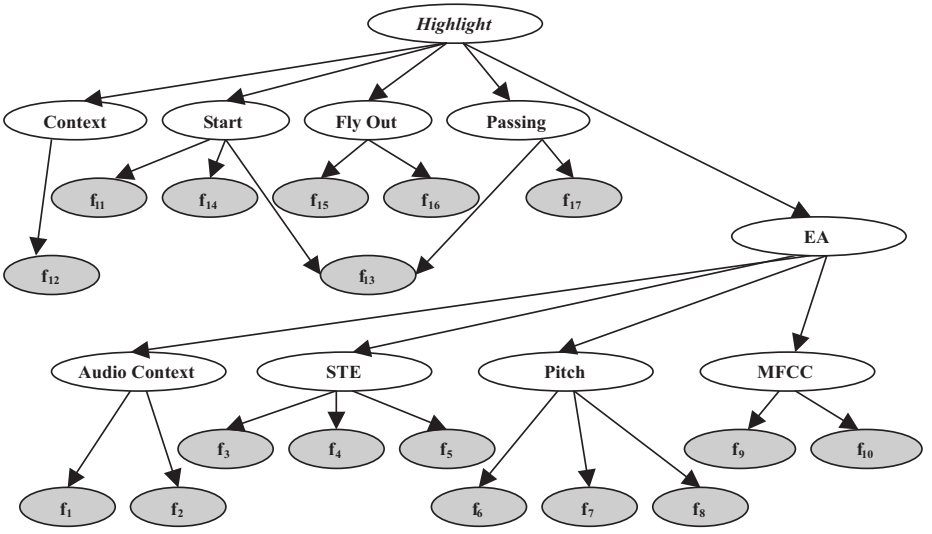

Fig. 10. Audio-visual DBN for one time slice

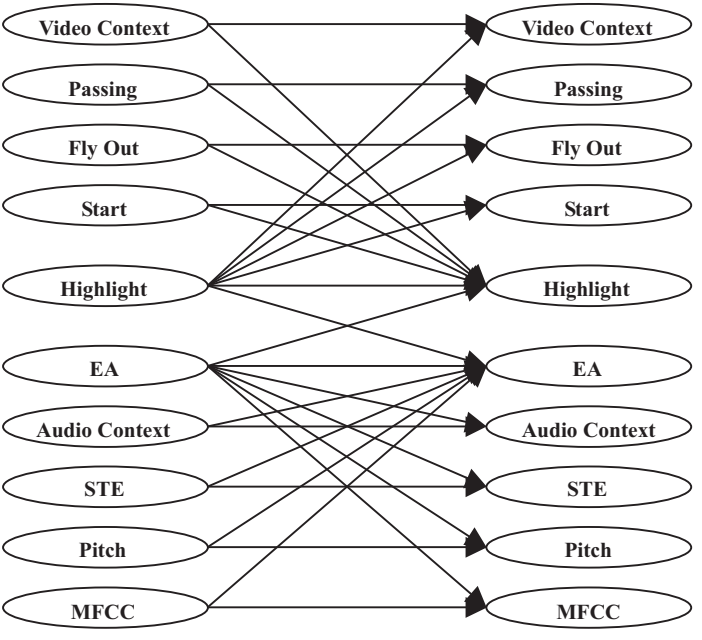

Fig. 11. Temporal dependencies for DBNs

Table 3. The audio-visual DBN

Audio-visual DBN		German Grand Prix
Highlights	Precision	84 %
	Recall	86 %
Start	Precision	83 %
	Recall	100 %
Fly Out	Precision	64 %
	Recall	78 %
Passing	Precision	79 %
	Recall	50 %

can see from Table 3 that we gained high accuracy for highlights and start, while the accuracy for fly out and passing were a little bit lower. Main reason for this is that we used very general and less powerful video cues for fly out, and especially passing. We performed evaluation of the same network structure on the Belgium and the USA Grand Prix, but we had a big decrement in our results, mostly because of the "passing" part of the network. Therefore, we simplified the overall audio-visual network, and excluded the "passing" sub-network. A significant difference in results obtained with and without the passing sub-network is presented in Table 4.

The network with the passing sub-network worked fine in the case of the German GP, but failed with the other two races. The explanation for this is a different camera work in the German GP. This just confirms the fact that general low-level visual features might yield very poor results in the context of high-level concepts (to characterize passing we used motion). Obviously, more domain dependent features, which characterize the trajectories of Formula 1 cars, will be much robust and give a better result for the passing event.

Table 4. Evaluation results for audio-visual DBN

Audio-visual DBN		Belgian Grand Prix [2]	USA Grand Prix
Highlights	Precision	44 %	73 %
	Recall	53 %	76 %
Start	Precision	100 %	100 %
	Recall	67 %	50 %
Fly Out	Precision	100 %	0 %[3]
	Recall	36 %	0 %[3]
Passing	Precision	28 %	
	Recall	31 %	

[2] These results were obtained by the audio-visual DBN that includes the passing subnetwork

[3] There were no fly-outs in the USA Grand Prix

5.6 Content-Based Retrieval

Except for highlights and the three events modeled by the DBN, our system can be used to query the Formula 1 videos based on recognized superimposed text, as well as based on audio-visual features directly. A user can ask for the race winner, the classification in the ith lap, the position of a driver in the ith lap, relative positions of two drivers in the ith lap, pit stop of a specific driver, the final lap, etc. To give an impression of the system capabilities, in the sequel we will list some query examples:

"Retrieve the video sequences showing the car of Michael Schumacher"

"Retrieve the video sequences with Michael Schumacher leading the race"

"Retrieve the video sequences where Michael Schumacher is first, and Mika Hakkinen is second"

"Retrieve the video sequences showing Barrichello in the pit stop"

"Retrieve the sequences with the race leader crossing the finish line"

"Retrieve all fly outs"...

Furthermore, our system benefits of combining the results obtained from Bayesian fusion and text recognition, and is capable to answer very detailed complex queries, such as:

"Retrieve all highlights showing the car of Michael Schumacher"

"Retrieve all fly outs of Mika Hakkinen in this season"

"Retrieve all highlights at the pit line involving Juan Pablo Montoya"...

In order to better demonstrate the advantages of the proposed system and simplify the querying process, we developed a graphical user interface. The interface is developed on the top of our DBMS using Java and Java Media Framework for video manipulation. Fig. 12 shows how the Barrichello's pit-stop query can be defined. The user interface allows a user to combine results obtained from the DBN and text detection. In addition, a user can define new compound events by specifying different temporal relationships among already defined events. He can also update meta-data through the interface by adding a newly defined event, which will speed up the future retrieval of this event.

6 Conclusions

This paper addresses the problem of recognizing semantic content in video data based on visual features. We have presented the architecture and implementation platform of our prototype video database management system. The system provides a framework for automatic extraction of high-level concepts (objects and events) from raw video data. It is independent of feature/semantic extractors, providing flexibility in using different video processing and pattern recognition techniques for that purpose.

The automatic extraction of concepts from raw video data is supported by few extensions. The video processing and feature extraction extension is used for video segmentation and feature extraction purposes. The rule-based extension formalizes descriptions of high-level concepts using spatio-temporal reasoning. Finally, the stochastic extensions exploit the learning capability of hidden

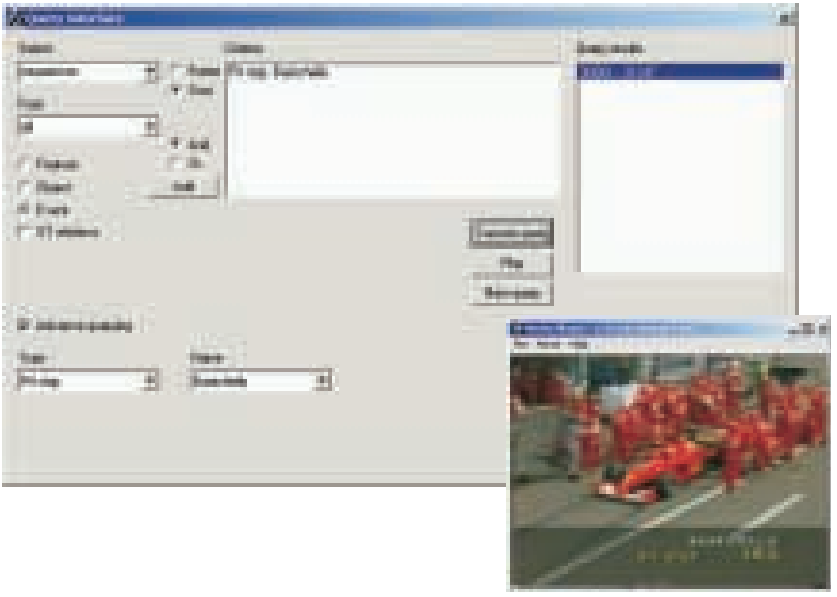

Fig. 12. The user interface

Markov models and DBNs to recognize events in video data automatically. By integrating these techniques within the DBMS, we provide users with ability to define and extract events dynamically. For example, a user can define the model for a new event by indicating example sequences and training the model (in the case of HMMs and DBNs). Then, he already can query the database.

In this paper, we focus on the DBNs, and their use for content-based retrieval, which is, to the best of our knowledge, the first time they are used for such purpose. We have conducted numerous experiments with different DBN and BN structures, and compare two different DBN learning algorithms. We have also explored the influence of different atemporal and temporal connections within a dynamic Bayesian network. Expectedly, the DBNs have outperformed BNs in our application. For DBNs, the exact representation of temporal dependencies has been found as the most powerful for learning and inference. We have shown that the structure and temporal connections within a DBN have strong influence on the learning and inference procedures.

The approach has been validated for retrieval in the particular domain of the Formula 1 TV program. We have based our analysis on the fusion of the evidence obtained from different information sources (audio, video, and text). Consequently, a robust feature extraction scheme has been introduced for the audio-visual analysis of our particular domain. For text detection and recognition, we presented a new technique, which is based on properties of Formula 1 race videos.

We can conclude that the usage of cues from the three different media has resulted in much better characterization of Formula 1 races. The audio DBN was able only to detect 50% of all interesting segments in the race, while the integrated audio-visual DBN was able to correct the results and detect about 80% of interesting segments in the race. However, this audio part is still useful for the detection of the segments with the excited announcer speech, where it showed high recognition accuracy. By integrating the superimposed text, audio and video subsystems we have built a powerful tool for indexing the Formula 1 races videos, which can answer very detailed and specific quires.

Although we have already presented a significant amount of work done to enable indexing and characterization of the multimedia documents of Formula 1 race, we state that still many improvements can be done. The main one is in the video analysis, where we only used the simplest features. For example, the problem of detecting and tracking moving objects supplemented with a lot of camera work and shot change is a challenging computer vision problem, which needs a further research.

References

1. Grosky, W.:Managing Multimedia Information in Database System. Communications of the ACM, 40(12), (1997) pp. 73-80.
2. Yoshitaka, A., Ichikawa, T.,:A Survey on Content-Based Retrieval for Multimedia Databases. IEEE Transactions on Knowledge and Data Engineering, 11(1), (1999) pp. 81-93.
3. Del Bimbo, A.:Visual Information Retrieval. Morgan Kaufmann, San Francisco, California (1999)
4. W. Al-Khatib, Y. Day, A. Ghafoor, P. Berra: Semantic Modeling and Knowledge Representation in Multimedia Databases. IEEE Transactions on Knowledge and Data Engineering, 11(1), (1999), pp. 64-80.
5. S. Intille, A. Bobick: Visual Tracking Using Closed-Worlds. Tech. Report No. 294, M.I.T. Media Laboratory, (1994)
6. Y. Gong, L. T. Sin, C. H. Chuan, H-J. Zhang, M. Sakauchi: Automatic Parsing of TV Soccer Programs. In Proc. of IEEE International Conference on Multimedia Computing and Systems, Washington D.C., (1995), pp. 167-174.
7. N. Haering, R.J. Qian, M.I. Sezan: "A semantic event-detection approach and its application to detecting hunts in wildlife video. Circuits and Systems for Video Technology, IEEE Transactions on, 10(6), Sept. 2000, pp. 857-868.
8. M. Shah, R. Jain (eds): Motion-Based Recognition. Kluwer Academic Publishers, (1997)
9. Y. Rui, A. Gupta, A. Acero: Automatically Extracting Highlights for TV Baseball Programs. In Proc. of ACM Multimedia, Los Angeles, CA, 2000, pp. 105-115.
10. M. Naphade, T. Kristjansson, B. Frey, T.S. Huang: Probabilistic multimedia objects (multijects): A novel approach to indexing and retrieval in multimedia systems. In Proc. of the IEEE ICIP, Chicago, IL, 1998, vol. 3, pp. 536-540.
11. N. Vasconcelos, A. Lippman: Bayesian Modeling of video editing and structure: Semantic features for video summarization and browsing. In Proc. of the IEEE ICIP, Chicago, IL, 1998, vol. 2, pp. 550-555.

12. A.M. Ferman, A.M. Tekalp: Probabilistic Analysis and Extraction of Video Content. In Proc. of the IEEE ICIP, Tokyo, Japan, 1999, vol. 2, pp. 91-95.
13. T. Syeda-Mahmood, S. Srinivasan: Detecting Topical Events in Digital Video. In Proc. of ACM Multimedia, Los Angeles, CA, 2000, pp. 85-94.
14. R.S. Jasinschi, N. Dimitrova, T. McGee, L. Agnihotri, J. Zimmerman, D. Li, "Integrated Multimedia Processing for Topic Segmentation and Classification", Proc. of IEEE ICIP, Greece, 2001.
15. M. Petković, W. Jonker: A Framework for Video Modelling. In the Proc. of International Conference on Applied Informatics, Innsbruck, 2000.
16. P. Boncz, A.N. Wilschut, M.L. Kersten: Flattering an objects algebra to provide performance", In Proc. of the IEEE Intl. Conf. on Data Engineering, Orlando, pp. 568-577, 1998.
17. M. Petković, W. Jonker: Content-Based Video Retrieval by Integrating Spatio-Temporal and Stochastic Recognition of Events. IEEE International Workshop on Detection and Recognition of Events in Video, Vancouver, Canada, July 2001.
18. P. Boncz, M.L. Kersten, Monet: An Impressionist Sketch of an Advanced Database System. Basque International Workshop on Information Technology, San Sebastian, 1995.
19. V. Mihajlović, M. Petković: Automatic Annotation of Formula 1 Races for Content-based Video Retrieval", Technical Report, TR-CTIT-01-41, 2001.
20. J. Christie: Completion of TNO-Abbot Research Project. Cambridge University Engeneering Department, Cambridge, England, December 1996.
21. X. Boyen, D. Koller: Tractable Inference for Complex Stochastic Processes. Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers, 1998.