PERGAMON

Information Systems 27 (2002) 559–586

www.elsevier.com/locate/infosys

On-line knowledge- and rule-based video classification system for video indexing and dissemination

Wensheng Zhou[a,b,*], Son Dao[a], C.-C. Jay Kuo[b]

[a] Information Science Laboratory, HRL Laboratories LLC, Malibu, CA 90265-4799, USA
[b] Department of EE—Systems, University of Southern California, Los Angeles, CA 90089, USA

Abstract

Current information and communication technologies provide the infrastructure to transport bits anywhere, but do not indicate how to easily and precisely access and/or route information at the semantic level. To facilitate intelligent access to the rich multimedia data over the Internet, we develop an on-line knowledge- and rule-based video classification system that supports automatic "indexing" and "filtering" based on the semantic concept hierarchy. This paper investigates the use of video and audio content analysis, feature extraction and clustering techniques for further video semantic concept classification. A supervised rule-based video classification system is proposed using video automatic segmentation, annotation and summarization techniques for seamless information browsing and updating. In the proposed system, a real-time scene-change detection proxy performs an initial video-structuring process by splitting a video clip into scenes. Motional, visual and audio features are extracted in real-time for every detected scene by using on-line feature-extraction proxies. Higher semantics are then derived through a joint use of low-level features along with classification rules in the knowledge base. Classification rules are derived through a supervised learning process that relies on some representative samples from each semantic category. An indexing and filtering process can now be built using the semantic concept hierarchy to personalize multimedia data based on users' interests. In real-time filtering, multiple video streams are blocked, combined, or sent to certain channels depending on whether or not the video streams are matched with the user's profile. We have extensively experimented and evaluated the classification and filtering techniques using basketball sports video data. In particular, in our experiment, the basketball video structure is examined and categorized into different classes according to distinct motional, visual and audio characteristics features by a rule-based classifier. The concept hierarchy describing the motional/visual/audio feature descriptors and their statistical relationships are reported in this paper along with detailed experimental results using on-line sports videos. © 2002 Published by Elsevier Science Ltd.

Keywords: Video indexing; Video semantic content; Video classification; Feature extraction; Internet video access; Decision tree; Rule-based reasoning; Knowledge-based systems; Multicast video; Internet video; Video summarization; Video annotation; Video semantics inference; Video content filtering

1. Introduction

New integrated multimedia services are emerging from the rapid technological advances in networking, multi-agents, media and broadcasting

*Corresponding author. Information Science Laboratory, HRL Laboratories LLC., Malibu, CA 90265-4799, USA. Tel.: +1-310-317-5278.

E-mail addresses: wzhou@hrl.com (W. Zhou), skdao@hr.-com (S. Dao), cckuo@sipi.usc.edu (C.-C. Jay Kuo).

0306-4379/02/$ - see front matter © 2002 Published by Elsevier Science Ltd.
PII: S0306-4379(02)00018-2

technologies. The development allows for large amounts of multimedia information to be distributed and shared on the Internet. Current information and communication technologies provide the infrastructure to transport bits anywhere, but the technologies do not presume to handle information at the semantic level due to insufficient indexing mechanisms and lack of good automated semantic extraction and interpretation mechanisms. Consequently the huge amounts of multimedia data impose a heavy burden of data manipulation on people, including searching/choosing, interpreting, skimming, and integrating information.

Smart Push and active Pull applications, such as user agent-driven media selection and filtering, personalized television services and intelligent media presentation, follow a paradigm more akin to broadcasting and thereby influence the emerging pattern of multicasting over the Internet. Such applications require the ability to analyze and index contents on-the-fly rather than by the normal store-and-analyze-later paradigm. Therefore, a distributed proxies architecture coupled with real-time indexing and content analysis techniques needs to be developed to support the requirements.

Traditional content-based video retrieval development focuses on the use of low-level features such as color, motion and texture to index video. However, while direct application of generic similarity metrics techniques to low-level features can give good results in cases having approximately similar "patterns", their application to discerning similar semantic classes is highly aspect. This is partly because the effective joint combination of multiple low-level features is a very difficult problem since it is hard to create generic models for multiple types or applications of video at once. Thus, efficient video classification into semantically meaningful classes will require more supervised approaches; besides, a static classification model is application dependent and as a result, it may end up not suitable for on-line applications because real-time life streams tend to be very versatile. The difference in our work and the existing accomplishments in the literature is that while most of them use a static model for video classification to provide semantic indexing of off-line multimedia databases, we have taken an approach using a supervised learning technique to form a classification system and applied it specifically to basketball video event indexing as an experimental example. Moreover, an effective and real-time multimedia data-sharing, filtering, and dissemination infrastructure is proposed using internet multicast protocols.

More specifically, we apply an inductive decision-tree learning method to arrive at a set of if–then rules that can be directly applied to a set of low-level-feature-matching functions. This decision tree is our trained knowledge and forms the on-line classifier. This knowledge-based representation approach is especially useful for on-line video semantic classification, indexing and filtering because it provides powerful rules that can be easily associated directly with the characteristic features of each class. Moreover, the rules show the order and priority of each low-level feature in the classifier when multiple features are present, which is very important for on-line fast video understanding and indexing. The proposed system also includes approximate accuracy and confidence measures for each classification output. Using this system, we have experimented and classified basketball video data into different categories such as left fastbreak, right fastbreak, left dunk-like events, right dunk-like events, close-up video sequences and so on. Such classification of high-level semantics can be used to answer queries such as "show all dunk-like shots where team A scored", as well as to support smart browsing of basketball games.

The rest of the paper is organized as follows. Section 2 gives a brief description of related work. The background and motivation of this research are also discussed in this section. The system architecture and the video data model are described in Section 3, which also details the classification-rule learning by the decision-tree learning algorithm and the on-line knowledge-based classification system. This section also describes knowledge creation, and video high-level semantic content analysis and query/filtering procedures based on the knowledge information stored in the knowledge base. Section 4 describes the implementation of on-line content analysis agents based on low-level feature-extraction processes. Section 5 gives the experimental results for our classifier and video applications over the Internet. The proposed system is evaluated with an application example of on-line basketball event indexing, and dissemination by filtering. Section 6 concludes the paper.

2. Background and related work

2.1. Background and motivation

Generally, the first step in video analysis is scene and shot boundary detection, which parses video into a collection of scenes and shots. Each scene can be represented by a sequence of shots, and the shots can be summarized by a couple of key frames (see Fig. 1(a)). The key frames can be further summarized according to low-level features such as color, shape, and motion. Based on this feature-based analysis, it is possible to perform effective video parsing to support video summarization, fast browsing and low-level content indexing.

On the other hand, video programs can be divided into stories at the semantic level [1], which are contained at every level of video streams; for example, video semantic contents are expressed and represented by video scenes, shots and key frames (see Fig. 1(b) and (c)). Low-level features, such as key frames and objects are widely used for content-based retrieval and are relatively easier to extract automatically. Indeed, up to the present, video data are still being annotated manually or semi-manually with key words in most applications. Automatic video semantic content analysis and extraction remains a very challenging research topic. Effective video classification can automatically group visual/audio multimedia into a certain level of semantic concepts. In addition, a flexible knowledge representation scheme coupled with reasoning and learning capabilities to bridge the gap between the video low-level features and high-level semantic concepts would facilitate the semantic level query and filtering for on-line video dissemination.

2.2. Issues in on-line and off-line video analysis

We distinguish between on-line and off-line video content analysis because the issues are considerably different in the two cases. Automatic

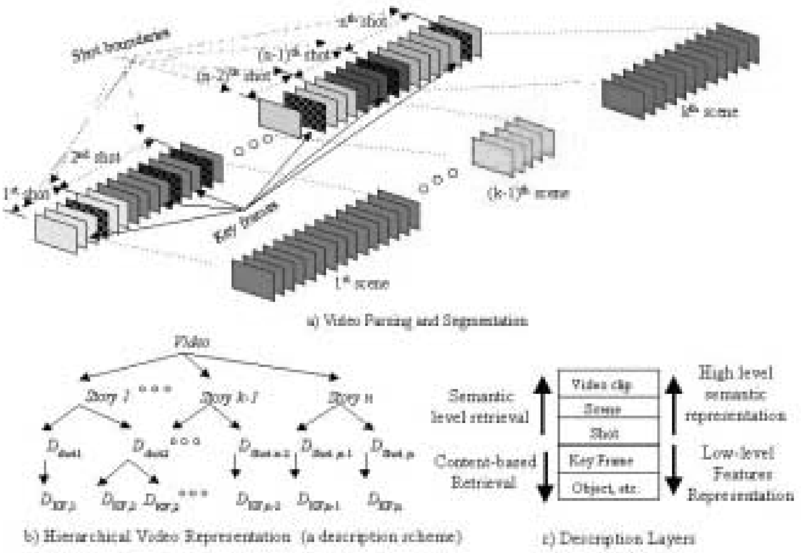


Fig. 1. A framework of video parsing and description: (a) Video parsing and segmentation, (b) Hierarchical video representation (a description scheme), and (c) Description layers.

video content analysis and annotation are generated by processing either directly on the media raw data or on lower-level annotations of features. Some examples of video content analysis and annotation are topic key words summarization or key frame summarization of videos, video scene-change tags to break video into visual meaning units and concept identifiers of a given collaboration session, etc.

There exist some unique characteristics in on-line video analysis compared with the off-line case. First, it is necessary to take into account the fact that video data, by nature, tends to be relatively larger in size as compared to other types of data. This means that real-time processing of video data should consider low-complexity techniques, such as fast image processing, so as to facilitate efficient content extraction. In other words, to avoid generating huge latencies caused by content analysis, we may need to trade off the complexity of the content analysis algorithm for increased speed, e.g. use the simpler binary classification.

Second, semantic extraction directly from visual data has been traditionally very difficult. Content-based (or feature-based) video retrieval is not efficient due to the lack of a comprehensive data model that captures structured abstractions and knowledge needed for video retrieval based on concepts. On the other hand, pixel-matching or feature-similarly matching methods employed by query-by-example techniques are time consuming, and have a limited practical use since little of video object semantics is explicitly modeled.

Third, not all video content annotations are generated by directly processing raw data because the annotations that are directly generated from raw data are very low level and not directly capable of aiding high-level decisions. Typically, further processing has to be on certain metadata (or annotations) to generate the higher-level annotations which are better suited for aiding high-level decisions. For example, video key frames can be used for fast browsing, but key frames alone are still very inefficient for semantic interpretation of video as they are raw images, too. To get the semantic meaning expressed in each key frame, visual features should first be analyzed and then pattern classification conducted, based on these extracted visual features.

Another unique characteristic is that many visual and motional features in video are based on multiple attributes, or based on other feature-extraction operations, so cooperation and synchronization techniques are necessary. Moreover, on-line video normally covers a huge range of topics which complicates the problem, so that we may need to learn what is of interest to the user from the user's profile, and then establish the knowledge base for that category of video. For example, if we know s/he is a CNN fan, we may establish a CNN news knowledge base in off-line based on the characteristics of CNN news frames, such as the CNN logo, a model of the spatial structure of anchor-person shots, the station background when the anchor-person is talking, etc. We may then use these characteristics to differentiate the CNN news video from other videos. Therefore, we see that in on-line video key frame and sequence classifications, prior domain knowledge and learning are very important. And in general, how to establish the base class and feature basis for a knowledge base from scratch within reasonable time duration is one of the most important research issues for on-line video classification. We will address some of the above-mentioned issues in this paper.

Apart from the on-line mode, the video content analysis and annotations are also utilized in the off-line mode, primarily as a means to aid access to the raw data based on sophisticated queries such as those based on semantic content. There are some differences in the raw data processing used to generate these off-line annotations compared to on-line annotations, and one of the biggest differences between the two is that the former one does not have the real-time constraint in creating the annotations. Thus, the off-line annotations might be more sophisticated than those generated using on-line processing. The generation of annotations allows full parsing of a given video record to generate context, which is not possible in on-line techniques. Moreover, there is the issue of indexing records, so as to allow fast access to a set of records, which is again not a consideration in on-line processing. In general, on-line processing to generate annotations has to consider real-time aspects, and the annotation is used to aid filtering, whereas in the off-line mode the annotations are primarily used to allow retrieval from a database at a laser stage, and do not have to be concerned with real-time aspects during metadata extraction.

2.3. Relation work

Today's video database community widely uses low-level features, such as color, motion and texture to index videos. Thus, many existing video database management systems content-based queries based on low-level features. To name a few, Chang et al. [2] used visual cues to facilitate video retrieval, and Deng et al. [3,4] proposed on object-based video representation to facilitate queries on the object. However, the low-level features are generally high-dimensional data and do not directly map to semantic classes, so these methods are still not convenient and efficient enough to support many applications, such as on-line video data searching and filtering based on a user's requests and profiles.

Besides indexing and retrieval with low-level features, researchers [5–7] have also studied video classifications based on low-level features. Efficient video classification can bridge the gap between the video's low-level features and its high-level semantic features, and thus facilitate the indexing of video databases, video summarization and on-line video filtering based on the user's profile. For example, Jaimes and Chang [7] used an interactive learning algorithm over a semantic data model. In this system, they allowed users to specify the classes and provide examples for learning, and the learning algorithm was used to train the classifiers. However, the interactive mode is difficult to apply to on-line applications. Picard [8,9] discussed in detail other data models for video and image libraries, which are mainly based on the classification of digital images. This system only applies to off-line content-based video database retrieval. As for the other applications such as real-time Internet video streaming and on-line video indexing and filtering, generic models for all these videos are hard to establish. It generally requires fast and efficient content analysis and efficient semantic classification. Besides, a specific data model is not effective for generic purposes because of the varied nature of the data. As a result, a generic, fast and efficient framework for on-line video classification still needs to be developed.

Many researchers have also worked on various sports video classification problems. For example, Saur et al. [10] worked on automated analysis and annotation of basketball videos, and they mainly used heuristics of basketball structure to guide the classification. Gong et al. [11] and Sudhir et al. [12] both used model-based classification methods. Similarly, most of these video classification systems are feature-based for off-line applications, and few of them study the relationships between low-level features and high-level conceptual meanings (semantics). In an off-line video classification environment, classification results can be obtained based on all available low-level features. However, this practice is usually not practical for real-time on-line video classification. Hence a fast video classification system based on certain simple, easily extracted low-level features is essential. It would be even better if they system could guide feature extraction low-level features is essential. It would be even better if the system could guide feature extraction to avoid extracting unnecessary features and thus save the valuable computing power and reduce latency for on-line video content analysis.

Knowledge-based techniques [13] are widely used in the development of vision systems, such as image and video segmentation [14]. Knowledge-based systems, which are also known as expert systems, have been traditionally used for the high-level interpretation of images. They incorporate mechanisms for spatial and temporal reasoning that are characteristic of intermediate and high-level image understanding. However, the knowledge-based systems are usually developed for specific applications, to maintain their efficiency. In our work, we are interested in developing a knowledge-based system for fast on-line video classification and filtering applications and a generic framework and methodology will be established first. We use supervised learning to establish any specific knowledge base while maintaining off-line learning/training as a method to establish any knowledge-base for new video types.

Then, we use the basketball video classification and filtering as a particular example to illustrate the concepts proposed.

In our system, the knowledge base consists of a predefined semantic class tree and off-line trained rules that define the if–then rules for each concept class with low-level feature descriptors. Once the rules are learned, on-line fast video classification becomes possible and efficient. We are not going to provide a classifier to classify any video's semantic meanings; rather for some given type of video such as the video type specified in a user's profile list, we provide a generic method to classify on-line video sequences into semantic units according to supervised learned rules. Some semantic video sequences might share the same features with similar features patterns; our goal is to find a generic way to find the most discernible features shared between any two ordered semantic meanings in order to distinguish the two, keeping priority and order in mind.

3. Knowledge-based system for video classification

In this section, we introduce and describe a layered video analysis model for video semantic content analysis and conceptual classification. We will then describe innovative tools for constructing the knowledge-based system with flexible rules covering both semantic content extraction using supervised learning techniques, and ad hoc queries and filtering based on users' requests.

3.1. Layers of video analysis model for video semantic classification

To satisfy both on-line and off-line video analysis requirements and a remedy the shortcomings of traditional content-based database management techniques, semantic inference (classification) and reasoning for conceptual meanings based on low-level perceptual features should be explored in detail. An example would be detecting all video clips containing dunk-like action in a basketball video by exploiting fast or slow movement patterns. To achieve this goal, we describe a generic layered video analysis model as depicted in Fig. 2.

The layered video model consists of: (1) the raw data layer, (2) the video segmentation layer, (3) the perceptual feature content layer, (4) the conceptual content layer and (5) the knowledge layer. Each layer is mapped to a processing module in the on-line video analysis and dissemination infrastructure, which is discussed in Section 5. The function of each layer is detailed below:

- *The raw data layer*: This layer contains original video data, either stored in the video database or received from on-line media streams such as video, audio and caption text from servers over the Internet. It is an abstraction of the coded media sources, including various formats of audio, video, and caption information. When media is queried or matched with a user's profile, audio, video and caption data are multiplexed, and/or synchronized in transmission and then presented to users.
- *The video segmentation layer*: Video is a continuous media and is unconstructed. To understand any content of a video, or to analyze any video data, an efficient mechanism is required to parse the video into smaller units based on certain criteria and the segmented units should be either suitable for perceptual feature analysis or for conceptual abstraction analysis. For example, a video is decomposed into a series of small units with similar visual patterns so that perceptual features, such as color extractions from a key frame can be analyzed effectively and efficiently, while motion patterns need to be extracted from video shots or scenes. As for the conceptual meaning of video, we need to decompose the video into units which correspond to the conceptually meaningful abstract of the video content based on the conceptual model. Our system contains several levels of video content analysis proxies working on various levels of hierarchical video decomposition, which will be discussed in detail in Section 4.
- *The perceptual feature content layer*: Annotation tags are represented in this layer. The annotations are generated by processing either

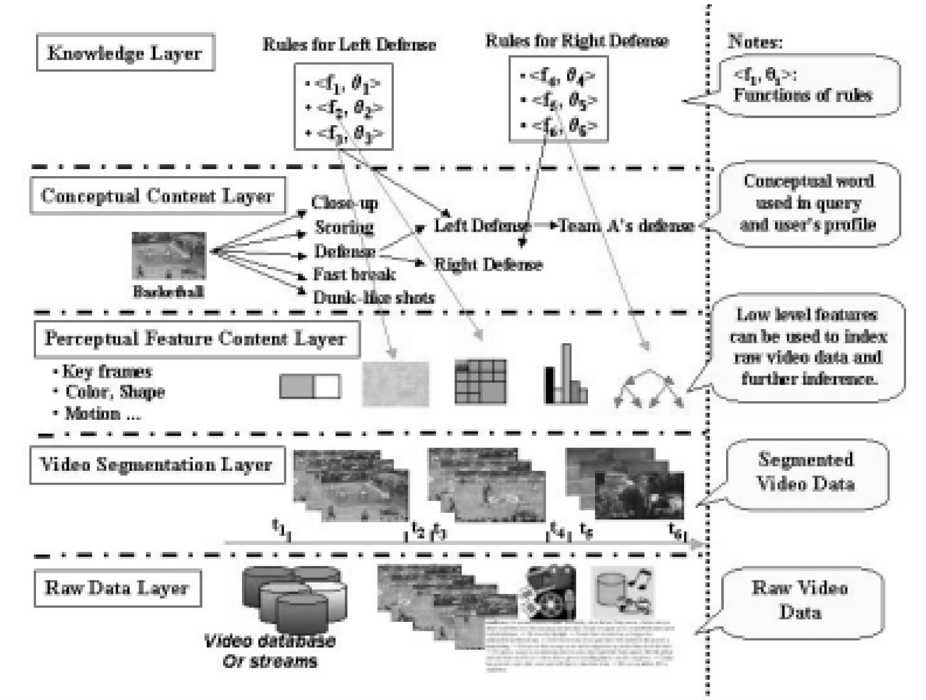


Fig. 2. Layers in the video analysis model.

directly on the raw data or on lower-level annotations. Thus, the perceptual feature content layer contains all the low-level video features extracted from the raw data, including physical video parsing tags and feature descriptors such as color, texture and motion. The extraction methods for the generic low-level features used for system evaluation with the basketball video will be described in detail in Section 4. Correspondingly, our system contains several levels of video low-level analysis and extraction proxies working on various media source, including video, audio and text.

- *The conceptual content layer*: Visual data in a video clip contain rich and unconstructed information. The video conceptual content layer is an abstraction of various visual data semantic types and their structures. In our system, the video conceptual content is mapped to the user's and/or application's model, which the query engine and user's profile will follow. For example, in the basketball video, most people may query certain key game events, such as scores and dunks. The conceptual content layer for the basketball video is defined to have nine events as given in Section 5, the conceptual content of the visual data can be further expanded as needed, depending upon the user's interests and requests.
- *The knowledge layer*: The knowledge layer contains rules to map low-level features in the perceptual feature content layer from each video clip into classes of the conceptual content layer. The knowledge rules are automatically derived from off-line learning algorithms and constructed as a tree structure. The feature attributes used for video classification are general and insensitive to the context. The build-up of knowledge will be described in detail in Section 3.2. Visual entities in the perceptual feature content layer are linked to the knowledge tree in the knowledge content layer to provide present values for conceptual terms. The knowledge content layer to provide preset values for conceptual terms. The knowledge will be used for concept inferencing to

support further data dissemination decision-making. Moreover, the query engine can use the inference engine to automatically generate feature compositions for content-based retrieval with low-level features. High-level semantic content analysis proxies, such as the video feature clustering proxy and the video feature classification proxy, analyze and identify media concepts (such as the subject of a news video or the topic of a story) exhibited by the data streams.

The above conceptual layers are derived by novel feature extraction and content analysis techniques, and are used for on-line media stream filtering and for ad hoc querying based on user interests. Archiving and further off-line analysis of the data can also be performed to provide additional semantic structures for subsequent retrievals.

3.2. Building the knowledge base

In order to classify incoming video streams into meaningful semantic classes, we should classify video with a small number of features that can be easily extracted. Fig. 3 illustrates the overall system for knowledge base building and the on-line video classification process. It consists of two steps. First, it performs the off-line training. Sample video clips of different categories are first identified, and appropriate low-level features are created. Second, we utilize an entropy-based inductive tree-learning algorithm [15] to establish the trained knowledge base. Rules are learned by training, which includes ways to determined characteristic features for each class. The classification rules for each class contain the functions of certain feature descriptors and their corresponding threshold values, order and weight. Off-line training can also update the existing knowledge, if necessary. Once we have the knowledge of the classification rules for each class, they are used to build and guide on-line feature extraction in response to filtering specified by users, i.e. choose the appropriate operators to get target feature descriptors, and output the binary classification result (yes/no) to user's specification. The rules can specify the order, the value and the priority of each feature test. The above process will be described in detail below.

3.2.1. Rules for the knowledge base

Our knowledge base is represented as a decision tree, as shown in Fig. 4, where each node in the tree is an if–then rule applied to a similarity metric based on appropriate low-level features along with well-derived thresholds. The rule is depicted as

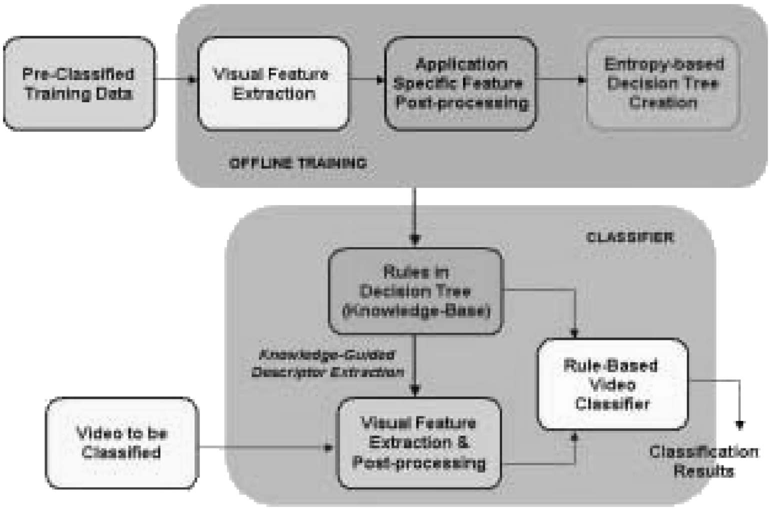

Pre-Classified Training Data
Visual Feature Extraction
Application Specific Feature Post-processing
Entropy-based Decision Tree Creation
OFFLINE TRAINING
CLASSIFIER
Rules in Decision Tree (Knowledge-Base)
Knowledge-Guided Descriptor Extraction
Rule-Based Video Classifier
Video to be Classified
Visual Feature Extraction & Post-processing
Classification Results

Fig. 3. Knowledge-based video classification system.

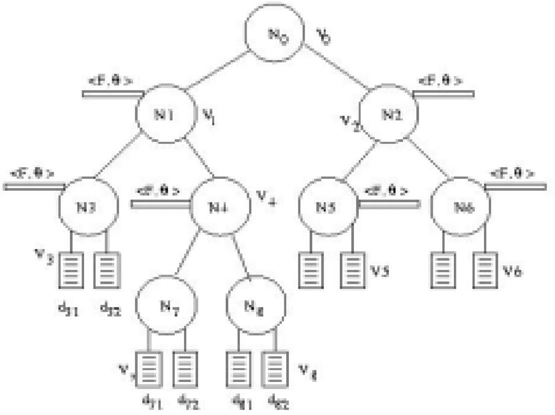


Fig. 4. Illustration of rule-based tree for classification.

$f = \langle F, \theta \rangle$, where F denotes the appropriate feature and θ denotes the threshold which is automatically created during the training process. Semantic categories form leaves of the decision tree. Each node in the tree is either a leaf or an intermediate node with two children. A set of videos is associated with each node N_i, while a decision rule f_i is associated with each intermediate node.

An example of a rule-based tree with nine noes is shown in Fig. 4. The entire set of videos is associated with the root. Let N_I be an intermediate node, with its children labeled as N_{i1} and N_{i2}. Then the video subsets V_I, V_{i1}, V_{i2} satisfy

$$V_i = V_{i1} \cup V_{i2}, V_{i1} \cap V_{i2} = 0. \tag{3.1}$$

In other words, the leaves form a partition of the database into disjoint subsets. In the example above, sets V_3, V_5, V_6, V_7 and V_8 are disjoint, and their union is V_0. Without loss of generality, it is assumed that the decision rule f_I is a discriminate function with the following interpretation. Let x denote a video clip in V_i. If $f_i(x, F_i) \leqslant \theta_i$ then $x \in V_{i1}$. If $f_i(x, F_i) > \theta_i$ then $x \in V_{i2}$. When these discriminant functions represent important visual characteristics, a visual-content rule tree partitions the set of videos into distinct clusters with feature-similar video clips in each cluster. In the example above, all images in V_7 have the following properties in common:

$$F_0(x, F_0) \leqslant \theta_0, \quad f_1(x, F_1) > \theta_1, \quad f_4(x, F_4) \leqslant \theta_4. \tag{3.2}$$

They should look different from the video set in V_2 if the characterization according to the rule of $f_0(x, F_0)$ is visually meaningful. In order words, we need at most rules to classify a video into class N_7.

3.2.2. Supervised decision-tree learning and rules extraction

The classification scheme uses the rule-based system to build a binary tree and associates tree nodes with subclasses of videos. This enables fast video classification. The key computational step is to create the two children of a node so their associated classes are clustered with respect to a meaningful visual characteristic. This implies that video subsets associated with each one of the two child nodes are more alike with respect to the visual property then the video clips associated with the parent node. If the tree is deep enough, its leaves should correspond to clusters of similar video events, and a query by visual-content or classification based on visual features should return one or more of these video clusters.

The decision tree [16] is one of the most widely used and practical methods used to generate rules in inductive inference. Given a collection of S that contains a total of c categories of some target concepts, the *Entropy* of S relative to this c-wise classification is defined as

$$\mathit{Entropy}(S) = \sum_{i=1}^{c} p_i \log_2 p_i, \tag{3.3}$$

where P_I is the proportion of S belonging to class i. The entropy is equal to 0, which is the minimum, when all cases in a set belong to the same class. The entropy is equal to 1, which it the maximum, when each class is equally distributed in the given set. The *Information Gain* is simply the expected reduction in the entropy caused by partitioning examples according to an attribute. More precisely, the information in Gain *Gain(S,A)* of an attribute A relative to a collection of examples S is defined as

$$\mathit{Gain}(S, A) = \mathit{Entropy}(S) - \sum_{v \in \mathit{Values}(A)} \frac{|S_v|}{|S|} \mathit{Entropy}(S_v). \tag{3.4}$$

where *Value*(A) is the Set of all possible values for attribute A, and S_v is the subset of S for which attribute A has value v (i.e. $S_v = \{s \in S | A(s) = v\}$). Note the first term in Eq. (3.4) is just the entropy of the original collection S, and the second term is the expected vale of the entropy after S is partitioned using attribute A. *Gain*(S, A) is the information provided about the target function value, given the value of some other attribute A. The value of *Gain*(S, A) is the number of bits saved when encoding the target value of an arbitrary of S, by knowing the value of attribute A. To determine the first attribute to be tested in the tree, the algorithm determined the *Information Gain* for each candidate attribute (all feature attributes used in the training process), then selects the one with highest information gain. The process of selecting a new attribute and partitioning the training examples is now repeated for each non-terminal descendant node, which uses the training examples associated with that node only. Attributes that have been incorporated higher in the tree are excluded, so that any give attribute can appear at most once along any path through the tree. This process continues for each new leaf node until either of these two conditions is met: (1) every attribute has already been included along this path through the tree, or (2) the training examples associated with this leaf node all have the same target attribute value (i.e. their entropy is zero). In summary, the decision-tree learning procedure has the following steps:

- At each step, we split the tree upon the variable that maximizes the entropy gain. If S contains one or more tuples labeled by C_I and the decision tree is a leaf identifying class C_i, Stop.
- Otherwise, S contains tuples with mixed classification. We split S into $S_1, S_2, \ldots, S_m$ that "tend to belong" to the same class. The split is executed according to possible outcomes $\{O_1, O_2, \ldots, O_m\}$ of a certain feature attribute r_k. Thus, S_I contains all r in S such that $r\_k = O_i$. In this case, the tree for S is a node with m children. The node is labeled with feature attribute r_k and function $f = \langle r_k, O_i \rangle$.
- Perform the above two steps recursively for each $S_1, S_2, \ldots, S_m$.

The whole tree induction depends upon the split criterion and the stop criterion. A good tree should have few levels as it is better to classify with as few decisions as possible; in addition, a good tree should have a large leaf population as it is better to have leaves with as many cases as possible. In the learning algorithm, we split the training data upon the variable that maximizes the *Gain*(S, A); here, the "gain" value is only based on the class distribution, which makes the computation easy to perform, and the "Entropy gain measure" does not take popularity into consideration. If the stop criterion was chosen when the entropy was 0 (same class for all cases), then it will cause over-fitting and yield to very deep trees with few cases on the leaf nodes, which is undesirable. So it is possible to choose the stop criterion either at a minimum popularity allowed for the leaves, or at a certain entropy value to be reached, or even better by combining the previous two conditions. Usually the algorithm will do tree over-fit, first, and then do pruning.

In practice, one quite successful method for finding high-accuracy hypotheses is a technique called *rule post-pruning*. A variant of this pruning method is used by C4.5 [15]. Rule post-pruning involves the following steps:

1. Infer the decision tree from the training set, growing the tree until the training data is fit as well as possible and allowing over fitting to occur.
2. Convert the learned tree into an equivalent set of rules by creating one rule for each path from the root node to a leaf node.
3. Prune (generalize) each rule by removing any preconditions that result in improving its estimated accuracy.
4. Sort the pruned rules by their estimated accuracy, and consider them in this sequence when classifying subsequent instances.

In rule post-pruning, one rule is generated for each leaf node in the tree. Each attribute test along the path from the root to the leaf becomes a rule antecedent (precondition) and the classification at the leaf node becomes a rule consequent (post-condition). Next, each such rule is pruned by removing any antecedent, or precondition, whose removal does not worsen its estimated accuracy. Rule post-pruning would select whichever of the above pruning steps produces the greatest improvement in estimated rule accuracy, then consider pruning the second precondition as a further pruning step. No pruning step is performed if it reduces the estimated rule accuracy.

The 4.5 algorithm [15] evaluates the performance based on the training set itself, using a pessimistic estimate to make up for the fact that the training data gives an estimate biased in favor of the rules. More specifically, C4.5 calculates its pessimistic estimate by calculating the standard deviation in this estimated accuracy assuming a binomial distribution. For a given confidence level, the lower-bound estimate is then taken as the measure of rule performance. For large data sets, this pessimistic estimate is very close to the observed accuracy.

The rules can be generally expressed as follows:

IF (feature 1 = value 1) and (feature 2 = value 2)
THEN Class concept = class 1.

The major advantages of converting a decision tree to rules are:

- Converting to rules allows distinguishing among the different contexts in which a decision node is used.
- Converting to Rules removes the distinction between attribute tests that occur near the root of the tree and those that occur near the leaves.
- Converting to rules improves the readability. Rules are often easier for people to understand.
- Rules are easier to incorporate with a knowledge base for further intelligent inferring and reasoning.

3.3. Video classification procedure for query and filtering

Learned decision trees are constructed with a top-down approach, beginning with the question "Which attribute should be tested at the root of the tree?" The central points of the video classification algorithm are "Which attribute is the most useful in classifying examples?" and "What is a good quantitative measure of the worth of an attribute?" The goal of the algorithm is to find a value which can measure how well a given attribute separates training examples according to their target classification. The properties described above are very important in video classification because, since there are so many possible features to be used as keys to query video/image databases, each feature is not always the best for all queries under different applications. To solve the feature indexing and classification problem efficiently, the study of feature effectiveness for a certain classification application is essential.

The rule tree provides the optimal procedure to find a value that can measure how well a given attribute separates training examples according to their target classification. A new video clip is then classified as follows. Following the tree, the feature to be utilized in Level 1 (the root level) test is first extracted and the corresponding rule is applied. The result of this first test dictates the next feature selection, extraction and test to be followed, and the same procedure will be repeated at each level until the leaf node is reached. In this system, only those features that are relevant are extracted and they are matched with the rule threshold directly. Further processing, such as data indexing, will be made right after the classification is done.

This system of knowledge-based video classification/inference processing is depicted in Fig. 5. It consists of the following three major steps:

- *Preprocessing for feature and feature-extraction section*: Based on the target video specified in a user's profile or query, the system will search for nodes in the rule tree of the knowledge base by traversing up and down tree structure. If a concept key work is identified in the video semantic hierarchy tree, relevant visual/audio features and their thresholds will be selected. At the same time, the feature-extraction operators for the corresponding feature descriptor will be decided.
- *Knowledge-based content matching*: Once each feature-extraction operator obtains the feature value, it is compared with rule. If it matches, then the next feature extraction and matching operators are selected. The procedure is

continued until the leaf node in the binary tree is reached.

- *Video dissemination based on video classification*: The raw video data, including audio and texts, are then filtered and disseminated to the end users depending upon the concept matching between the incoming real-time media streams and the users' requests or profiles.

4. On-line video content extraction and analysis

This section describes the flow chart of video content analysis for on-line video indexing and filtering based on video semantic conceptual content (see Fig. 6). One key technical component integrated into this system is the decomposition of unconstructed and continuous video streams into structured units with both perceptual and conceptual meanings, which are composed of three major components.

Figs. 1 and 7 depict schematic analyses of the video structure and the relationship between video low-level perceptual features and high-level conceptual contents. At first, video data needs to be parsed either based on visual criteria or logic criteria from the top down (see Arrow 1 in Fig. 7). Video is analyzed by segmentation into shots. Then shots can be defined as a set of contiguous frames which depict the same scene or signify a single camera operation or contain a distinct event or action like a significant present or persistence of an object [17]. Scene changes have to be detected when segmenting the video initially. In the meantime, the low-level perceptual features are being extracted for two purposes: one is to serve as content-based indexing to support query and retrieval based on low-level feature similarities; the other is to be used to infer the video's high-level semantic meanings based on the patterns shown in each video concept category. The video segmentation process is followed by shot analysis in order to obtain the final structured video that contains link relations between different shots as well as content features for different shots. While perceptual features need to be captured bottom up (see Arrow 2 in Fig. 7), the video conceptual meanings tend to be associated with larger blocks of video sequences and should be analyzed top down (see Arrow 3 in Fig. 7). Based on the proposed layered video analysis model and the observations given above, hierarchical multi-level

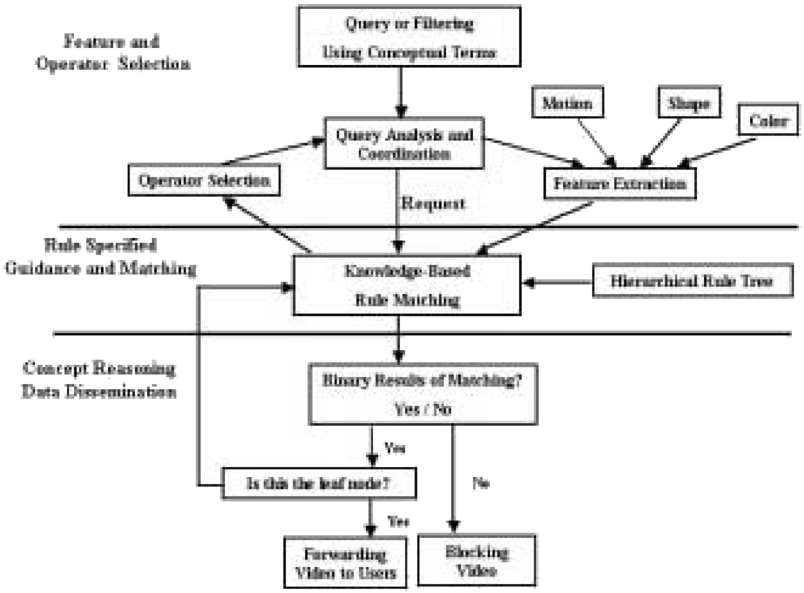


Fig. 5. The flow chart of knowledge-based video classification for query and filtering.

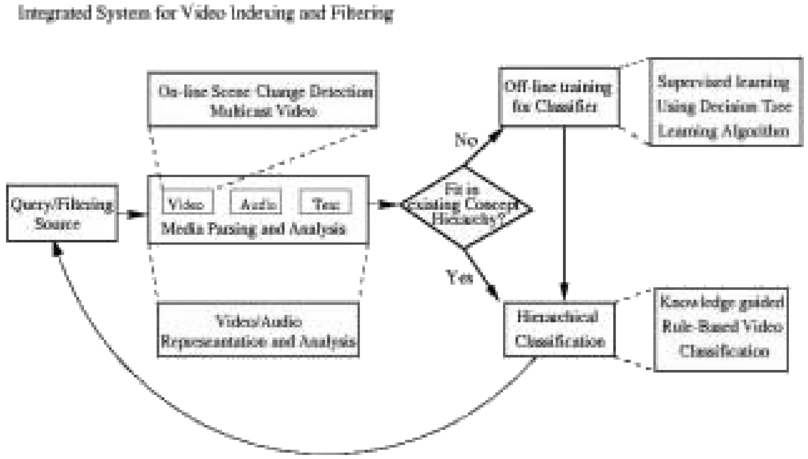


Fig. 6. Flow chart of on-line video content analysis for multimedia information access.

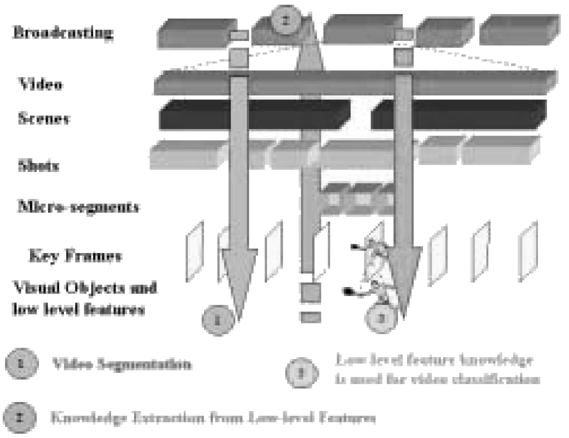


Fig. 7. Video feature-extraction and content analysis structure.

video segmentation and content analysis schemes have been successfully implemented and evaluated in our research work.

We extract low-level video features and knowledge using a bottom-up approach (Arrow 1 in Fig. 7). In this approach, a video is parsed into scenes and the key frames of the video scenes are extracted to represent and summarize the whole video. Then, the key frame images are analyzed by using image processing methods to obtain features such as colors, textures and shapes. Besides, motion and audio features are also analyzed. These low-level features can be used to index video databases or to link key frames. The main advantage of using low-level features in video database indexing is that database organization can be completely automated. At the same time, its main disadvantages it that it is very difficult to find the description of an image/video, which is close to the semantic description of its contents. However, conceptually similar video clips generally share common perceptual patterns, and this provides the foundation of video classification for conceptual meanings. Here, we used a supervised learning method to generate functions to bridge the two.

High-level video meanings are analyzed by classifying video contents with a top-down approach (see Arrow 2 in Fig. 7) as described in previous sections. Since a video may have a wide range of contents, effective classification must be done in a hierarchical way. The advantage of using a semantic description of video is that it is possible to give very detailed and semantically precise descriptions of an image/video including terms which are unlikely to be determined by using video or image processing techniques alone. The disadvantage of this technique is that as yet there are no efficient and fast methods for fast video/image classification. To overcome this obstacle and to organize the multimedia database more suitably for human use, we propose a rule-based classification system by supervised learning, and a hierarchical video conceptual model specified in terms

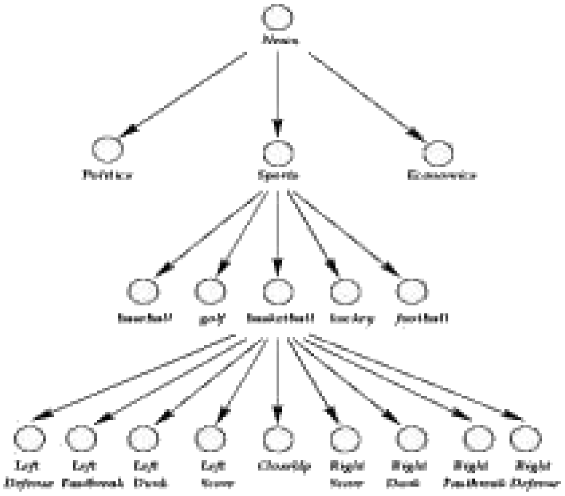


Fig. 8. Example of hierarchical concept tree.

of the video concept tree which Fig. 8 shows an example. Furthermore, novel machine learning tools are developed to establish relations between the low-level perceptual features and high-level conceptual video contents. The rule-based knowledge representation is unique but general enough to be used on a variety of problems in different application domains.

4.1. Real-time scene-change detection and key frame extraction

Scene-change detection is very important, since it is the first step and the most fundamental element for video analysis. We have developed a content proxy that implements a scheme for summarizing real-time video in terms of a few representative key frames from video sequences by the scene-change detection process. The processing scheme is shown in Fig. 9. The main component of this scheme is based on real-time scene-change detection on RTP intra-H.261 compressed video streams that are commonly used in MBone broadcasts. A video is split into meaningful scenes such that each image frame corresponds to a different shot detected out as a key one based on a particular criterion. Fig. 9 shows a scene-change detection based on histogram comparisons between adjacent frames of a video stream. Therefore, a video clip can then be summarized via key frames extracted from different scenes that make up the whole stream. We take into account changes in both luminance and chrominance values in making the decision. If the change in the luminance or chrominance histogram over successive frames is larger than the threshold, we categorize that frame as a scene-change frame. As the result, a key frame is generated from the stream and is sent out as a scene in a multicase channel. To avoid unclear images caused by editing effects such as dissolve, we choose the frame that sits at the end of the first 10% of video sequences right after a new scene has been detected, as the key frame of the scene. The scene-change tag and key frame can also be inserted into a multicase channel so that other relevant proxies that are attempting to do other kinds of processing on the network can utilize this extracted information.

We have developed a family of algorithms, including one that utilizes full decompression to get full frames and a partial decompression to get information on the changed blocks so as to estimate the extent to which the full frame has changed since the last frame. In our earlier investigations, joint algorithms based on video codec characteristics were carried out to acquire fast and accurate scene detection. Experiment results shows that our algorithms are capable of supporting real-time video processing and satisfying on-line annotation needs [18,19] given different networks and data characteristics. We also adopt the scene-change detection algorithm in the compressed domain proposed by Yeo and Liu [20] for MPEG-1 video. A good survey of technologies for parsing digital video was given by Ahangera and Little [21].

4.2. Video events segmentation with visual and audio cues

Our segmentation process for sports events is shown in Fig. 10. To segment sports into logical units such as events, we employed a heuristic rule that can detect sports event boundary by identifying the sound of a whistle or the change associated with a new scene. Since a whistle sound from a

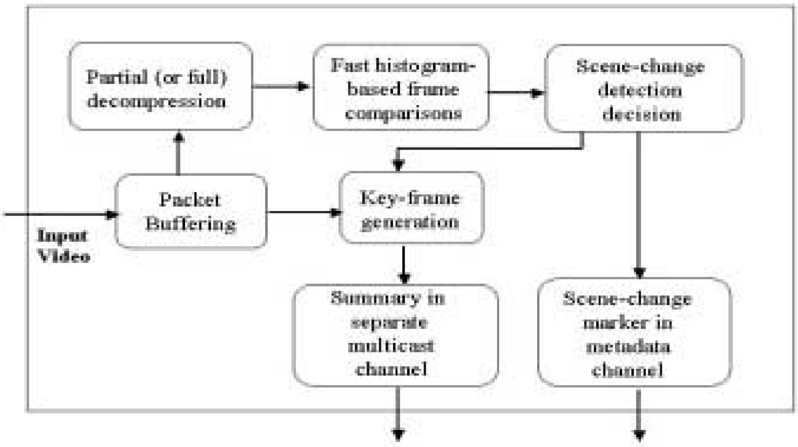

Partial (or full) decompression
Fast histogram-based frame comparisons
Scene-change detection decision
Input Video
Packet Buffering
Key-frame generation
Summary in separate multicast channel
Scene-change marker in metadata channel

Fig. 9. On-line scene-change detection for H.261 video.

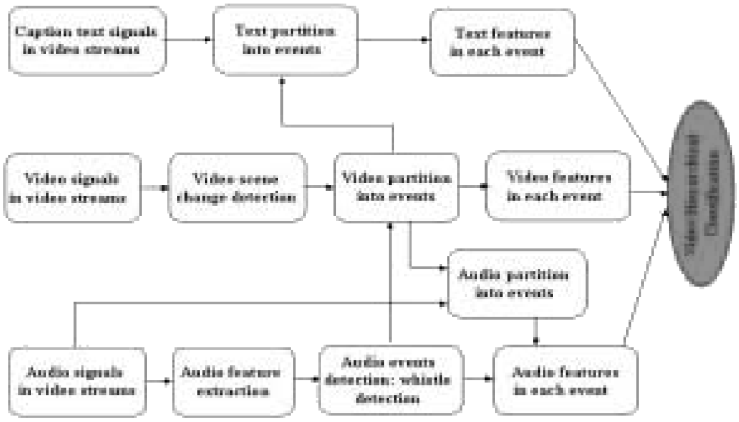

Caption text signals in video streams
Text partition into events
Text features in each event
Video signals in video streams
Video scene change detection
Video partition into events
Video features in each event
Audio partition into events
Audio signals in video streams
Audio feature extraction
Audio events detection: whistle detection
Audio features in each event

Fig. 10. Sports event segmentation flow chart.

referee usually indicates the start or end of an event at games, we treat it as the logic boundary break for a new event even if there is no scene change. This audio feature analysis and whistle sound detection will be discussed in the following subsections.

4.3. Feature extraction and analysis for audio cues

Generally, audio features are divided into two distinct categories, time and frequency domain. To extract both domain features, we first sample audio signals at 11 o25 Hz with 16 bits/sample. Then for each of the audio clips corresponding to a distinct visual scene, we calculate audio features as follows.

4.3.1. Time-domain features

In the time domain, we calculate statistical parameters (such as mean, standard deviation, and dynamic range, etc.) of the trajectories of short-time audio volume and zero-crossing rate for each audio clip, Non-silence ratios are also calculated based on both short-time volume and zero-crossing rate. The short-time audio volume and the zero-crossing rate are defined in Eqs. (4.1) and (4.2), respectively.

$$V_n = \sqrt{\frac{1}{N}\sum\nolimits_m [x(m)w(n-m)]^2}, \tag{4.1}$$

$$Z_n = \frac{1}{2}\sum |\mathrm{sgn}[x(m)] - \mathrm{sgn}[x(m-1)]|w(n-m), \tag{4.2}$$

where for both equations,

$$w(n) = 1 \quad \text{when } 0 \leqslant n \leqslant N-1 \text{ otherwise } w(n) = 0.$$

And for Eq. (4.2) only

$$\mathrm{sgn}[x(m)] = 1 \quad \text{when } x(m) \geqslant 0 \text{ while } \mathrm{sgn}[x(m)] = -1 \text{ when } x(m) < 0.$$

In both the above equations, $x(m)$ is the discrete time audio signal with index of m, n is the time index of the short audio frame whose size is specified by a rectangular window of $w(n)$ with window length N. Here, we choose the frame size of $N = 150$ samples (i.e. the audio frame is about 15 ms long) and calculate both features once every 100 samples (about 10 ms apart) in the audio clips. The statistical parameters of the above two features, such as mean and variance, are calculated based on index n. Since the dynamic ranges of these statistical features differ a lot, we normalize them by their maximum volume and maximum zero-crossing rate, respectively, for each audio clop. To detect the silence ratio in each clip, we compare the volume and the zero-crossing rate with a certain threshold for each. It is claimed that the audio frame is silent when both its volume and zero-crossing rate with a certain threshold for each. It is claimed that the audio frame is silent when both its volume and zero-crossing rate are smaller than each of its thresholds. Thus, the non-zero ratios of V_n and Z_n are the percentage of non-silent audio frames over the whole audio clip.

4.3.2. *Frequency-domain features*

In the frequency domain, we first calculate the spectrum of an audio clip by using a direct FFT transform with a 512-point FFT size which generates a 2-D plot of the short-time Fourier transform (over each audio frame) with frequency as the X-axis and the amplitude in db as the Y-axis for all audio frames over the time domain. We compute the following features for each frame and their distributions over the entire audio clip:

- *Short-time fundamental frequency (FuF)*: The FuF is defined as follows. When the sound is harmonic, the FuF value is equal to the fundamental frequency estimated from the audio signal; and when the sound is non-harmonic, the Fuf is set to zero. We calculate each frame's FuF as described by Zhang and Kuo [22].
- *FuF distribution for the whole audio clip*: Statistical parameters, such as mean and variance, are computed for the trajectory of the FuF of each audio frame over the time through the entire audio clip.
- *Centroid frequency and bandwidth*: Similar to the work of Wold et al. [23], the frequency centroid, $C(i)$, and bandwidth $B(i)$, of each audio frame are defined as

$$C(i) = \frac{\int_0^\infty \omega |S_i(\omega)|^2 \,\mathrm{d}\omega}{\int_0^\infty |S_i(\omega)|^2 \,\mathrm{d}\omega}, \tag{4.3}$$

$$b(i) = \sqrt{\frac{\int_0^\infty (\omega - C(i))^2 |S_i(\omega)|^2 \,\mathrm{d}\omega}{\int_0^\infty |S_i(\omega)|^2 \,\mathrm{d}\omega}}, \tag{4.4}$$

where $S_i(\omega)$ represents the short-time Fourier transform of the ith frame. Using Eqs. (4.3) and (4.4), we compute the centroid and the bandwidth for every audio frame over the entire audio clip, thus generating 3-D plots of the centroids and the bandwidths of audio clips along the time axis. The mean and the standard deviation of both the centroid and the bandwidth of an audio clip are used as four frequency domain features:

- Energy ratio of some sensitive sub-bands. The energy distribution in different frequency bands varies quite significantly among different audio signals. For example, the spectral peak tracks in speech normally lie in the lower frequency bands, ranging from 100–300 Hz; while whistles, which are often heard in sports videos, have high frequencies and strong spectrum energy with frequencies ranging from 3500–

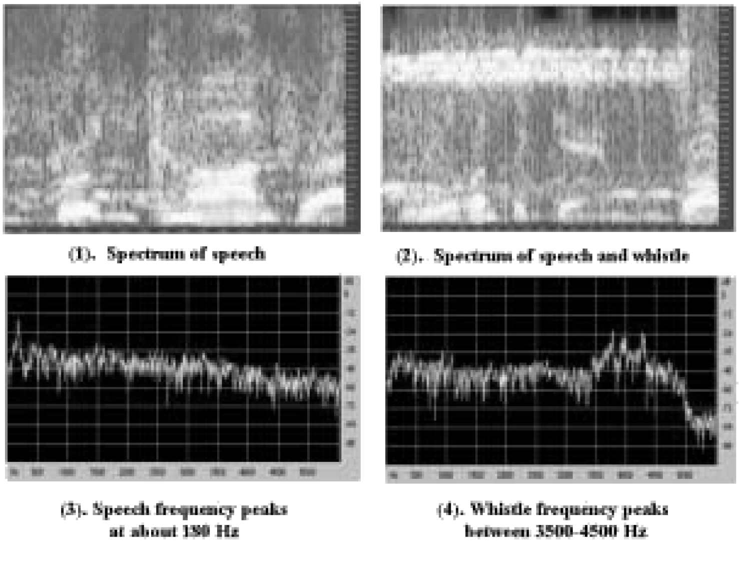
4500 Hz. To differentiate special audio events, like speech, whistles or noise, we calculate energy ratios in the sub-bands [0–400 Hz], [400–1720 Hz], [1800–3500 Hz], [3500–4500 Hz] with respect to the overall energy for all frames in the audio clip.

- Peak of spectrum on each audio frame and peak distribution of the entire audio clip. The peak track in the spectrum of an audio signal often provides us with some characteristics property of the sound. In the sports video games, one typical sound is a referee's whistle, which often occurs right after fouls in basketball and soccer, or at the beginning of a serve in volleyball, etc. Whistles in sports videos often last at least 1 s, and have stronger energy then speech and music. We link whistle detection to both video semantic boundary detection and semantic meaning inference. Peaks of whistle spectrums normally range from 3500 to 4500 Hz. We detect the most prominent frequency from FFT transformed spectrums for every frame in an audio clip so as to get a 2-D graph with the time domain on the X-axis. It is claimed that a sound of whistle is detected if there is a longer than 1 s window of peak frequencies which fall into the range between 3500 and 4500 Hz, as shown in Fig. 11.

4.4. Motion feature extraction and analysis

Motion information is a good cue to use in video [24], as it is an integral part of a motion sequence. In addition, motion is typically already calculated in most video codecs, and motion compensation information is available in the compressed data stream. In MPEG-1 video, there are three types of frames, I frame, B frame and P frame (see Fig. 12). To find the motion patterns for certain videos, we focus on the direction and magnitude of the video sequences motion flows of P frames only. The reason is that P frames give only forward prediction, and the information is useful for us in order to calculate the direction of the "flow" of that video clip. Motion information is specifically important to us as we are focusing on sports videos where the "motion flow" is a significant cue.

We did not track the object and extract objection motion information directly as this approach is typically computationally complex and time consuming. Instead we tried to use some statistical motion descriptors to see if they could satisfy our requirement. We calculated two such statistical features, dominant motion direction and the motion magnitude of the motion vectors in that clip.



(1). Spectrum of speech

(2). Spectrum of speech and whistle

(3). Speech frequency peaks at about 180 Hz

(4). Whistle frequency peaks between 3500-4500 Hz

Fig. 11. Whistle sound detection by using audio spectrum features.

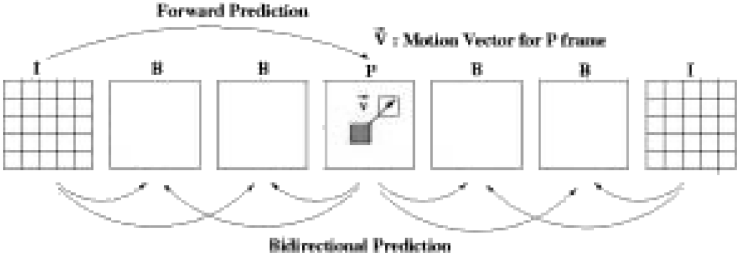
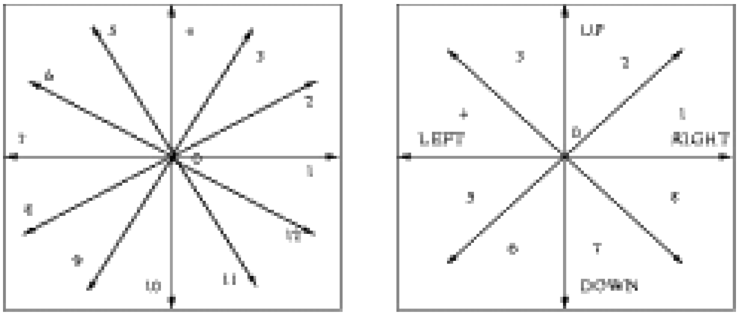


Fig. 12. Forward prediction and bi-directional prediction for MPEG-1 video.



(a) (b)

Fig. 13. Motion directions extraction.

4.4.1.1. Direction of motion descriptor extraction. For each of the motion vectors in the vector image, we can cluster, and then classify each motion vector's direction according to Fig. 13. We cluster the vectors first to Fig. 13(b); the criteria is as follows: the vectors in Regions 1 and 8 are termed as RIGHT; the vectors in the Regions 2 and 3 as UP; the vectors in Regions 4 and 5 as LEFT; and the vectors in Regions 6 and 7 as DOWN. We then calculate the amount of motion along each direction by counting the total number of vectors along that direction in each class for the whole video sequence. This results in a 4-D motion direction vector.

4.4.1.2. Motion magnitude calculation. Normally the magnitude of the motion is also encoded in the motion vector. To get the instances of magnitude and speed of the motion descriptors along both X and Y directions, we calculated the motion magnitude of the whole frame according to Eq. (4.5).

$$x_{ave}(i,j) = \frac{\sum_{i=0}^{n-1} x_i}{n}, \quad y_{ave}(i,j) = \frac{\sum_{j=0}^{m-1} y_i}{m}, \qquad (4.5)$$

where n and m are the total number of motion vectors in the frame with respect to the X direction and Y direction, respectively. We also calculated average and biggest motion magnitude along the X and Y directions for the whole video sequence.

4.4.2. Color features

In addition to motion, color and edge information also play an important role in object identification. In particular, we wanted to use color and edge information to classify a given video clip key frame into four categories such as left court, right court, middle court and so on, as shown in Figs. 14–16.

To achieve this goal, a key frame was first extracted for every newly detected scene. To avoid unclear images caused by editing effects, such as dissolve, we chose the frame which marks the end of the first 10% of video sequences after a new scene has been detected as the key frame by using histogram scene-change detection algorithm. We then extracted color information such as color histogram, dominant color and regional color information for each key frame. YV histograms are automatically generated by the shot change detection agents. The color histogram and the dominant color orientation histogram [25] are statistical visual cues. They do not contain the spatial information. For example, some visually different key frames might have exactly the same color distribution, but different localized colors. Thus, to get more detailed color information and increase the differentiating power of the color feature, we considered the dominant color and the localized color information. We used the median-cut algorithm [26] to reduce the color map to about 256 colors. That is, colors in a image were mapped to their closest match in the new color map so that the colors of the original images were mapped to their closest match in the new color map so that the colors of the original images were clustered. We used the clustering method to automatically detect a number of dominant colors and output them as a color tree for each frame. Then, all pixels were back mapped into homogeneous regions, if the distance to a dominant color was not bigger than a given threshold, to get regions corresponding to the first five maximum dominant colors. Once we got the region for each dominant color. The centroid and the region boundary were calculated as the attribute values of the regional color.

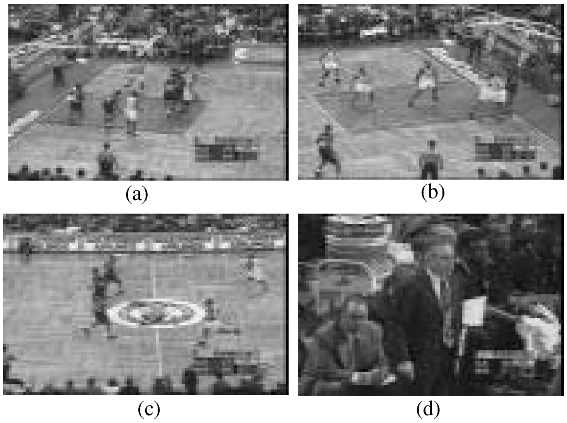


Fig. 14. Various key frames of basket ball scenes (a) left court, (b) right court, (c) middle court, and (d) close-ups.

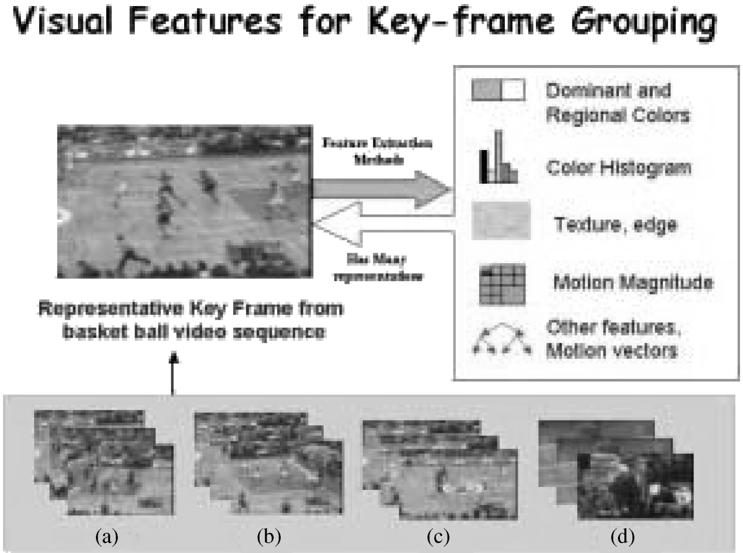

Visual Features for Key-frame Grouping
Feature Extraction Methods
Bin Map representation
Representative Key Frame from basket ball video sequence
Dominant and Regional Colors
Color Histogram
Texture, edge
Motion Magnitude
Other features, Motion vectors
(a)
(b)
(c)
(d)

Fig. 15. Features to cluster key frames into: (a) left court, (b) right court, (c) middle court, and (d) close-up.

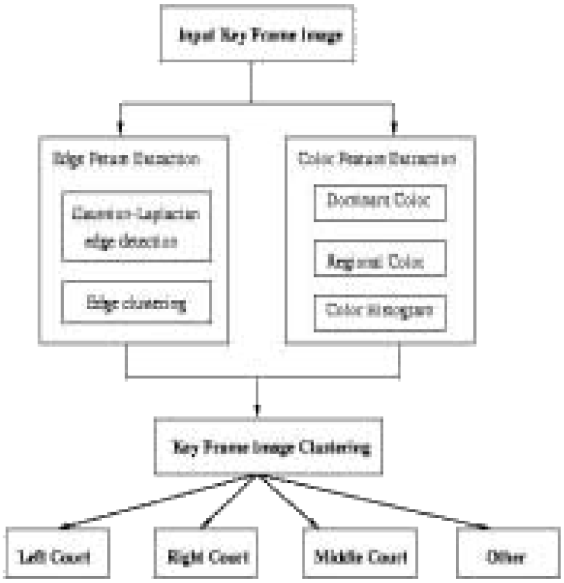


Fig. 16. Feature extraction for key frame clustering.

4.4.3. Edge detection and analysis

Changes or discontinuities in an image amplitude attribute such as the luminance or the tristimulus value are fundamental characteristics of an image since they often provide an indication of the physical extent of object within the image. Edges characterize object boundaries and they can be used in image segmentation, registration, identification and representation of an object in scenes. Also, edge patterns can be used to analyze key characteristics and scene classification. For example, in boxing sports, there are generally fence-like edges which contain more lines of horizontal edges than for soccer videos. Edge patterns can also be treated as a simple texture in each key frame image.

Here, we used a gradient edge operator to detect edges and take edge detection masks to fulfil the edge detection task. Edge detection masks included the first-order derivative masks, such as the prewitt and the robinson three-level masks. Edge detection with an option of the second-order Laplacian mask was also provided. In the first-order derivative masks, all eight directional masks were used to detect out edges along different directions. Edge densities along different directions and different lengths around the dominant color region were calculated as edge features. Furthermore, we analyzed the edge information along four directions by calculating the distribution of the visible edges and clustering them into horizontal, vertical and other category edge types. In the meantime, each key frame of the video clip was clustered into the categories shown in Figs. 14 and 15. The flow chart for key frame clustering is shown in Fig. 16. In the experiment, the left court has regional blue color in the middle of the left half region; the edge around the region is horizontal and vertical. By contrast, while the middle field does not have a blue dominant color, it has a vertical edge in the middle.

4.5. Video key frame clustering

Once all low-level visual proxies have generated visual feature values, clustering proxies can group similar low-level features to identify key frames. For the basketball video, we used the color and edge information to classify a key frame into four categories such as the left court, the right court, the middle court and others, as shown in Fig. 14–16. In the experiment, the left court has a regional green color in the middle of the left half region and the edge around the region was horizontal and vertical. The middle field did not have the green dominant color. Instead, it has a vertical edge in the middle.

5. Evaluation of the proposed system

5.1. System architecture

According to our layered video analysis model, we proposed and implement a system prototype for a content-aware and user-preference-oriented multimedia data distribution system over the Internet based on the multicast protocol. Fig. 17 shows an infrastructure prototype to support on-line media content analysis and present a model of service that realizes the goal of providing effective on-line content-based media dissemination by filtering. More specifically, we develop a real-time intelligent multimedia system prototype consisting

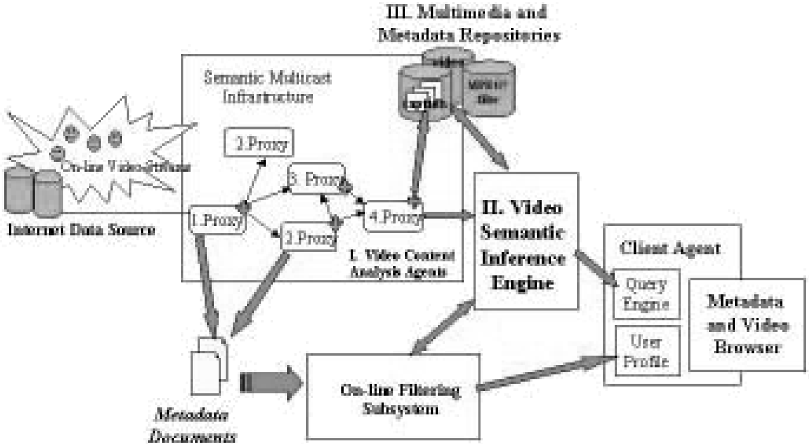


Fig. 17. The proposed on-line video stream analysis and dissemination infrastructure.

of coordinated proxies for fast video content analysis and dissemination over the Internet based on users' profiles. One of its applications is to manage contents of real-time collaborative sessions which generate various types of data streams, primarily raw audio, video, and graphics data, along with application-specific data types. As part of its operation, this system aims to analyze input streams based on users' profiles, enrich the raw data streams with semantic description tags, and create knowledge rules to capture high-level conceptual meanings. Then, it will use the created knowledge-based system to support filtering and ad hoc queries of these data streams. However, a number of multimedia data streams in their raw forms are not amenable to automated semantic interpretation, and typically will have to be enhanced with other features, which are either manually created/attached or are extracted by analyzing the raw data in off-line mode. Our system provides a set of intelligent tools to solve this challenging task.

As shown in Fig. 17, the system will provide on-line feature extraction, multimedia stream content understanding and organization, and data filtering by matching with user profiles for real-time media distribution and sharing over the Internet. This system is expected to provide synchronized multimedia data stream distribution and filtering. In addition, the system will attempt to organize multimedia resources over the Internet in a scalable way, allowing users to find items related to their interest based on the content of the data. The system can also be extended to applications such as interactive and personalize TV broadcast services, personalized web services and training services and collaborative applications. The whole system is composed of the following three modules.

1. *Content agents/proxies*: To meet the goal of fast on-line multimedia information access, every stream must be transmitted in real time or near real time, and quick content analysis and annotations are required to be properly classified and disseminated by the architecture. Most of the functionalities for content extraction, content analysis and data filtering/redistribution as per user interests/query are fulfilled by intelligent content agents residing at proper Internet nodes. In this architecture, the agent or proxy is an active software module that can be placed throughout the network grid to perform various operations needed for on-line multimedia data processing. Typically, the order of executed for a set of proxies depends on a particular request. For example, after an annotation proxy generates annotations of a stream, a filtering proxy will perform matching functions based on the learned knowledge and users' profiles, to properly re-route or cut off the stream as necessary, and a transcoding

proxy (and/or a summarization proxy) will transform the raw data to adapt to low bandwidth networks. Note that it is possible to combine some of these operations into a single proxy. For example, the same proxy can perform both annotations and filtering. Furthermore, annotations extracted by content agents can also support indexing and content-based retrieval.

In the on-line video content analysis infrastructure as shown in Fig. 17, each semantic multicase content proxy consists of modules to do one or more specific data processing operations and each runs as a daemon. The video content analysis proxies are centered to the hierarchical decomposition of video data, and extract visual/audio/motional characteristic contents by combining and/or coordinating video semantic class inference engines. High-level video semantics are inferred from low-level features for the filtering purpose. Extracted features and semantic content can further serve as annotation or indexing for off-line database management. Some of the functions, performed by different proxies, are described below.

- ○ *Annotation*: The creation of a high-level annotation tag for an information stream is an important form of content enrichment and is essential to effective information dissemination in semantic multicast. As agent may generate tags on session sub-streams (e.g. timestamp, or concept) to prepare for archival and filtering. In their raw form, multimedia data types such as video and audio are not amenable to automated semantic interpretation and typically have to be enhanced with higher-level features such as keywords, video scene-change tags, and representative sample frames. For example, and audio classifier can classify the audio signals into categories, such as speech, noise or whistle. And speech-understanding systems can automatically transcribe the audio stream in order to create a text of the spoken words that can be utilized to allows the creation of a time-aligned transcript of the spoken words contained in the audio stream. At the next level, natural language processing techniques can be applied to correct and summarize the transcript as well as to identify key words that will describe logical sub-units of the entire session, as defined by the video segmentation operation.
- ○ *Filtering*: An agent may subset a session based on annotations to reduce the scope to the interests of a particular group. Such filtering is generally time constrained to minimize the latency incurred in the delivery of filtered information to users. In our filtering algorithms, we come with a knowledge base which can accommodate the media low-level feature descriptor, plus description schemes to facilitate the filtering. Each feature descriptor has its own specific definition and extraction operator.
- ○ *Archival*: An agent may store "appropriate" sub-sessions in an associated multimedia archive. As the agent archives the stream, it performs a more detained and off-line analysis to provide additional semantic structuring and indexing for subsequent retrieval and feedback to the semantic multicast graph.
- ○ *Temporal synchronization of content with descriptions*: To allow the temporal association of descriptions with content (AV objects) that can vary over time and effective media stream consumption, we use timestamps of RTP media streams as a synchronization connection between various media streams.
- ○ *Synthesis of multiple low-level features associated with a content item*: An agent will allow flexible localization of descriptor data with one or more content objects. A variety of descriptors and description schemes could be associated with each content item. Depending on the user's profile, not all of these will be used in all cases. In push applications (e.g. real-time multicast/broadcast), effective feature synthesis and data multiplexing are needed to satisfy various content requests from users.

- ○ *Buffer management of further decisions and actions*: Local storage and buffering management is essential for real-time applications, as every step of content analysis or decision making generates latency.

2. *Video semantic inference engine*: However, as we discussed in previous sections, low-level contents have limits in their ability to support more advanced information access demands, such as on-line filtering based on semantic contents. To overcome this limitation, agents may archive information streams and perform more detailed analysis on the data to provide additional semantic structures for subsequent filtering and retrieval. The video semantic inference engine consists of the rule-based knowledge base for the video classification subsystem. A service assigner manages and coordinates the content agents for on-line perceptual and semantic content analysis other than filtering and query procedures.
3. *Multimedia repository*: A multimedia repository will store the data streams and any annotations and transformations made by the content agents, such as feature extraction proxies. As we discussed, video content is analyzed in both perceptual and conceptual aspects; and multimedia streams, especially video, can be organized and stored with indexings pointing to the continuous streams with both perceptual and conceptual features. A smart interface to the multimedia repository provides the tools for off-line data searching, retrieval and browsing.

5.2. On-line basketball video semantic classification for filtering and indexing

5.2.1. Basketball video classification rules and experimental results

The proposed knowledge- and rule-base video classification system is shown in Figs. 3 and 5. Knowledge for video classification is trained off-line first. In our experiment, sample video clips of the different categories were first identified and appropriate low-level features were created. We then utilized an entropy-based inductive tree-learning algorithm [16] to establish the trained knowledge base of the video data type. This knowledge base is represented as a decision tree with each node in the tree being an if–then rule as applied to a similarity metric utilizing an ‘‘appropriate’’ low-level feature along with a good ‘‘derived threshold:. The rule scheme for basketball is shown in Fig. 18, where the rule at each level is depicted as $\langle F, \theta \rangle$. Note that the appropriate feature F and a good threshold θ are automatically created by the training process. Note also that the semantic categories into which

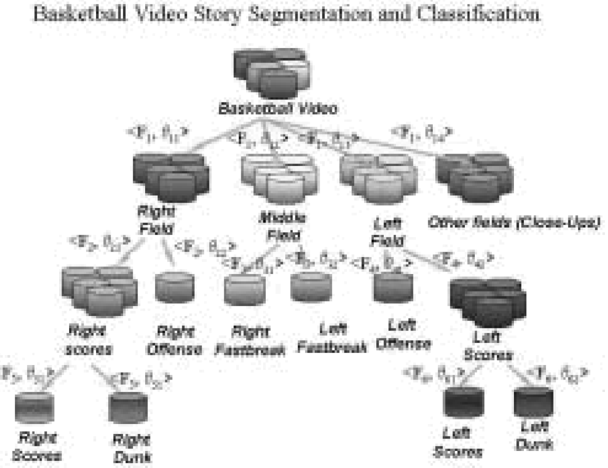


Fig. 18. Rule tree for basketball video classification.

the video sequences will be classified form the leaves of the tree. This rule-based classifier begins with the question "Which attribute should be tested at the root of the tree?; and we aim for the attributes which are the most useful for classifying the examples. Next we may ask "What is a good quantitative measure of the worth of an attribute?" and the tree provides the optimal procedure to find a value that can measure how well a given attribute separates the training examples according to their target classification. A new video clip is then classified as follows: following the tree, the feature which was utilized at Level 1 (the root level) I first extracted and the corresponding rule is applied, following which the path selected is chosen. At the next level, the same step is carried out whereby an appropriate feature is selected and the corresponding rule applied. In this system, only the relevant features are extracted and they are matched with the rule threshold directly. Further processing, such as data indexing, is made right after the classification is done.

We applied our system (Figs. 3 and 18) to basketball videos by using on-line classifying and filtering basketball into nine major meaningful events. They are:

1. Team offense at the left court;
2. Team offense at the right court;
3. fastbreak to the left;
4. fastbreak to the right;
5. dunk-like in the left court;
6. dunk-like in the right court;
7. scoring in the left court;
8. scoring in the right court; and
9. close-ups for audience or players.

We applied the proposed system to 157 basketball video clips segmented from a basketball game for training. After training, we arrived at an at most three-level decision tree that contains 14 rules, as shown in Fig. 15. Note that in the classification stage, at most we have to do three calculations for each class, as that is the level of the tree. No more than six features are needed to classify all nine basketball events. We used a set of basketball video data from one game to train the learning algorithm to get the nine classes' critical patterns and classifying rules that are of the differentiating powers. From the rule tree, we see that using the descriptor of key frame type alone, the program can judge weather the video sequence is a close-up or not. To discern right to left fastbreak, first, we need to judge key-frame type, and then judge the dominant motion direction and following that judge the average magnitude of the motion vector component along the X-axis. In other words, only key frame type and motion direction and average magnitude for the X-axis are relevant for right and left fastbreak classes and thus these features are suitable for fastbreak event threshold for each specific basketball event, which is especially useful for on-line user profile filtering.

By applying the learned rules to classifying a new set of 110 basketball game video clips, we reached a classification accuracy of from 70% to 85.7% for the above nine identified basketball events for the nine basketball classes as shown in Table 1. Here accuracy is defined as

$$Accuracy = \frac{\#corrected}{\#corrected + \#false - alarmed}.$$

5.2.2. Content agent coordination for filtering and indexing

The proposed rule-based video classification system is good for both on-line and off-line video classifications, which are applicable to video indexing systems, video scene understanding and mining, on-line video filtering and video intelligent summarization, and fast video browsing and so

Table 1
Results of basketball video classification by rules-based classification system

Class	Training sample	Testing sample	Accuracy (%)
Left offense	20	14	75.8
Right offense	22	14	85.7
Left fastbreak	20	14	78.5
Right fastbreak	21	15	80.0
Left scores	15	12	75.0
Right scores	17	10	70.0
Left dunk	12	10	70.0
Right dunk	10	9	77.8
Close-up scenes	20	12	75.0
Total	157	110	78.2

on. General video classification problems can easily follow the system prototype illustrated in Figs. 3 and 5. Once we have learned the knowledge to classify each basket ball event, the rules are used to build on-line feature extraction in response to both archiving purposes and user-specified filtering purposes.

It is straightforward to apply such a rule-based video classification system to on-line user-specified video filtering. Fig. 19 illustrates the data flow for such applications. For any user-specified video category, the knowledge base contains the corresponding characteristic features and rules to identify it. Only those relevant features are extracted on-line and they are matched with rule threshold. If they satisfy the rules, then the real-time stream matches with the user's expectation; otherwise, it does not, and a further decision based on this intelligent video classification will be made based on the application. For example, to classify left-scoring events, we need to first segment the video into small units, and then extract key frames and low-level features for clustering. It takes multiple cooperating agents to realize final results for video semantic classification, indexing and dissemination, as in shown in Fig. 20.

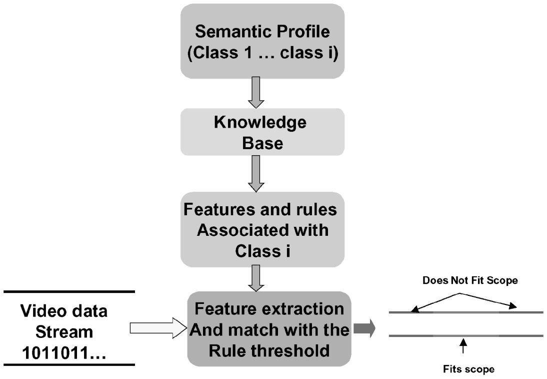

Semantic Profile (Class 1 … class i)
Knowledge Base
Features and rules Associated with Class i
Video data Stream 1011011…
Feature extraction And match with the Rule threshold
Does Not Fit Scope
Fits scope

Fig. 19. Flow chart for on-line video filtering based on user's profile.

In the current system design, the negotiation of agent service is realized by requiring each proxy to implement an "applicability" function that captures the behavior and capability of the agent and exposes it to the semantic multicase framework. However, an agent often can process the input data streams but not necessarily transform them into a form that satisfies the target group request. In this case, while the agent cannot by itself completely service the data-processing needs, it may still "partially" transform the input data streams into others that can be further processed by other agents to satisfy the target group request. As a result, the applicability function is defined to return: (i) an intermediate group request that the agent can process the input data streams into, (ii) the specification of the intermediate data streams, and similar to the last case, and (iii) the configuration parameters for the agent to perform the operation. The intermediate group request and data streams can then be treated as a new source request and data stream, and along with the target group request, get passed to other agents to determine if they can complete the mapping from the link between the source and target group requests to a series of agent instances.

The characteristic features include low-level features of video, such as key frames, color histograms, dominant colors and regional colors. For any incoming on-line key frame, video feature-extraction agents extract low-level features fast, using the same feature descriptor and algorithm as those specified by the knowledge base. A key frame classification proxy checks if the new

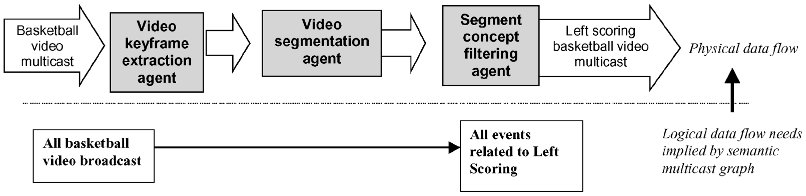

Basketball video multicast
Video keyframe extraction agent
Video segmentation agent
Segment concept filtering agent
Left scoring basketball video multicast
Physical data flow
All basketball video broadcast
All events related to Left Scoring
Logical data flow needs implied by semantic multicast graph

Fig. 20. Data flows in content analysis specified by rules: agents cooperating to realize video semantic classification for indexing and dessemination.

key frame's features are matched with those in the knowledge base and a binary decision (yes/no) for each key frame semantic category will be output. This is of great use in the sense that once we can make this distinction, we can (in the case of a basketball-interested user, say) either pass the video to the user if the user requires basketball video, or we may use the key frame of the basket ball video as the basket ball event boundary for finer semantics extraction from the video classification procedure, if his or her request is more specific.

Fig. 20 shows how the data streams carrying "all basketball video" may be processed by a series of semantic multicase agents to produce data streams carrying "Left-scoring-related basketball events" that satisfy the request represented by the corresponding semantic multicase graph node. In particular, a video key frame extraction agent may identify natural break points in the newscast video and a video segmentation agent may use that information to separate the newscast into segments representing independent new stories. Furthermore, after a video newscast has been segmented, the closed caption text associated with each news story segment may be analyzed and processed by a concept-filtering agent to identify news stories related to left-scoring basketball events for redistribution.

As specified by the on-line video annotation and classification procedure as shown in Fig. 5, appropriate agents choose appropriate algorithms to get target feature descriptors (see Figs. 17 and 20), and output the binary classification result (yes/no) to user's specification based on rules specified for each basketball event. After on-line video segmentation and automatic low-level feature annotation and classification, basketball videos can be parsed and annotated as shown in Fig. 21. The annotated metadata can be further stored in a relational database as shown in Table 2 and serve as the indexes for the continuous basketball video games.

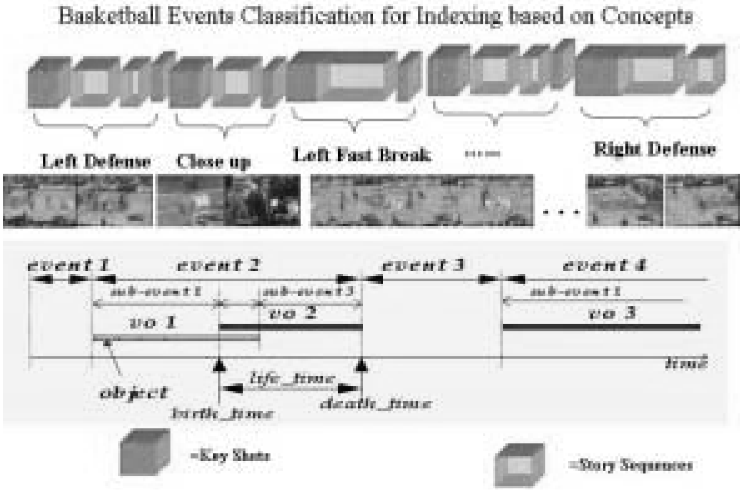


Fig. 21. Basketball indexed with event concepts.

Table 2
Video annotation indexing table based on both perceptual and semantic content

VideolD	SeqlD	Start time	End time	Events	Court Type	MotionDV	FeatureN
V_1	S_1	T_1	T_2	Left scoring	Left	(m_1, m_2)	$(f_1, f_2, \ldots, f_n)$
—	—	—	—	—	—	—	—

These annotated indexes are based on both the low-level perceptual features and the high-level semantic contents as specified by the basketball event categories.

The video archives of on-line live video streams can be queried based on both semantic content and low-level feature similarity matchings. For example, to search for basket ball left-scoring event, we can have query execution plans like Eqs. (5.1) and (5.2)

$$\begin{aligned}&\text{Select } V, S\\&\text{From VideoID } V, \text{SeqID } S,\ \text{Events } E\\&\text{Where Events} = \text{left scoring}\end{aligned}\tag{5.1}$$

Or from Fig. 18, we can

$$\begin{aligned}&\text{Select } V, S\\&\text{From VideoID } V, \text{SeqID } S,\ \text{CourtType } C,\\&\text{MotionDV } M, \text{FeatureN } F_n\\&\text{Where Court} = \text{Left AND } M_1 > \theta_m \text{ AND } F_n \leqslant \theta_n\end{aligned}\tag{5.2}$$

In Eq. (5.2), θ_m and θ_n are the low-level feature value thresholds for each test feature used in the knowledge base. By applying queries using Eqs. (5.1) and (5.2), fast on-line and off-line video information access are enabled.

6. Conclusion

In this paper, we have introduced a novel system approach to on-line knowledge- and rule-based video classification — one that supports automatic indexing filtering based on the semantic concept hierarchy. Our research addresses not only the challenges arising from general video management issues (either on-line or off-line) such as video semantic content analysis, but also the stringent requirements imposed by on-line video processing. The difference in our work and the existing accomplishments in the literature is that while most of them use static models for video classification to provide semantic indexing of off-line multimedia databases, we use supervised learning techniques to form an on-line classification system and apply it specifically to basketball video event indexing as an experimental example. a complete prototype system has been developed to facilitate intelligent access to the rich multimedia data over the Internet, and to evaluate the performance of clustering and video/audio content analysis and features-extraction techniques using sports data.

At the core of our system, we have developed a general video analysis model in conjunction with various techniques for fast and efficient video content analysis, such as scene-change detection, key frame selection, low-level feature extraction and clustering, and video semantic classification. A supervised rule-based video classification system is proposed using video automatic segmentation, annotation and summarization techniques for seamless information browsing and updating. The rules were calculated using an inductive decision-tree-learning approach applied to multiple low-level image features. The proposed rule-based video classification system is good both on-line and off-line, and is useful for numerous video applications. In particular, the classification system was applied to basketball clips with good accuracy, which shows that this system is effective and promising. Because the learning algorithm and low-level features are general, the proposed system is also suitable for other video domains if the appropriate new rules for the specific video are leaned in advance. ultimately, for videos from different domains, a more complete set of video features may be extracted for training processing. One of the major architecture advantages of our on-line multimedia content analysis system is the use of intelligent proxies to encapsulate, coordinate, and combine distinct operations in optimum processing orders for semantic analysis of video and audio contents. Moreover, the system prototype can be made modular and scalable. In the future, we will extend our work to wider video domains, such as for other sports like football or soccer.

Acknowledgements

The work presented in this paper is supported by the DARPA project on Semantic Multicase, No. N66001-97-2-8901.

References

[1] S. Dao, E. Shek, A. Vellaikal, R. Muntz, L. Zhang, M. Potkonjak, Semantic multicast: intelligently sharing collaborative sessions, J. ACM Comput. Surveys (1999).

[2] S.F. Chang, W. Chen, H.J. Meng, H. Sundaram, D. Zhong, VideoQ: an automated content based video search system using visual cues, ACM Multimedia (1997).

[3] Y. Deng, B.S. Manjunath, Content-based search of video using color, texture and motion, Proc. IEEE Int. Conf. Imaging Process. 2 (1997) 534–537.

[4] Y. Deng, B.S. Manjunath, Spatio-temporal relationships and video object extraction, Proceedings of the 32 Asilomar Conference on Signal, System and Computers, November 1998.

[5] N. Dimitrova, F. Golshani, Motion recovery for video content classification, ACM Trans. on Information. Syst. 13 (4) (1995) 408–439.

[6] G. Iyengar, A. Lippman, Models for automatic classification of video sequences, Proceedings of the SPIE Multimedia Storage and Archiving Systems, Vol. 3312, San Jose, CA, 1998, pp. 216–227.

[7] A. Jaimes, S.F. Chang. Model-based classification of visual information for content-based retrieval, Storage and Retrieval for Image and Video Database VII, IS & T/SPIE99, San Jose, January 1999.

[8] R.W. Picard, A society of models for video and image libraries, IBM Syst. J. 35 (1996) 292–312.

[9] T.P. Minka, R.W. Picard, Interactive learning with a society of models, Pattern Recognition 304 (4) (1997) 565–581.

[10] D.D. Saur, Y.P. Tan, S.R. Kulkarni, P.J. Ramadge, Automated analysis and annotation of basketball video. SPIE 3022 (1997).

[11] Y. Gong, L.T. Sin, C.H. Chuan, H. Zhang, M. Sakauchi, Automatic parsing of TV soccer programs, IEEE Trans. (1995) 167–172.

[12] G. Sudhir, J.C.M. Lee, A.K. Jain, Automatic classification of tennis video for high-level content-based retrieval, IEEE Multimedia (1997).

[13] D. Crevier, R. Lepage, Knowledge-based image understanding system: a survey, Computer Vision and Image Understanding 67 (1997) 161–185.

[14] A.M. Nazif, M.D. Levine, Low-level image segmentation: an expert system, Pattern Anal. Mach. Intelll. 6 (1984) 555–577.

[15] R. Quinlan, C4.5: programs for machine learning, Morgan Kaufmann, Berlin, 1993.

[16] T. Mitchell, Machine Learning, Mc-Graw Hills, New York, 1997.

[17] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, d. Steele, P. Yanker, Query by image and video content: the QBIC system, IEEE Comput. 28 (9) (1995) 23–32.

[18] W. Zhou, Y. Shen, A. Vellaikal, C.C. Kuo, On-line scene change detection of multicast (MBone) video, Proc. SPIE (1998).

[19] W. Zhou, A. Vellaikal, Y. Shen, C.C. Kuo, Real-time content-based processing of multicast video over the internet, Proceedings of the 32nd Asilomar Conference on Signals, System and Computers, November 1998.

[20] Boon-Lock Yeo, Bede Liu, Rapid scene analysis on compressed video. IEEE Trans. Circuit and System Video Technol. 5 (1995) 533–544.

[21] G. Ahangera, T.D.C. Little, A survey of technologies for parsing and indexing digital video, J. of Visual Commun. Representation 7 (1) (1996) 28–43.

[22] Tong Zhang, C.-C. JayKuo, Content-based classification and retrieval of audio. SPIE's 43rd Annual Meeting-Conference on advanced Signal Processing Algorithms, Architectures, and Implementations VII, SPIE Vol. 3461, San Diego, July 1998, pp. 432–443.

[23] E. Wold, T. Blum, D. Keislar, J. Wheaten, Content-based classification, search, and retrieval of audio, IEEE Multimedia 3 (3) (1996) 27–36.

[24] N. Dimitrova, F. Golshani, Motion recovery for video content classification, ACM Trans. Inform. Syst. 13 (4) (1995) 408–439.

[25] S. Sclaroff, L. Taycher, M.L. Cascia, ImageRover: a content-based image browser for the world wide web. Proceedings of the IEEE Workshop on Content-based Access of Image and Video Libraries, June 1997.

[26] P. Heckbert, Color image quantization for frame buffer display. SIGGRAPH Proc. (1982) 297.