Journal of Electronic Imaging 10(4), 895–908 (October 2001).

MultiView: Multilevel video content representation and retrieval*

Jianping Fan
University of North Carolina at Charlotte
Department of Computer Science
Charlotte, North Carolina 28223
E-mail: jfan@uncc.edu

Walid G. Aref
Ahmed K. Elmagarmid
Purdue University
Department of Computer Sciences
West Lafayette, Indiana 47907

Mohand-Said Hacid
LISI–UFR d'Informatique
Université Claude Bernard
Lyon 1–Bâtiment Nautibus
8, boulevard Nieles Bohr
69622 Villeurbanne Cedex France

Mirette S. Marzouk
Xingquan Zhu
Purdue University
Department of Computer Sciences
West Lafayette, Indiana 47907

Abstract. *In this article, several practical algorithms are proposed to support content-based video analysis, modeling, representation, summarization, indexing, and access. First, a multilevel video database model is given. One advantage of this model is that it provides a reasonable approach to bridging the gap between low-level representative features and high-level semantic concepts from a human point of view. Second, several model-based video analysis techniques are proposed. In order to detect the video shots, we present a novel technique, which can adapt the threshold for scene cut detection to the activities of variant videos or even different video shots. A seeded region aggregation and temporal tracking technique is proposed for generating the semantic video objects. The semantic video scenes can then be generated from these extracted video access units (e.g., shots and objects) according to some domain knowledge. Third, in order to categorize video contents into a set of semantic clusters, an integrated video classification technique is developed to support more efficient multilevel video representation, summarization, indexing, and access techniques.* © *2001 SPIE and IS&T.* [DOI: 10.1117/1.1406944]

1 Introduction

Digital video now plays an important role in entertainment, education, and other multimedia applications. It has become increasingly important to develop mechanisms that process, filter, search, and organize the digital video information so that useful knowledge can be derived from the exploding mass of information that is now accessable. Since it is difficult to index and categorize video data automatically compared with similar operations on text, search engines for video data are still rare. Content-based video database modeling, representation, summarization, indexing, retrieving, navigating, and browsing have emerged as challenging and important areas in computer vision and database management.

All the existing video database systems first partition videos into a set of access units such as shots, objects or regions,[1–7] and then follow the paradigm of representing video via a set of feature attributes, such as color, texture, shape, and layout.[8,9] Those features are properly indexed, according to some indexing structure, and are then used for

*The short version of this work was first presented at SPIE Electronic Imaging: Storage and Retrieval for Media Databases, San Jose, 24–26 January 2001. This work was supported by NSF under 9972883-EIA, 9974255-IIS, and 9983249-EIA, a grant from the state of Indiana 21th Century Fund, and by grants from HP, IBM, Intel, NCR, Walmart, and Telcordia.

Paper SPR-07 received Mar. 28, 2001; revised manuscript received July 1, 2001; accepted for publication July 3, 2001.
1017-9909/2001/$15.00 © 2001 SPIE and IS&T.

video retrieval. Retrieval is performed by matching the feature attributes of the query object with those of videos in the database that are *nearest* to the query object in high-dimensional spaces. The query-based video database access approaches typically require that users provide an example video or sketch, and a database is then searched for videos which are relevant to the query. Some other approaches to video database management have focused on supporting hierarchical browsing of video contents. In order to support hierarchical video browsing, the video contents are first classified into a set of clusters on the basis of the similarity of their representative visual features.[10–12] However, the up and coming networked content-based video database system still suffers from the following problems.

1. **Video analysis problem:** Video shots or even video objects, which are directly related to video structures and contents, are used as the basic units to access the video sources. A fundamental task of video analysis is to extract such video units from the videos to facilitate the user's access (e.g., retrieving and browsing). Only after such video units become available can content-based retrieving, browsing, and manipulation of video data be facilitated. Automatic semantic video analysis is still hard in current computer vision techniques.[13–16]
2. **Indexing problem:** After the video content analysis procedure is performed, video contents in the databases are represented as independent data points in high-dimensional feature space, and a similarity-based query is equivalent to a *nearest neighbor* (NN) search. High-dimensional indexing structures that have been investigated in recent years seem to be a promising solution to this problem.[17–19] Unfortunately, the efficiency of these existing high-dimensional indexing structures deteriorates rapidly as the number of dimensions increases.[20] On the other hand, the visual features, which are selected for describing video contents, are almost high dimensional.
3. **Representation problem:** It is not easy for a naive database user to express queries appropriately in terms of the given features, thus naive users are interested in browsing or querying the databases at a semantic level. However, the low-level visual features, which can be automatically extracted from the videos, do not correspond in a direct or convenient way to the underlying semantic structure of video contents.[21–24]
4. **Access control problem:** A shortcoming of existing video database systems,[1–7] however, is the lack of suitable access control mechanisms. The development of such mechanisms is increasingly relevant because video data today are used for very different objectives. User-adaptive video database access control is thus becoming one of the important problems, because different network users may have different permissions for accessing different videos or even the same video with possibly different quality levels.
5. **QoS problem:** Given the heterogeneous and dynamic (i.e., varying performance) natures of networks, video contents should be scalable over a wide range of bandwidth requirements to provide fast video retrieving and browsing over networks. However, the current network techniques cannot provide efficient quality of service (QoS) guarantees.

Based on the above observations, a novel multilevel video modeling and indexing approach, called MultiView, is proposed to support hierarchical video retrieving and browsing. This article is organized as follows. In Sec. 2 we discuss related work on content-based video database systems. In Sec. 3 we propose a multilevel video model to support hierarchical video representation, summarization, indexing, retrieving, and browsing. Automatic content-based video analysis techniques used in MultiView are introduced in Sec. 4. A novel integrated video classification algorithm is proposed in Sec. 5. In Sec. 6 we present a multilevel video indexing and accessing structures. We give our conclusions in Sec. 7.

2 Related Work

Content-based video database has emerged as an important and challenging area of research and a number of useful systems have been proposed in the past few years. Here in Sec. 2, a brief overview of these existing content-based video database systems is given. Detailed performance analysis of these systems can be found in Ref. 25.

QBIC, developed at the IBM Almaden Research Center, is an open framework and developing technology, which can be used for both static and dynamic image retrieval.[1] QBIC allows users to graphically pose and refine queries based on multiple visual features such as color, shape and texture. QBIC also supports video querying through shots or key frames.

Virage, developed by Virage Inc.,[4] can adjust the weighting associated with different visual features. Virage includes visual features such as color, texture, color layout, and structure. Virage can also classify images according to visual features or domain specification.

Blobworld,[26] developed at the University of California, Berkeley, can segment images automatically into regions, and these may be semantic objects or parts of semantic objects. The Blobworld system includes color, shape, spatial, and texture features.

Photobook,[2] developed at the Massachusetts Institute of Technology Media Laboratory, supports a set of interactive tools for browsing and searching images. Photobook uses color, shape, texture, and face features. The more recent version of Photobook also includes image annotation and retrieval loop.[24]

VideoQ,[5] developed at Columbia University, supports video querying by examples, visual sketches, and keywords. This system includes color, texture, motion trajectory, shape, and size. VideoQ can support several query types: single-object query and multiple-object query. The same group at Columbia has also developed several other video search engines such as VisualSEEK and WebSEEK.[27,28]

Netra-V,[7] developed at the University of California, Santa Barbara, first segments the videos into a set of regions, and these regions are then tracked among frames. The system uses color, texture, shape, affine motion vec-

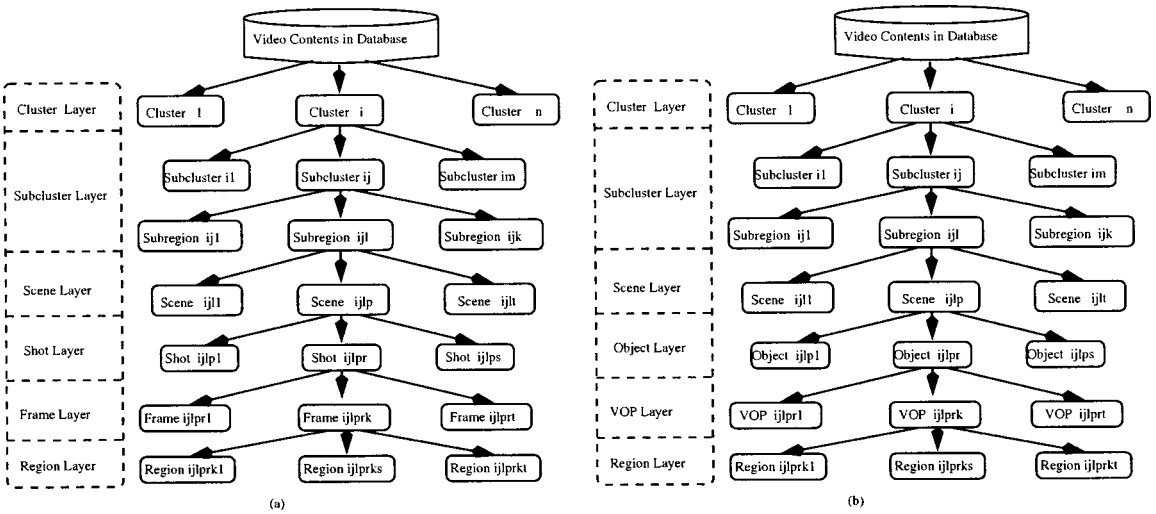


Fig. 1 Multilevel video model of MultiView: (a) main components of the video database model for a shot-based accessing approach; (b) main components of the video database model for an object-based accessing approach.

tors, and spatial location information to search and retrieve similar regions from the database.

MARS,[3] developed at the University of Illinois, Urbana–Champaign, differs from other systems in terms of both the research scope and the techniques used. The main focus of MARS is not on finding a single ''best'' feature representation, but, rather, on how to organize various visual features into a meaningful retrieval architecture which can dynamically adapt to different applications and different users.

Name-It,[6] developed at Carnegie Mellon University, associates name and faces in news videos. To do this, the system detects faces from a news video, locates names in the sound track, and then associates each face with the correct name.

PicHunter,[29] developed at the NEC Research Center, represents a simple instance of a general Bayesian framework for using relevance feedback to direct a search. It also attempts to maximize the information obtained from a user at each iteration of the search.

Browsing-based video database systems,[30–32] which classify video contents into different classes according to their low-level visual features, are also widely studied, and several practical systems have been proposed. Video browsing is useful for identifying relevant video content from a human point of view.

One common shortcoming of these existing image and video database systems is that only a small number of these systems addresses the embedded high-dimensional video indexing structures. Video indexing is fast becoming a challenging and important area when truly large video data sets come into view.[33] Therefore, the cutting-edge research on integrating the computer vision with the database management deserves attention.

3 Multilevel Video Model

Efficient content-based retrieving and browsing of video require well-defined database models and structures. Unlike traditional database models, a suitable video database model must include the elements that represent inherent structures of a large collection of videos and the semantics that represent the video contents. In order to support more efficient video representation and indexing in MultiView, a multilevel video model is introduced by classifying video contents into a set of hierarchical manageable units, such as clusters, subclusters, subregions, scenes, shots or objects, frames or video object planes (VOPs), and regions. Moreover, the semantics at the database level are obtained by an integrated video classification procedure, so that high-level concept-based querying, browsing, and navigating can be supported.

Basic video access units, such as shots and key objects, are first obtained and represented by a set of visual, meta, and semantic features. The related video shots and video objects are further classified into meaningful video scenes according to some domain knowledge (e.g., a scene model).[13–16] The video scenes, which convey the video contents in a database, are then categorized into a set of semantic clusters, and each semantic cluster may consist of a set of subclusters. The subclusters can further be partitioned into a set of subregions to support more efficient high-dimensional video indexing. Each subregion consists of a limited number of similar video contents (e.g., video scenes), so that linear scanning can be used to generate the indexing pointers of the video contents in the same subregion.

The cluster layer may consist of a set of semantic clusters, shown in Fig. 1, which is used to describe the physical structures and semantics of video contents in a database. In order to obtain this cluster layer, we have developed an integrated video classification technique. The subcluster layer includes the physical structures and compact semantic contents of the clusters. The subcluster layer can be obtained by discovering the interesting relationships and characteristics that exist implicitly in the cluster. We will see that including a subcluster layer can provide a more efficient video indexing structure. The scene layer, which is very useful for high-level video database browsing, de-

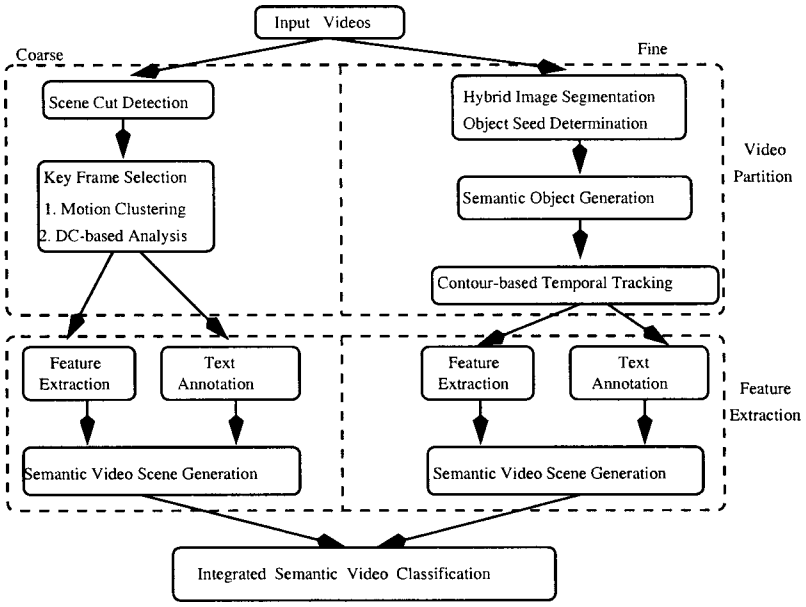


Fig. 2 Block diagram of the semantic video analysis scheme in MultiView.

scribes the semantic video contents. The video shot or object layer describes the video representation, summarization, indexing, and access units. The frame or VOP layer represents visualization of the video content. The region layer describes the spatial components of a visual content and their relationships.

Each video layer is represented by a set of meta, visual, and semantic features. In the cluster layer, each component is characterized by the cluster centroid, radius, feature dimensions, subcluster number, dimensional weighting coefficients, and its node identifier. The cluster centroid and radius are represented by a set of visual features. In the subcluster layer, each component is also characterized by the subcluster centroid, radius, feature dimensions, subregion or object number, dimensional weighting coefficients, and its leaf node identifier. The subcluster centroid and radius are again represented by a set of visual features. The scene layer is represented by a set of visual features, meta features, and semantic features. In the shot or object layer, each component is represented by an indexing identifier, meta features, semantic features, and a set of visual features. In the frame or VOP layer, each component is represented by meta features, semantic features, and a set of visual features which can be obtained from the image regions.

Since all of the video database representation layers are characterized by a set of related visual, meta, and semantic features, a framework for bridging the gap between the low-level features and the high-level concepts can be provided by using an integrated video clustering technique. This multilevel representation and summarization scheme can also provide a scalable method for retrieving and viewing video contents in a database.

4 Content-Based Video Analysis

There are two approaches to accessing video source in databases: *shot based* and *object based* (or even region based). The objective of video analysis is to obtain these basic video access units (e.g., shots and objects). Figure 2 shows a block diagram of the automatic video content analysis scheme developed in MultiView.

4.1 *Video Scene Detection*

Video shots, which are directly related to video structures and contents, are the basic units used for accessing video sources. An automatic shot detection technique has been proposed for adaptive video coding applications,[34] however, in this article we focus on video shot detection on compressed MPEG videos.

Since there are three frame types (I, P, and B) in a MPEG bit stream, we first propose a technique to detect the scene cuts occurring on I frames, and the shot boundaries obtained on the I frames are then refined by detecting the scene cuts occurring on P and B frames. For I frames, block-based DCT is used directly as

$$F(u,v)=\frac{C_u C_v}{4}\sum_{x=0}^{7}\sum_{y=0}^{7} I(x,y) \times\cos\frac{(2x+1)u\pi}{16}\cos\frac{(2y+1)v\pi}{16}, \tag{1}$$

where

$$C_u, C_v=\begin{cases}\dfrac{1}{\sqrt{2}}, & \text{for } u,v=0,\\ 1, & \text{otherwise,}\end{cases} \tag{2}$$

One finds that the dc image [consisting only of the dc coefficient $(u=v=0)$ for each block] is a spatially reduced version of an I frame. For a MPEG video bit stream, a sequence of dc images can be constructed by decoding only the dc coefficients of I frames, since dc images retain most of the essential global information of image components.

Yeo and Liu have proposed a novel technique for detecting scene cuts on the basis of dc images of a MPEG bit stream,[35] in which the scene cut detection threshold is determined by analyzing the difference between the highest and second highest histogram difference in the sliding window. In this article, an automatic dc-based technique is proposed which adapts the threshold for scene cut detection to the activities of various videos. The color histogram differences (HD) among successive I frames of a MPEG bit stream can be calculated on the basis of their dc images as

$$\mathrm{HD}(j,j-1)=\sum_{k=0}^{M}[H_{j-1}(k)-H_j(k)]^2, \tag{3}$$

where $H_j(k)$ denotes the dc-based color histogram of the jth I frame, $H_{j-1}(k)$ indicates the dc-based color histogram of the $(j-1)$th I frame, and k is one of the M potential color components.

The *temporal relationships* among successive I frames in a MPEG bit stream are then classified into two opposite classes according to their color histogram differences and an optimal threshold $\overline{T_c}$.

$$\begin{aligned}&\mathrm{HD}(j,j-1)>\overline{T_c}, \quad \text{scene\_cut},\\ &\mathrm{HD}(j,j-1)\leqslant\overline{T_c}, \quad \text{non\_scene\_cut}.\end{aligned} \tag{4}$$

The optimal threshold $\overline{T_c}$ can be determined automatically by using the fast searching technique given in Ref. 34. The video frames (including the I, P, and B frames) between two successive scene cuts are taken as one video shot. Since the MPEG bit stream is generated by a fixed periodic frame types, the scene cuts may not always occur on the I frames; these scene cuts may also occur on the P frames and B frames. Therefore, these detected shot boundaries should be refined by detecting scene cuts occurring on the P and B frames. These scene cuts are detected according to the following criteria.

1. If a scene cut occurs before a P frame, the most macroblocks in the P frame should be encoded as I blocks because the assumption of motion-compensation prediction coding is lost. If such a P frame is detected, the corresponding shot boundary (the scene cut obtained by using I frame) should be reset to the corresponding P frame.
2. If a scene cut occurs before a B frame, the most macroblocks in the B frames should be encoded as I blocks or backward-predicted blocks because the temporal correspondence between the B frame and its forward reference frame is lost. If such a B frame is detected, the shot boundary should be reset to the corresponding B frame.

Gradual transitions such as cross dissolves, fade ins, and fade outs allow two shots to be connected in a smooth way. Gradual transitions, which are attractive for detecting high-level semantic events, can be determined by analyzing the variance of the histogram differences. The average and variance of the histogram difference for the nth video shot with M frames can be calculated as

$$\begin{aligned}\mu_n&=\frac{1}{M}\sum_{i=1}^{M}\mathrm{HD}(i,i+1),\\ \sigma_n&=\frac{1}{M}\sum_{i=1}^{M}|\mathrm{HD}(i,i+1)-\mu_n|^2.\end{aligned} \tag{5}$$

The *absolute variance* σ_i, between the ith frame in the nth video shot and its average histogram difference μ_n, can be defined as

$$\sigma_n^i=|\mathrm{HD}(i,i+1)-\mu_n|^2. \tag{6}$$

The activity of each frame in a dissolve transition shot produces a U-shaped curve of the absolute variance. In the case of fade in and fade out, the absolute curve shows a monotonous increase or decrease. The gradual transitions can be detected when an appropriate number of subsequent frames exhibit values of $\mathrm{HD}(i,i-1)$ that are greater than the determined threshold $\overline{T_c}$, together with the occurrence of a value of σ_n^i greater than $\overline{T_\sigma}$. $\overline{T_\sigma}$ can be determined by

$$\begin{aligned}\overline{T_\sigma}&=\overline{T_c}-\delta,\\ \delta&=c_1\mu_n+c_2\sigma_n,\end{aligned} \tag{7}$$

where the coefficients c_1 and c_2 are determined by experiment.[8] Since dissolve and fade processes have a long duration (this property is very different from that of scene cut), shot length can also be included as a critical parameter for gradual transition detection. The experimental results for scene cut detection from two compressed MPEG medical videos are given in Figs. 3 and 4. The average performances of our scene cut detection technique for various video types are given in Tables 1 and 2. The semantic video scenes can be further generated from these extracted video shots according to some domain knowledge.[13–15] Moreover, the meta data, which are represented by the keywords of text annotation, can also be used for generating a semantic video scene.

4.2 *Video Object Extraction*

The previous shot-based video representation and access technique does not capture the underlying semantic structure of video sources. Extracting the semantic structure of video sources is very important for providing more effective video retrieval and browsing, because people watch videos based on semantic contents, not on physical shots or key frames. Due to their inherent content dependence, video objects are especially suitable for representing semantic video contents.

Automatic moving object extraction also plays a fundamental role in computer vision, pattern recognition, and object-oriented video coding. Many approaches to automatic moving object extraction have been proposed in the past.[36–39] However, the outputs of these feature-based video segmentation techniques are only the homogeneous regions according to the selected visual features. It is still hard for current computer vision techniques to extract the semantic objects from a human point of view, but semantic object generation for content-based video indexing is becoming possible because the videos can be indexed by some se-

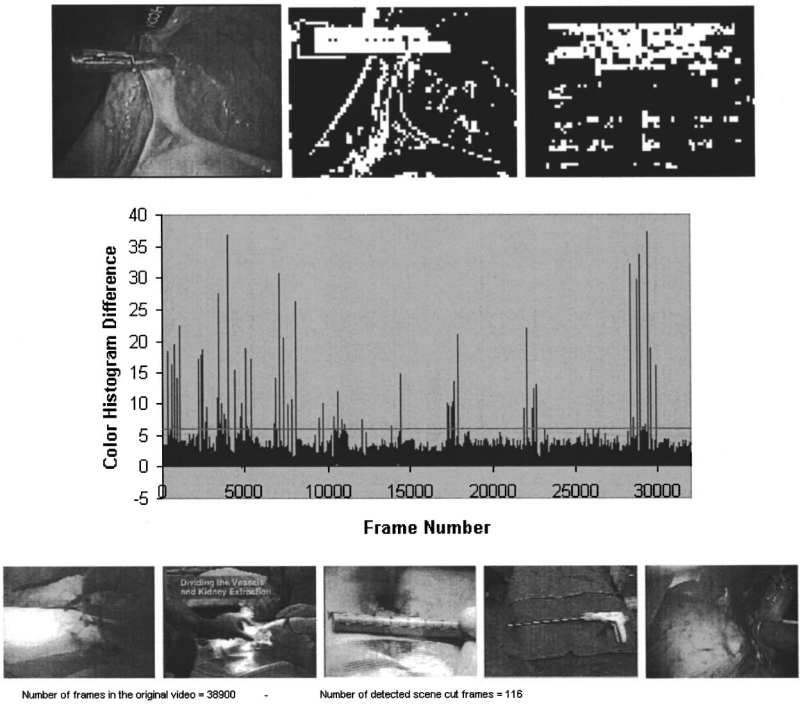


Fig. 3 Scene cut detection results and the corresponding color histogram difference: (a) the first I frame; (b) spatial segmentation result on block resolution; (c) temporal change regions on block resolution; (d) color histogram difference with the determined threshold; (e) part of the detected scene cut frames.

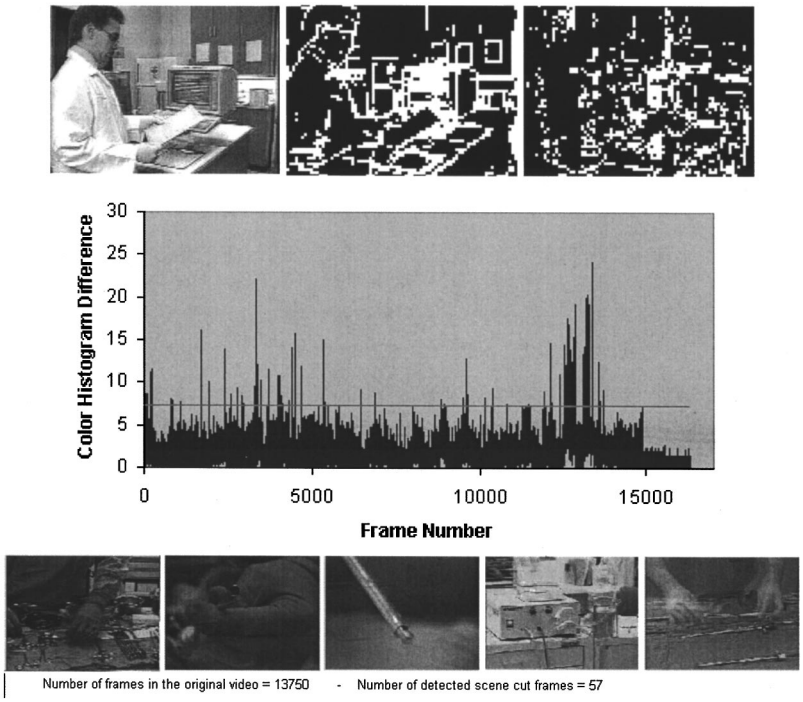


Fig. 4 Scene cut detection results and the corresponding color histogram difference: (a) the first I frame; (b) spatial segmentation result on block resolution; (c) temporal change regions on block resolution; (d) color histogram difference with determined threshold; (e) part of the detected scene cut frames.

Table 1 Average performance of our adaptive scene cut detection technique for news sequences.

Test videos frame numbers	news1.mpg 5177	news2.mpg 6288	news3.mpg 7024
Break shots	86	98	129
Gradual shots	6	11	7
Missed shots	4	7	15
False alarms	5	6	13

Table 2 Average performance of our adaptive scene cut detection technique for medical image sequences.

Test videos Frame numbers	med1.mpg 33 200	med2.mpg 15 420
Break shots	116	57
Gradual shots	21	48
Missed shots	6	9
False alarms	5	11

mantic objects of interest, such as human beings, cars, or airplanes. This interest-based video indexing approach is reasonable because users do not focus on all the objects presented in the videos.[16] This reduces the difficulties of automatic semantic object generation for video indexing.

Based on the above observations, a *seeded region aggregation* and temporal tracking technique is proposed for generating semantic objects. The steps in this process are as follows.

1. A hybrid image segmentation technique integrates the results of an edge detection procedure and a similarity-based region growing procedure.
2. Several independent functions are designed such that every function provides one type of semantic object. Each function uses the *object seed* and *region constraint graph* (e.g., a perceptual model) of its corresponding semantic object.[16]
3. If an object seed is detected, a *seeded region aggregation* procedure is used to merge the adjacent regions of the object seed as the semantic object.[40] The perceptual model of a semantic object can guide the way the adjacent regions of object seeds should be put together.
4. If the above automatic object extraction procedure fails to obtain the semantic object from a human point of view, human interaction defines the semantic objects in the initial frame.[41]
5. Once the semantic objects have been extracted, they are tracked across frames, i.e., along the time axis. To this end, we use a *contour-based temporal tracking* procedure.[38] The procedure uses two semantic features, *motion* and *contour*, to establish object correspondence across frames. The *k*th Hausdorff distance technique is used to guarantee the temporal object tracking procedure.[42]

A set of results for four video sequences that are well known in the video coding community, namely, ''Akiyo,'' ''Carphone,'' ''Salesman,'' and ''News,'' are given in Figs. 5, 6, and 7. Since the seeds for different semantic objects are identified, the proposed seeded semantic video object extraction technique is very attractive for multiple object extraction. The semantic video objects, which are obtained by integrating human–computer interaction to define the semantic objects with an automatic temporal tracking procedure, are shown in Fig. 8. A set of visual features can also be selected to represent the video contents in the database.

5 Integrated Video Classification

There are three conventional approaches for accessing the video contents in database.

1. *Query by example* is widely used in existing video database systems. The example-based approach is necessary in a situation where users cannot clearly describe what they want by using only text. In order to provide query by example, all the videos in the databases are indexed through a set of high-dimensional visual features according to some indexing structures. Retrieval is then performed by match-

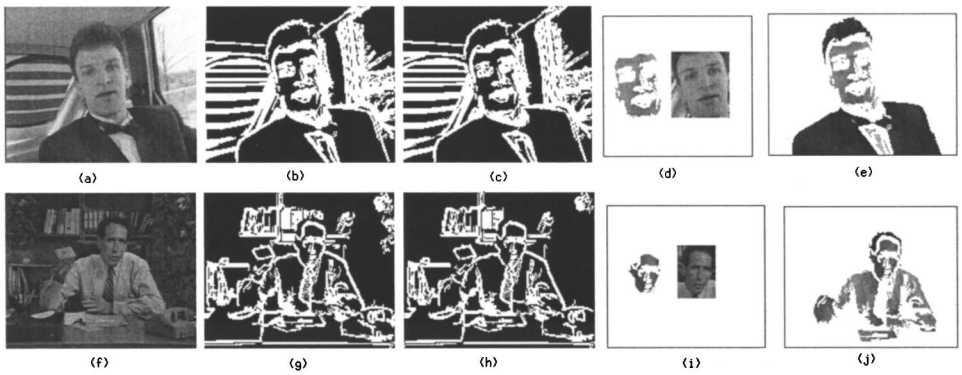


Fig. 5 (a) Original image of ''Carphone;'' (b) the color edges; (c) the intensity edges; (d) the detected face and its rectangular region; (e) the extracted objects with region edges; (f) the original image of ''Salesman;'' (g) the color edges; (h) the intensity edges; (i) the detected face and its rectangular region; (j) the extracted object with region edges.

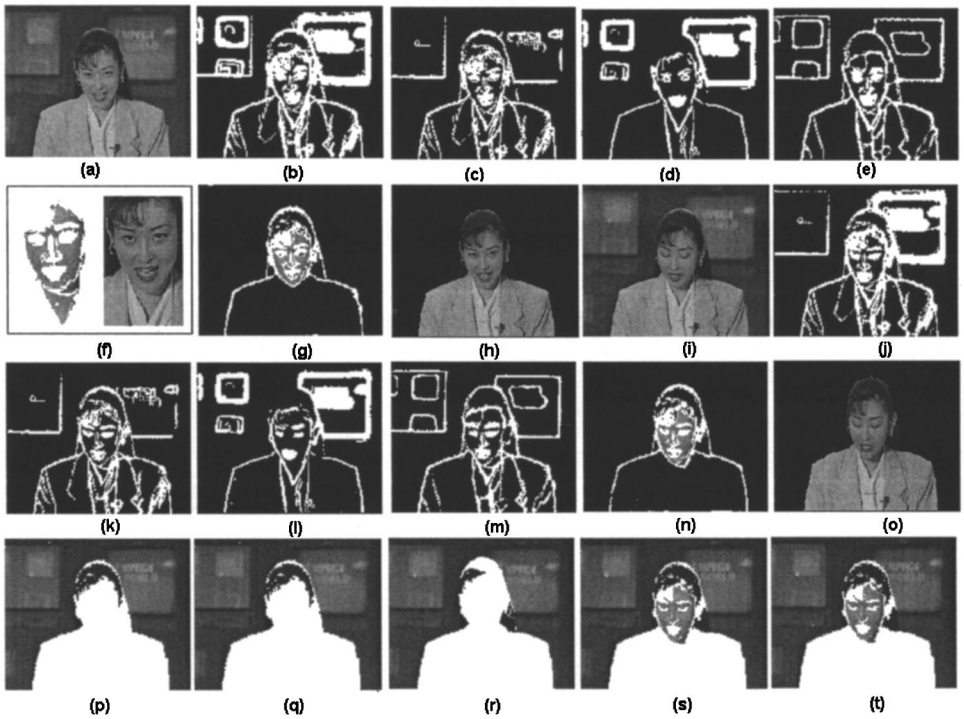


Fig. 6 (a) Original image of "Akiyo;" (b) the color edges; (c) the intensity edges; (d) the chrominance edges; (e) the region boundaries; (f) the human face and its rectangular region; (g) the connective edges of the object seed; (h) the extracted semantic object; (i) the original image; (j) the color edges; (k) the intensity edges; (l) the chrominance edges; (m) the region boundaries; (n) the connective edges of the object seed; (o) the extracted semantic object; (p) the object region obtained by using size ratio constraint (frame 10); (q) the object region obtained by using size ratio constraint (frame 15); (r) the object region obtained by using size ratio constraint (frame 120); (s) the object region (with the face seed) obtained by using size ratio constraint (frame 290); (t) the object region (with the face seed) obtained by using size ratio constraint (frame 298).

ing the feature attributes of the query object with those of videos in databases.[43,44] However, the query-by-example approach suffers from at least two problems. The first one is that not all database users have video examples at hand. Even if the video database system interface can provide some video templates, there is still a gap between the various requirements of different users and the limited templates provided by the database interface. The second one is that naive users may prefer to query the video database at a semantic level through keywords. However, it is not easy for current computer vision techniques to bridge the gap between the low-level visual features and high-level concepts from a human point of view.

2. *Query by keywords* is also used in some video database systems based on text annotation. There are three approaches that can provide text annotation of video contents: (a) obtain keywords from the text captions in videos through OCR techniques;[45,46] (b) use free text annotation by humans with domain-specific knowledge to provide a semantic interpreta-

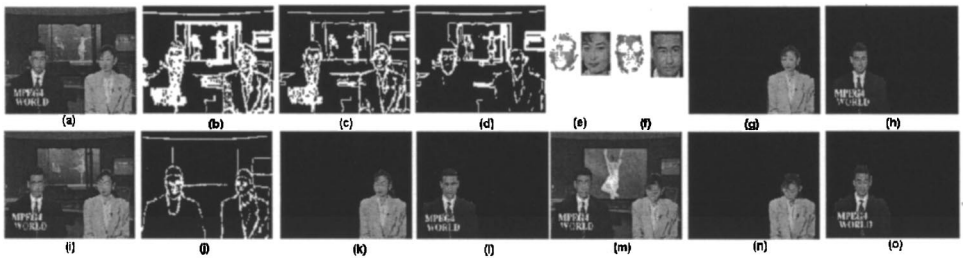


Fig. 7 Object extraction results from "News." First frame: (a) original image; (b) color edges; (c) luminance edges; (d) chrominance edges; (e) human face of object 1; (f) human face of object 2; (g) object 1; (h) object 2; 10th frame: (i) original image; (j) region boundaries; (k) tracked object 1; (l) tracked object 2; 260th frame: (m) original image; (n) tracked object 1; (o) tracked object 2.

Fig. 8 Results of semantic object definition and temporal tracking.

tion of the video contents; (c) perform speech recognition and natural language understanding procedures on the audio channel.[47,48] This query-by-keywords approach also presents at least two problems. The first is that different people may have a different understanding of the same video content, and it is not easy for a naive video database user to figure out the exact keywords to use for the query. The second is that it cannot provide query by example because it would be difficult for a naive user to translate the contents of the video examples at hand into keywords. A practical video database system should support both query by example and query by keywords, thus a mapping from high-level semantic concepts (e.g., those represented by keywords of text annotation) to low-level visual features should be available.

3. *Random browsing* is widely accepted by the naive video database users. Naive users are interested in browsing the video database at the semantic level, rather than having to use visual features or keywords to describe their requests. In order to support random browsing, the video contents should be classified into a set of semantic clusters from a human point of view. Since the low-level visual features do not correspond in a direct way to the semantic concepts, a good solution to bridge the gap between them is needed.

One way to resolve the semantic gap comes from sources outside the video that integrate other sources of information about the videos in the database. In MultiView, the video contents in the database are jointly represented by a set of visual features and keywords of the text annotation. There are two different similarity measures for comparing two video contents with semantic labels s and t:[3,21] the weighted *feature-based similarity distance* $d_F(O_s,O_t)$ and the *semantic similarity distance* $d_S(O_s,O_t)$.

$$d_F(O_s,O_t)=\sum_{i=1}^{n}\frac{1}{a_i}\cdot d_F^i(O_s,O_t), \tag{8}$$

$$d_S(O_s,O_t)=\sum_{i=1}^{m} d_S^i(O_s,O_t), \tag{9}$$

where a_i is the ith dimensional weighting coefficient, $d_F^i(O_s,O_t)$ is the feature-based similarity distance according to the ith dimensional representative feature, $d_S^i(O_s,O_t)$ is the semantic distance according to the ith keyword of the content interpretation, n is the total number of dimensions of visual features, and m is the total number of keywords used for content interpretation.

$$d_F^i(O_s,O_t)=\sum_{j=1}^{n}\sum_{k=1}^{n} b_{jk}(f_{s,j}^i-f_{t,j}^i)(f_{s,k}^i-f_{t,k}^i), \tag{10}$$

$$d_S^i(O_s,O_t)=\begin{cases}0, & \text{if } O_s^i=O_t^i,\\ 1, & \text{otherwise,}\end{cases} \tag{11}$$

where $f_{s,j}^i$ is the ith dimensional visual feature of the jth video sample, and an $n\times n$ matrix $\mathbf{W_i}=[b_{jk}]$ defines a *generalized ellipsoid distance.*

The aim of MultiView is to provide maximum support in bridging the semantic gap between low-level visual features and high-level human concepts given by text annotation, thus an integrated video content classification technique is used. We first assume that the video contents in the

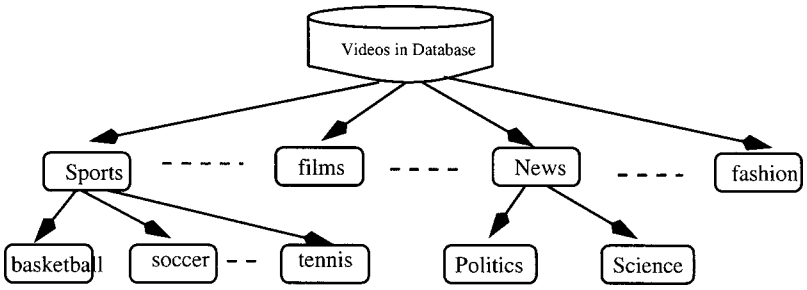


Fig. 9 Knowledge-based tree structure for hierarchical video classification and categorization.

database can be classified into a set of semantic clusters through a knowledge-based hierarchical partition procedure, shown in Fig. 9. The semantic video classification can first be obtained by clustering video contents according to the keywords of their text annotation, and these video contents can then be indexed and accessed through the keywords of the text annotation. On the other hand, the video contents can also be categorized into a set of the same semantic clusters according to the similarity of their visual features. Since the video similarity on low-level visual features does not correspond directly to the similarity on high-level concepts from a human point of view, the results obtained by these two different video classification approaches must be integrated to support more efficient video database access. There are four possible ways to integrate the results obtained by these two different video classification approaches.

1. *Good matching video contents*, which are similar according to both the keywords of text annotation and low-level visual features, should be put into the same semantic cluster. This means that the semantic similarity of these video contents corresponds directly to their weighted feature-based similarity.
2. *Bad matching video contents*, which are similar according to the keywords of their text annotation but dissimilar according to their low-level visual features, should be put into the same semantic cluster. However, their dimensional weighting coefficients should be renormalized, so that their weighted feature-based similarity corresponds in a direct way to their semantic similarity from a human point of view. Since different visual features may play different degrees of importance in making the final decision on the semantic similarity from a human point of view, a learning-based optimization technique can be used to choose the suitable dimensional weighting coefficients.
3. *Wrong matching video contents*, which are similar according to their low-level visual features but dissimilar from a human point of view, should be put into different semantic clusters. A learning-based optimization procedure is performed for reweighting the importance of their different dimensional visual features, so that these dissimlar video contents from a human point of view can have large weighted feature-based distances.
4. *Good mismatching video contents*, which are dissimilar according to both the keywords of text annotation and low-level visual features, should be put into different semantic clusters.

The good matching and good mismatching video contents are taken as positive examples. The wrong and bad matching video contents are taken as negative examples. The positive and negative examples, which are taken as the input of a learning-based optimization processor, are used to select the suitable dimensional weighting coefficients, so that the weighted feature-based similarity can correspond directly to its concept-based similarity. Since the semantic similarities among these labeled video contents are given, the system then learns from these video content examples and selects the suitable dimensional weighting coefficients, shown in Fig. 10.

6 Multilevel Video Indexing and Access

The objective of MultiView is to provide a reasonable solution to the problems related to the up and coming networked video database systems. Three kinds of video database accessing approaches can be supported by MultiView: query by example, query by keyword, and random browsing. Many tree structures have been proposed for indexing high-dimensional data,[17–19] however, it is widely accepted that the efficiency of these existing high-dimensional indexing structures deteriorates rapidly as the number of dimensions increases. Therefore, more efficient high-

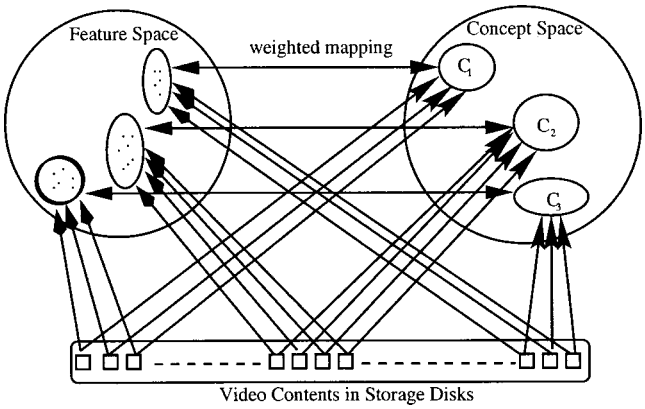


Fig. 10 Relationships among the video contents in the database and classification of the data points in feature space and in concept space.

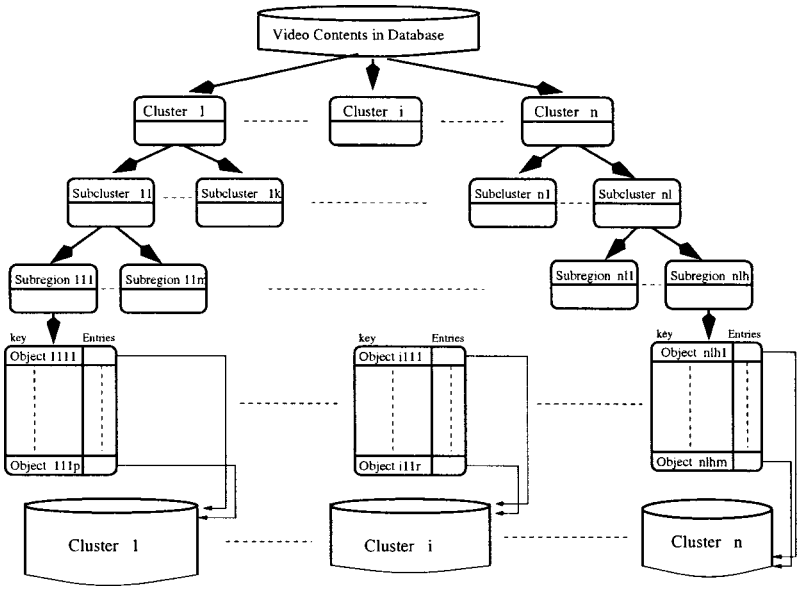


Fig. 11 Multilevel video indexing structure and the distributed storage disks in MultiView.

dimensional video indexing technique should be proposed before video search engines can be provided.

The existing tree structure divides the high-dimensional space into a number of subregions, and each subregion contains a subset of objects that can be stored in a small number of disk blocks. From this point of view, the multilevel video modeling and partitioning techniques in MultiView have also provided a multilevel video representation and indexing structure. We now study how semantic video clustering and multilevel video representation techniques can be effectively combined to support more efficient high-dimensional video indexing.

Based on the proposed multilevel video model, the video contents in a database are first classified into a set of semantic clusters by using the integrated video clustering technique introduced in Sec. 5. In order to support more efficient query processing, each semantic cluster is then partitioned into a set of subclusters by discovering the interesting relationships and implicit characteristics. Each subcluster may consist of a set of subregions, so that linear scanning can be used for generating the indexing pointers of the high-dimensional video contents in the same subregion. This hierarchical partitioning of a semantic cluster will end when the number of multidimensional video contents in each subregion is less than a predefined threshold, $\log N \ll D_i$, where N is the total number of multidimensional video contents in the subregion, and D_i is the number of dimensions of the representative features for the corresponding subregion.

The indexing structure consists of a set of separate indices for the clusters and each cluster is connected to a single root node as shown in Fig. 11. The indexing structure includes a set of hash tables for different layers of a video database: a root hash table for keeping track of information about all the clusters in database, a leaf hash table for each cluster for preserving information about all its subclusters, a second-leaf hash table for each subcluster for keeping information about all its subregions, and a hash table for each subregion for mapping all its data points to the associated disk pages where the videos reside, as shown in Fig. 12.

The root hash table keeps information about all the semantic clusters, and each root node may consist of a set of leaf nodes to access its subclusters. Recall that the representative features associated with each root node are the centroid, radius, meta features, semantic features, dimensional weighting coefficients, number of leaf nodes, and representative icons. Each leaf node is also represented by a set of parameters. Hash tables for the clusters, subclusters, and subregions are devised where the keys are the representative features that characterize their centroids and radiis, and the entries are the pointers to the lower-level components of the hierarchy.

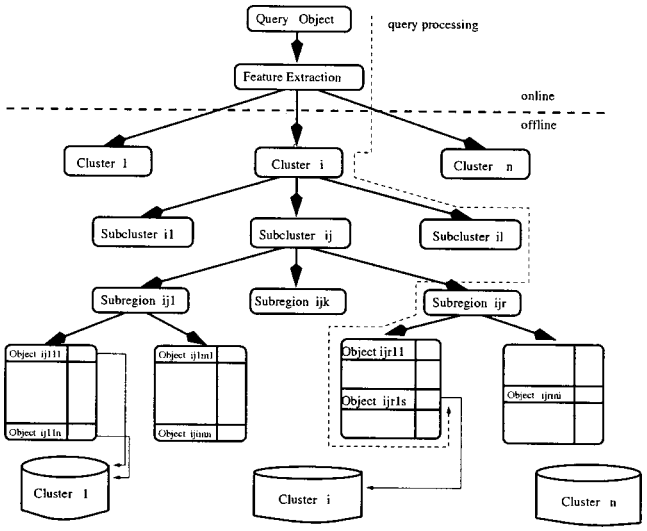


Fig. 12 Multilevel video query processing in MultiView.

The hash table for each subregion is built by mapping all of its videos to the associated disk pages, and an indexing pointer is assigned for each video. Each subregion contains a subset of videos that can be stored in a small number of disk blocks, $\log N \ll D_i$. Hash tables for the objects in a subregion can be devised where the keys are their representative features and the entries are pointers to the disk pages where the videos reside.

To improve input/output (I/O) efficiency, all the semantic clusters are stored in a set of independent disks, shown in Fig. 11. To answer a query, only the semantic clusters that are relevant to the query object are retrieved. The traditional high-dimensional indexing trees, such as *R* tree, X tree, and SR tree, can also be used for indexing these high-dimensional video contents in the same subregion. However, it is widely accepted that if $D_i \gg \log N$, then no nearest neighbor algorithm can be significantly faster than a linear search. Therefore, a linear scanning technique is used to obtain the pointers for these video contents in the same subregion.[49]

In order to answer a query by example, the similarity search is performed in three steps, as shown in Fig. 12.

1. It performs a *where-am-I* search to find out which subregion the given query object resides in. To do this, the search engine first tries to find the cluster C_i that is relevant to the query object O. Their weighted feature-based similarity distance $d_F(O,\vec{x}_c)$ is also calculated as

$$d_F(O,C_i)=d_F(O,\vec{x}_C)=\sum_{j=1}^{D_i}\frac{1}{a_j}d_F^j(O,\vec{x}_{c,j}), \tag{12}$$

where $d_F^j(O,\vec{x}_{c,j})$ is the similarity distance between the query object O and the centroid of cluster C_i according to their jth dimensional features. The query processor returns the cluster C_k, which has the smallest weighted feature-based similarity distance with the query object O or where the associated similarity distance $d_F(O,C_k)$ is no more than the radius ϕ_c^k of C_k.

$$d_F(O,C_k)=\min_{i\in[1,2,\ldots,q]}\{d_F(O,C_i\} \tag{13}$$

If such a cluster C_k exists, the query processor finds the associated subcluster in the cluster C_k which is most relevant to the query object O, and then finds the most relevant subregion by invoking a similar searching procedure.

2. It then performs a *nearest-neighbor* search in the relevant subregion to locate the neighboring regions where the similar objects may reside. Since a multidimensional linear scanning technique is used for generating the pointers for the objects in the same subregion, the weighted feature-based similarity distances between the query object O and all the objects in the selected subregion are calculated. The search engine then returns a set of ranked objects which are relevant to the query object.
3. It visualizes the icon images of the ranked query results. The users are then in a final position to make a decision as to which one they really want by browsing the content summaries of these ranked query results.

The time for the *where-am-I* step T_s, that is, finding the most relevant subregion hierarchically, is bounded by $(n+l+k)\cdot T_{s_1}$, where T_{s_1} is the time needed to calculate the weighted feature-based similarity distance between the query object and the centroid of a cluster, subcluster, or subregion. n indicates the total number of clusters in the database, l denotes the total number of subclusters in the relevant cluster, and k is the total number of subregions in the relevant subcluster. The time for the *nearest-neighbor-searching* step T_c, that is, finding the nearest neighbor of the query object in the relevant subregion, is bounded by $S\cdot T_{s_1}$, where S is the total number of objects in the relevant subregion. The time for the *ranking* step T_r, that is, ranking the objects in the relevant subregion, is bounded by $O(S\log S)$, where S is the total number of objects in the relevant subregion. Therefore, the total search time of this multilevel query procedure is bounded by

$$T=T_s+T_c+T_r=(n+l+k+S)\cdot T_{s_1}+O(S\log S). \tag{14}$$

Due to the semantic gap, visualization of the query results in video retrieval is of great importance for the user to make a final decision. Since clusters are indexed independently, users can also start their query by first browsing the clusters to find the one relevant to their query, and then send their query to this relevant cluster. This *browsing-based-query* procedure can provide more semantic results based on the user's concept because only the users know exactly what they want. Moreover, this browsing-based query technique can speed up query by example.

Our semantic clustering technique and multilevel video representation, summarization, and indexing structures are very suitable for fast browsing. Moreover, a semantic manual text title and a set of icon images are associated with each cluster, and these semantic titles or icon images can then be categorized into the form of a table to provide an overview of the video contents in the databases. This categorization of video contents into semantic clusters can be seen as one solution for bridging the gap between low-level visual features and high-level semantic concepts, and it can be helpful both in organizing video databases and in obtaining automatic annotation of video contents.

Three kinds of browsing can be provided: browsing the whole video database, browsing the selected semantic cluster, and browsing the selected video sequence. Browsing the whole video database is supported by arranging the available semantic titles into a cluster-based tree. The visualization of these semantic clusters (root nodes) contains a semantic text title and a set of icon images (semantic visual templates, seeds of cluster).

Browsing the selected semantic cluster is supported by partitioning the video contents in the same cluster into a set of subclusters, and the icon video content for each subcluster is also visualized. Browsing the selected semantic cluster, which is supported by arranging the available semantic icon video contents into a tree, is the same as the procedure of browsing the whole database. Browsing a single video

sequence is, in some respects, a more complicated problem. The shot-based abstraction of video content, which is constructed by a set of key frames or key VOPs, is used to provide fast browsing of a single video sequence.

Our multilevel video modeling structure can also guarantee more efficient video content description schemes. High-level MPEG-7 description schemes can be developed based on our multilevel video representation and indexing structures.[27,50–52]

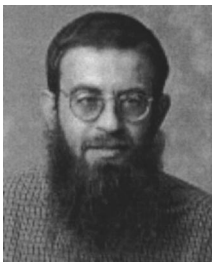
7 Conclusion

An integrated content-based video retrieving and browsing approach, called MultiView, was presented. MultiView focuses on multilevel video modeling and representation to guarantee high-dimensional video indexing. The multilevel video indexing structure used in MultiView cannot only speed up query by example but can also provide more effective browsing. Moreover, high-level MPEG-7 video description schemes can be supported by our multilevel video representation and indexing structures.

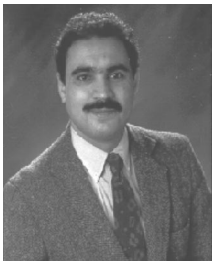
References

1. C. Faloutsos, W. Equitz, M. Flickner, W. Niblack, D. Petkovic, and R. Barber, ''Efficient and effective querying by image content,'' *J. Intell. Inf. Syst.* **3**, 231–262 (1994).
2. A. Pentland, R. W. Picard, and S. Sclaroff, ''Photobook: Content-based manipulation of image databases,'' *Int. J. Comput. Vis.* **18**, 233–254 (1996).
3. Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra, ''Relevance feedback: A power tool for interactive content-based image retrieval,'' *IEEE Trans. Circuits Syst. Video Technol.* **8**, 644–655 (1998).
4. A. Hampapur, A. Gupta, B. Horowitz, C. F. Shu, C. Fuller, J. Bach, M. Gorkani, and R. Jain, ''Virage video engine,'' *Proc. SPIE* **3022**, 188–197 (1997).
5. S. F. Chang, W. Chen, H. J. Meng, H. Sundaram, and D. Zhong, ''A fully automatic content-based video search engine supporting spatiotemporal queries,'' *IEEE Trans. Circuits Syst. Video Technol.* **8**, 602–615 (1998).
6. S. Satoh and T. Kanade, ''Name-It: Association of face and name in video,'' in *Proc. Computer Vision and Pattern Recognition* (1997).
7. Y. Deng and B. S. Manjunath, ''NeTra-V: Toward an object-based video representation,'' *IEEE Trans. Circuits Syst. Video Technol.* **8**, 616–627 (1998).
8. H. J. Zhang, J. Wu, D. Zhong, and S. Smoliar, ''An integrated system for content-based video retrieval and browsing,'' *Pattern Recogn.* **30**, 643–658 (1997).
9. A. K. Jain, A. Vailaya, and X. Wei, ''Query by video clip,'' *ACM Multimedia Syst.* **7**, 369–384 (1999).
10. D. Zhong, H. J. Zhang, and S.-F. Chang, ''Clustering methods for video browsing and annotation,'' *Proc. SPIE* **2670**, 239–246 (1996).
11. M. M. Yeung and B. Liu, ''Efficient matching and clustering of video shots,'' *IEEE Int. Conf. on Image Processing*, pp. 338–341 (1995).
12. B.-L. Yeo and M. M. Yeung, ''Classification, simplification and dynamic visualization of scene transition graphs for video browsing,'' *Proc. SPIE* **3312**, 60–70 (1997).
13. Y. Rui, T. S. Huang, and S. Mehrotra, ''Constructing table-of-content for videos,'' *Multimedia Syst.* **7**, 359–368 (1999).
14. M. Yeung, B.-L. Yeo, and B. Liu, ''Extracting story units from long programs for video browsing and navigation,'' *Proc. IEEE Conf. on Multimedia Computing and Systems* (1996).
15. D. Swanberg, C.-F. Shu, and R. Jain, ''Knowledge guided parsing in video database,'' *Proc. SPIE* **1908**, 13–24 (1993).
16. J. Fan, D. K. Y. Yau, M.-S. Hacid, and A. K. Elmagarmid, ''Model-based semantic object extraction for content-based video representation and indexing,'' *Proc. SPIE* **4315**, 523–535 (2001).
17. A. Guttman, ''R-trees: A dynamic index structure for spatial searching,'' *ACM SIGMOD'84*, pp. 47–57 (1984).
18. N. Katayama and S. Satoh, ''The SR-tree: An index structure for high dimensional nearest neighbor queries,'' *ACM SIGMOD* (1997).
19. S. Berchtold, D. A. Keim, and H. P. Kriegel, ''The X-tree: An index structure for high-dimensional data,'' *Proc. VLDB'96*, Bombay, India, pp. 28–39 (1996).
20. A. Thomasian, V. Castelli, and C.-S. Li, ''Clustering and singular value decomposition for approximate indexing in high dimensional space,'' *CIKM'98*, Bethesda, Maryland, pp. 201–207.
21. Y. Rui and T. S. Huang, ''A novel relevance feedback technique in image retrieval,'' *Proc. ACM Multimedia'99*, pp. 67–70 (1999).
22. Y. Ishikawa, R. Subramanya, and C. Faloutsos, ''Mindreader: Querying databases through multiple examples,'' *Proc. VLDB'98* (1998).
23. C. Yang and T. Lozano-Perez, ''Image database retrieval with multiple-instance learning techniques,'' *Proc. ICDE*, pp. 233–243 (2000).
24. T. P. Minka and R. W. Picard, ''Interactive learning using a 'society of models,'' *Proc. IEEE CVPR*, pp. 447–452 (1996).
25. Y. Rui, T. S. Huang, and S.-F. Chang, ''Image retrieval: Past, present, and future,'' *J. Visual Commun. Image Represent.* **10**, 39–62 (1998).
26. C. Carson, S. Belongie, H. Greenspan, and J. Malik, ''Blobworld: Image segmentation using expection-maximization and its application to image querying,'' *Proc. Int. Conf. Visual Inf. Systems* (1999).
27. J. R. Smith and S.-F. Chang, ''VisualSEEK: A fully automated content-based image query system,'' *ACM Multimedia* **3**, 87–98 (1996).
28. J. R. Smith and S.-F. Chang, ''Visually searching the web for content,'' *IEEE Multimedia* **4**, 12–20 (1997).
29. I. J. Cox, M. L. Miller, T. P. Minka, T. V. Papathomas, and P. N. Yianilos, ''The Bayesian image retrieval system, PicHunter: Theory, implementation, and psychophysical experiments,'' *IEEE Trans. Image Process.* **9**, 20–37 (2000).
30. J.-Y. Chen, C. Taskiran, A. Albiol, E. J. Delp, and C. A. Bouman, ''ViBE: A compressed video database structured for active browsing and search,'' *Proc. SPIE* **3846**, 148–164 (1999).
31. J. R. Smith, ''VideoZoom spatial-temporal video browsing,'' *IEEE Trans. Multimedia* **1**(2), 121–129 (1999).
32. M. M. Yeung and B.-L. Yeo, ''Video visualization for compact presentation and fast browsing of pictorial content,'' *IEEE Trans. Circuits Syst. Video Technol.* **7**, 771–785 (1997).
33. A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, ''Content-based image retrieval at the end of the early years,'' *IEEE Trans. Pattern Anal. Mach. Intell.* **22**, 1349–1380 (2000).
34. J. Fan, D. K. Y. Yau, W. G. Aref, and A. Rezgui, ''Adaptive motion-compensated video coding scheme towards content-based bit rate allocation,'' *J. Electron. Imaging* **9**(4), 521–533 (2000).
35. B. L. Yeo and B. Liu, ''Rapid scene change detection on compressed video,'' *IEEE Trans. Circuits Syst. Video Technol.* **5**, 533–544 (1995).
36. A. Alatan *et al.*, ''Image sequence analysis for emerging interactive multimedia services—The European COST 211 framework,'' *IEEE Trans. Circuits Syst. Video Technol.* **8**, 802–813 (1998).
37. J. D. Courtney, ''Automatic video indexing via object motion analysis,'' *Pattern Recogn.* **30**, 607–626 (1997).
38. J. Fan, J. Yu, G. Fujita, T. Onoye, L. Wu, and I. Shirakawa, ''Spatiotemporal segmentation for compact video representation,'' *Signal Process. Image Commun.* **16**, 553–566 (2001).
39. B. Gunsel, A. M. Ferman, and A. M. Tekalp, ''Temporal video segmentation using unsupervised clustering and semantic object tracking,'' *J. Electron. Imaging* **7**, 592–604 (1998).
40. Y. Xu and E. Uberbacher, ''2D image segmentation using minimum spanning trees,'' *Image Vis. Comput.* **15**, 47–57 (1997).
41. C. Gu and M. C. Lee, ''Semantic segmentation and tracking of semantic video objects,'' *IEEE Trans. Circuits Syst. Video Technol.* **8**, 572–584 (1998).
42. D. P. Huttenlocher, G. Klanderman, and W. Rucklidge, ''Comparing images using the Hausdorff distance,'' *IEEE Trans. Pattern Anal. Mach. Intell.* **15**, 481–491 (1993).
43. M. Swain and D. Ballard, ''Color indexing,'' *Int. J. Comput. Vis.* **7**(1), 11–32 (1991).
44. Y. Rui, A. C. She, and T. S. Huang, ''Modified Fourier descriptors for shape representation—A practical approach,'' in *Proc. First Int. Workshop on Image Databases and Multimedia Search*, Amsterdam (1996).
45. R. Lienhart and W. Effelsberg, ''Automatic text segmentation and text recognition for video indexing,'' *Multimedia Syst.* **8**, 69–81 (2000).
46. Y. Zhong, H. J. Zhang, and A. K. Jain, ''Automatic caption location in compressed video,'' *IEEE Trans. Pattern Anal. Mach. Intell.* **22**, 385–392 (2000).
47. Z. Liu, Y. Wang, and T. Chen, ''Audio feature extraction and analysis for scene classification,'' *J. VLSI Signal Proc. Syst. Signal, Image, Video Technol.* **15**, 61–79 (October 1998).
48. Y. Wang, Z. Liu, and J.-C. Huang, ''Multimedia content analysis,'' *IEEE Trans. Signal Process.* **40**, 12–36 (November 2000).
49. A. Gionis, P. Indyk, and R. Motwani, ''Similarity search in high dimensions via hashing,'' *Proc. VLDB*, Edinburgh, pp. 518–529 (1999).
50. MPEG Requirements Group, MPEG-7 context, objectives and technical roadmap, Document ISO/IEC JTC1/SC29/WG11 N2729, Seoul (March 1999).
51. P. Salembier, R. Qian, N. O'Connor, P. Correia, I. Sezan, and P. van Beek, ''Description schemes for video programs, users and devices,'' *Signal Process. Image Commun.* **16**, 211–234 (2000).
52. A. Beritez, S. Paek, S.-F. Chang, A. Puri, Q. Huang, J. R. Smith, C.-S. Li, L. D. Bergman, and C. N. Judice, ''Object-based multimedia content description schemes and applications for MPEG-7,'' *Signal Process. Image Commun.* **16**, 235–269 (2000).

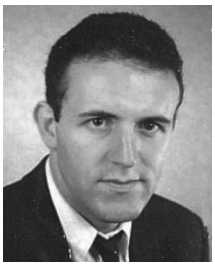
Jianping Fan received his MS degree in theoretical physics from Northwestern University in 1994, and his PhD in optical storage and computer science from Shanghai Institute of Optics and Fine Mechanics, Chinese Academy of Sciences, in 1997. He spent a half year at the Department of Computer Science, Fudan University, Shanghai, as a researcher, and one and a half years at the Department of Information System Engineering, Osaka University, as a JSPS researcher. From 1999 to 2001, he was a researcher at the Department of Computer Science, Purdue University, West Lafayette, Indiana. He is now an assistant professor at Department of Computer Science, University of North Carolina at Charlotte. His research interests include image processing, computer vision, video content computing, indexing and security.



Walid G. Aref received his PhD degree in computer science from the University of Maryland, College Park, Maryland, in 1993. Since then, he has been working at Matsushita Information Technology Laboratory and the University of Alexandria, Egypt. Currently, Dr. Aref is an associate professor at the Department of Computer Science, Purdue University. His current research interests include efficient query processing and optimization algorithms and data mining in spatial and multimedia database. He is a member of the IEEE Computer Society and the ACM.



Ahmed K. Elmagarmid received a Presidential Young Investigator award from the National Science Foundation, and distinguished alumni awards from Ohio State University and the University of Dayton in 1988, 1993, and 1995, respectively. Professor Elmagarmid is the Editor-in-Chief of *Distributed and Parallel Databases: An International Journal* and of the book series on *Advances in Database Systems*, and serves on the editorial boards of *IEEE Transactions on Knowledge and Data Engineering, Information Sciences* and *Journal of Communications Systems*. He has served on the editorial boards of *IEEE Transactions on Computers* and the *IEEE Data Engineering Bulletin*. He is on the steering committees for the IEEE International Conference on Data Engineering and the IEEE Symposium on Research Issues in Data Engineering and has served on the organizational committees of several international conferences. Professor Elmagarmid is the Director of the Indiana Center for Database Systems (ICDS) and the newly formed Indiana Telemedicine Incubator. His research interests are in the areas of video databases, multidatabases, data quality and their applications in telemedicine and digital government. He is the author of several books on databases and multimedia. He has broad experience as an industry consultant and/or adviser to Telcordia, Harris, IBM, MCC, UniSql, MDL, BNR and others. Professor Elmagarmid received his BS degree in Computer Science from the University of Dayton and his MS and PhD degrees from the Ohio State University in 1977, 1981, and 1985, respectively. He has served as a faculty member at the Pennsylvania State University from 1985 to 1988 and has been with the Department of Computer Science at Purdue University since 1988. He is a senior member of the IEEE.



Mohand-Said Hacid graduated as an engineer in computer science from the University of Tizi-Ouzou, Algeria, in 1987, and received his PhD degree in computer science from the National Institute of Applied Sciences, Lyon, France, in 1991. He is currently a professor at the University Claude Bernard Lyon 1, Lyon, France. He has been a visiting researcher at the Theoretical Computer Science Laboratory, Aachen University of Technology, Germany, and at the Indiana Center for Database Systems, Purdue University. His research interests include knowledge representation and reasoning, data models and query languages for multimedia databases and semistructured databases.

Mirette S. Marzouk received her BS and MS degrees in computer science from Alexandria University, Egypt. She is a senior programmer at the Department of Computer Science, Purdue University. Her major research interests are image processing and video database systems.

Xingquan Zhu received his PhD degree in computer science from Fudan University, Shanghai, China, in 2001, and BS and MS degrees from Xidian University, Shannxi, China, in 1995 and 1998, respectively. Currently, he is a postdoctoral research assistant at the Department of Computer Science, Purdue University. His research interests include image processing, video processing, content based image/video retrieval and video databases. He received SIEMENS and INTEL scholarships in 1999 and 2000, respectively, for his PhD thesis research on key techniques of content-based video retrieval.