

# A Hierarchical Access Control Model for Video Database Systems

ELISA BERTINO

University of Milano

JIANPING FAN

University of North Carolina

ELENA FERRARI

University of Insubria

MOHAND-SAID HACID

University Claude Bernard Lyon 1

AHMED K. ELMAGARMID

Hewlett Packard

and

XINGQUAN ZHU

Purdue University

---

Content-based video database access control is becoming very important, but it depends on the progresses of the following related research issues: (a) efficient video analysis for supporting semantic visual concept representation; (b) effective video database indexing structure; (c) the development of suitable video database models; and (d) the development of access control models tailored to the characteristics of video data. In this paper, we propose a novel approach to support multilevel access control in video databases. Our access control technique combines a video database indexing mechanism with a hierarchical organization of visual concepts (i.e., video database indexing units), so that different classes of users can access different video elements or even the same video element with different quality levels according to their permissions. These video elements, which, in our access control mechanism, are used for specifying the authorization objects, can be a semantic cluster, a subcluster, a video scene, a video shot, a video frame, or even a salient object (i.e., region of interest). In the paper, we first introduce our techniques for obtaining these multilevel

---

This project was supported by National Science Foundation under grants 0208539-IIS 9972883-EIA, 9974255-IIS, and 9983249-EIA, and by grants from HP, IBM, Intel, NCR, Telcordia, and CERIAS. This project was also supported by a grant from the AO Foundation, Switzerland.

Authors' addresses: E. Bertino, Dipartimento di Scienze dell'Informazione, University of Milano, Milan Italy; email: bertino@dsi.unimi.it; J. Fan, Department of Computer Science, University of North Carolina, Charlotte, NC 28223; email: jfan@uncc.edu; E. Ferrari, Dipartimento di Scienze Chimiche, Fisiche e Matematiche, University of Insubria, Como, Italy; email: elena.ferrari@uninsubria.it; M.-S. Hacid, LISI-UFR d'Informatique, University Claude Bernard Lyon 1, Villeurbanne, France; email: mshacid@bat710.univ-lyon1.fr; A. K. Elmagarmid, Hewlett Packard, Palo Alto, CA 94304; email: ahmed.elmagarmid@hp.com; X. Zhu, Department of Computer Science, Purdue University, West Lafayette, IN 47907; email: xqzhu@c.s.purdue.edu.

Permission to make digital/hard copy of part or all of this work for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication, and its date appear, and notice is given that copying is by permission of ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or a fee.

© 2003 ACM 1046-8188/03/0400-0155 \$5.00

video access units. We also propose a hierarchical video database indexing technique to support our multilevel video access control mechanism. Then, we present an innovative access control model which is able to support flexible multilevel access control to video elements. Moreover, the application of our multilevel video database modeling, representation, and indexing for MPEG-7 is discussed.

Categories and Subject Descriptors: I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding—*Video analysis*; D.4.6 [**Operating Systems**]: Security and Protection—*Access control*; H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Indexing methods*

General Terms: Security

Additional Key Words and Phrases: Video database models, access control, indexing schemes

---

## 1. INTRODUCTION

Large collections of video data play today a major role on new types of networked information systems. In recent years, there has been a growing interest in developing more effective methods for retrieving and browsing video databases over networks. Several content-based video database systems have been proposed, such as QBIC, VideoQ, MARS, Netra-V, Virage, Photobook, Name-It, and MultiView [Flickner et al. 1995; Pentland et al. 1996; Rui et al. 1998; Humrapur et al. 1997; Chang et al. 1998; Satoh and Kanade 1997; Deng and Manjunath 1998; Fan et al. 2001a]. In order to support video retrieval, most of those systems first partition videos into a set of access units such as shots, objects, or regions, and then follow the approach of representing video via a set of feature attributes, such as color, texture, shape, and layout [Fan et al. 2001a; Zhang et al. 1997; Jain et al. 1999; Tamura et al. 1978]. Those high-dimensional visual features are properly indexed, according to some indexing structures, and are then used for video retrieval. A common shortcoming of all those video retrieval systems, however, is the lack of suitable access control mechanisms. The development of such a mechanism is increasingly relevant because of the value which is today associated with information, in whatever form is represented. We know very well that not all information is intended for every person; this is especially true in the business world. Video data are used in a large variety of environments with different goals and usages. Examples of these application environments include medical applications, e-learning systems and tools, environmental protection, manufacturing processes, scientific research. Moreover, the performance of the video database access control schemes largely depends on the progresses of the following related research areas:

- (1) A good video database access control scheme should be able to control user accesses more effectively based on visual concepts because video database users prefer to access video database via semantic visual concepts rather than via low-level visual features. As mentioned above, videos in databases are normally characterized and indexed by their low-level visual features that can be extracted automatically; thus it is important to map the low-level visual features to the relevant high-level visual concepts. Unfortunately, there is a *semantic gap* between low-level visual features and

semantic visual concepts. In order to support more efficient access control in video database systems, it is crucial to support more effective video representation and visual concept characterization, so that the semantic visual concepts can be characterized more effectively by using the suitable discriminating low-level visual features.

- (2) A good video database access control scheme should be integrated with the database indexing structure, so that video database access control can be achieved more effectively. Since video database access control schemes should exploit semantic visual concepts and not low-level visual features; these database indexing units should correspond to the relevant semantic visual concepts. Traditional database indexing structures are, however, not suited for supporting video database access control. The reasons are twofold: (a) visual features are normally in high dimensions but the traditional indexing structures suffer from the problem of *curse of dimensionality*; (b) hierarchical video database indexing structure should be able to represent the concept hierarchy of video contents, so that each database indexing unit corresponds to one specific visual concept.
- (3) A good video database access control scheme should be combined with the video database model effectively. The video database model should be able to tell us what kind of visual concepts should be detected, how we can organize them to support more effective video database management and access control, and which access control rules should be applied on these visual concepts.

Based on the above observations, a novel multilevel video database access control technique is proposed in this paper. In order to support multilevel user-adaptive video database access control, we integrate a hierarchical video indexing mechanism with a semantic video classifier. Each node in the hierarchical video database indexing structure represents one specific semantic visual concept. The contextual and logical relationships among these semantic visual concepts (i.e., nodes on the database indexing structure) are organized by a domain-dependent concept hierarchy. Under our multilevel access control model, users can access the video database semantically according to different granularities, such as clusters, subclusters, video sequences, scenes, shots, frames, or even regions of interest, according to the authorizations they are given. The remainder of this paper is organized as follows. Section 2 discusses related work. Section 3 proposes a hierarchical video database model supporting multilevel video representation, indexing, and access control. Section 4 introduces our approach to video analysis for obtaining video access control units such as shots and regions of interest. A novel semantic video classification algorithm is proposed in Section 5 for obtaining the video access control units at the database layer, such as semantic clusters and subclusters. Section 6 proposes a hierarchical video database indexing structure to support our multilevel access control mechanism. Section 7 presents our video access control mechanism and its performance. Section 8 describes applications of our multilevel video database representation and indexing structures to MPEG-7. Section 9 concludes the paper by outlining future work.

## 2. RELATED WORK

Several efforts have been reported to extend conventional database access control models to deal with new data types and to provide new functions in authorization management. Such efforts include authorization models for object-oriented databases, and for Web pages, temporal authorization models, and extended authorization models for relational databases [Bertino et al. 1998; Bertino et al. 1990; Fernandez et al. 1994; Samarati et al. 1990]. These models are not, however, fully adequate for the protection of information in a video database system, since they do not take into account the complex structure of video data, the need for content-based video retrieval, and the need for access modes different from the conventional ones (i.e., read, write, execute). Content-based authorizations for video data objects is novel and has never been addressed before. Also, how to support varying protection granularity levels for video data objects has not been addressed before. The only two approaches we are aware of have been proposed by Kumar and Babu [1998] and by Bertino et al. [2000]. The approach by Kumar and Babu [1998] is, however, very primitive in that it only allows one to hide some frames for specific classes of users; it neither supports content-based authorizations nor does it allow one to hide part of a frame. The work by Bertino et al. [2000] provides access control units with different levels of granularity and also supports content-based access control, through the use of textual annotations associated with video object. Such an approach, however, does not actually support access control based on visual features. Supporting video access control based on visual features is becoming attractive and important because most existing video retrieval systems rely on the use of visual features. Moreover, the approach by Bertino et al. [2000] does not address the development of access structures specifically tailored for a hierarchical access control model. Finally, the current work also casts the proposed approach in the framework of MPEG-7, whereas the work by Bertino et al. [2000] does not consider this standard. In addition to the work mentioned above, there has been some work on specifying authorizations with different levels of granularity and on content-based authorizations, which we discuss below. A proposal for a content-based access control for textual Digital Libraries (DLs) has been recently proposed in Adam et al. [2002]. In such an approach, authorizations are based on concepts associated with textual data objects. Concepts to be associated with a given set of documents are identified by means of a document classification system [Holowczak 1997], which is based on information extraction techniques. Such an approach also supports a two-level granularity for authorization objects by which authorizations can be associated either with a whole document or with portions of it (called *slots*). Slots must be manually marked by the Security Administrator or some other users through some slot-identifiers. A main difference with our proposal is that this model has no provision for video access control and therefore the object granularity the model supports is very limited. By contrast, our model provides a more articulated object granularity model deriving from the need of supporting video data. Such data can be organized, from the point of view of access control, according to several hierarchical levels, such as semantic clusters, subclusters,

entire videos, logical segments, frames, parts of frames. The second difference is that access modes required for video data objects are different with respect to those used for textual documents. For video data objects one must provide access modes such as query, browsing, as well as the possibility of browsing for a specified duration (such as the first 10 min of the videos).

### 3. HIERARCHICAL VIDEO DATABASE MODEL

Most existing content-based video retrieval systems use one of two widely-accepted approaches to access videos in a database [Flickner et al. 1995; Pentland et al. 1996; Rui et al. 1998; Humrapur et al. 1997; Chang et al. 1998; Satoh and Kanade 1997; Deng and Manjunath 1998; Fan et al. 2001a]: *shot-based* and *object-based*. These video retrieval systems, however, only focus on how to obtain these video access units (i.e., video shots and video objects), whereas few of them reveal or publish their video database models and indexing structures. When very large video data sets come into view, video database models and indexing can no longer be ignored if one wants to support effective video retrieval and access control. An obvious solution to video database indexing is to use the traditional high-dimensional indexing trees [Guttman 1984; Berchtold et al. 1996; White and Jain 1996; Lin et al. 1995; Chakrabarti and Mehrotra 1999]. However, this approach suffers from the problem of *curse of dimensionality* because the visual features used for video representation are normally in high dimensions. One reasonable solution to this problem is first to classify videos into a set of clusters and then to perform the dimensionality reduction on these clusters independently [Fan et al. 2001a; Zhang et al. 1997]. Traditional database indexing trees can supposedly be used for indexing these video clusters independently, with relatively low-dimensional features. However, the pure feature-based clustering techniques are unsuitable for video classification because of the semantic gap [Baraani-Dastjerdi et al. 1997; Del Bimbo et al. 1995; Yeo and Yeung 1997; Zhong et al. 1996; Vailaya et al. 1999; Minka and Picard 1997; Ortega et al. 1998; Rui and Huang 1999; Ishikawa et al. 1998; Huang et al. 1998; Sheikholeslami et al. 1998]. Decision-tree-based classifiers are very attractive for video classification via learning from the labeled training examples [Quinlan 1986], but the hundreds or thousands of internal nodes typical of these classifiers do not make sense with respect to the video database indexing. A semantic video classifier is expected not only to be efficient in bridging the semantic gap but also in providing an effective video database indexing and access control scheme. Thus, the structure of the semantic video classifier should be related to the video database management model. Unfortunately, no existing work has discussed what kind of *database model* should be used for managing large-scale video collections. In order to provide a video database model suitable for supporting more effective video database access control based on semantic visual concepts, we classify video shots into a set of hierarchical database units as shown in Figure 1. We need, however, to address the following key problems: (a) How many *levels* should be included in such a video database model and how many *nodes* should be included in each level? (b) Do the nodes in the hierarchy make sense to

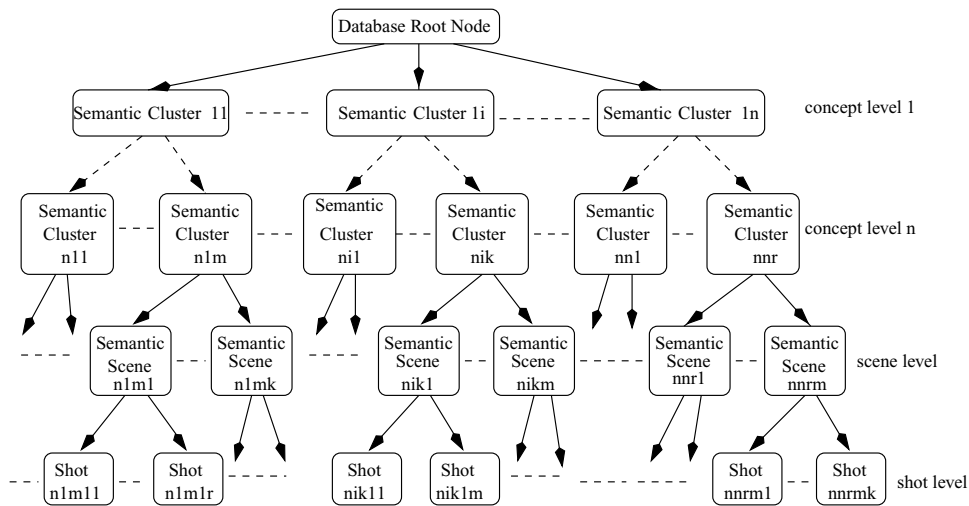


Fig. 1. The proposed hierarchical video database model, where a cluster may include several levels according to the concept hierarchy. Clusters at level  $l + 1$  refine clusters at level  $l$  ( $1 \leq l \leq n - 1$ ).

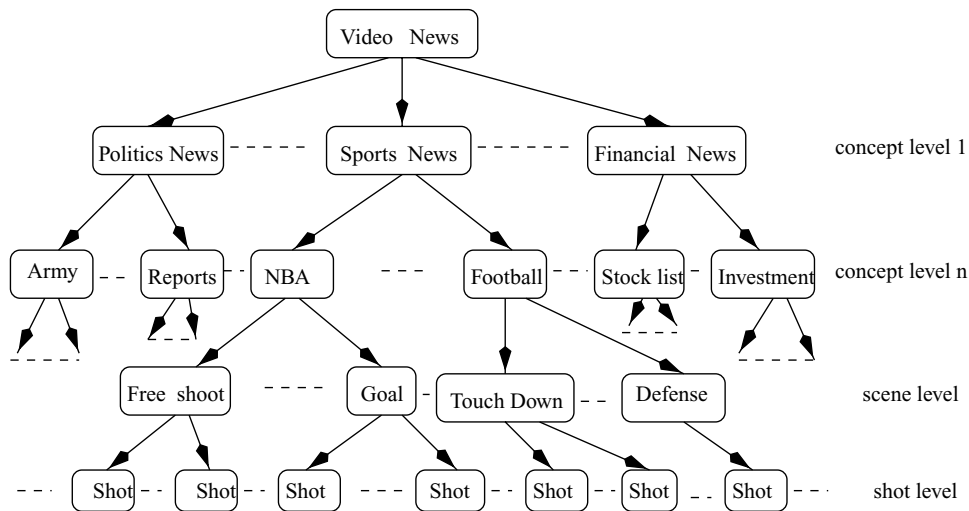


Fig. 2. The concept hierarchy for News used in our system.

human beings? In order to support hierarchical video browsing and access control, the interpretations of the nodes should be meaningful for human beings. (c) What kind of *discriminating visual features* should be selected and what kind of *classification rules* should be used for each kind of node? We solve the first and second problem by deriving the database model from the concept hierarchy of video contents. Obviously, the concept hierarchy is domain-dependent. A video News example is given in Figure 2. The concept hierarchy defines the contextual and logical relationships between the higher-level and the lower-level visual concepts. In order to support more effective hierarchical video database

access control, our system exploits the following techniques:

- (a) *An efficient video analysis technique* to support more effective high-level visual concept characterization by using the principal video shots and the associated salient objects. The physical video shots are too general to characterize the associated semantic visual concepts. Since the presence or absence of the *salient objects* (i.e., regions of interest) can indicate the presence or absence of the related semantic visual concepts [Fan et al. 2001b, 2001c, 2001d; Wang and Chang 1999; Zhong et al. 2000], salient objects are very attractive for characterizing the semantic visual concepts and supporting semantic video classification.
- (b) *A semantic video classifier* to shorten the semantic gap between the low-level visual features and the high-level semantic visual concepts. The hierarchical structure of our semantic video classifier is provided by domain experts or is obtained using WordNet [Miller et al. 1990; Aslandogan et al. 1997]. Each node in this classifier defines a semantic visual concept; the contextual and logical relationships between the higher-level nodes and the lower-level nodes are organized by the domain-dependent concept hierarchy. Different visual features capture different aspects of visual perception. Thus, the video database management units (i.e., semantic visual concepts) shown in Figure 1 are indexed by their discriminating visual features with different significance. Note that the goal of semantic video classification is not to understand videos in the way that human beings do [Li et al. 2000], but to classify the unlabeled video clips to the known semantic visual concepts defined by the concept hierarchy so that more efficient access control structures can be supported.
- (c) *A hierarchical video database management and visual summary organization technique* to support more effective video database access control. The video database management structure (i.e., video database model) is provided by the domain-dependent concept hierarchy which is also used for determining the structure of the semantic video classifier. The hierarchical organization of the visual summaries is also integrated with the database indexing structure. For each node  $R$  of the proposed hierarchical video database model, we use the following parameters to represent each node (i.e., visual concept) for indexing:

*semantic label* :  $L_R$ , *subset for discriminating features* :  $\Xi_R$   
*subspace dimensions* :  $D_R$ , *feature weights* :  $\theta_{\mathbf{R}} = (\theta_1, \dots, \theta_{D_R})$   
*feature parameters* : *mean*  $\mu_{\mathbf{R}} = (\mu_1, \dots, \mu_{D_R})$ , *variance*  $\sigma_{\mathbf{R}} = (\sigma_1, \dots, \sigma_{D_R})$   
*concept seeds* :  $\{ST_1, \dots, ST_m\}$   
*access rules* :  $\mathfrak{R}_R$

where  $L_R$  is the semantic label (i.e., visual concept defined by the concept hierarchy) for node  $R$ ,  $\Xi_R$  is the subset of its discriminating visual features,  $D_R$  is the dimensions of its discriminating visual features,  $\theta_{\mathbf{R}}$  indicates the weights associated with these discriminating features, and  $\mu_{\mathbf{R}}$  and  $\sigma_{\mathbf{R}}$  are the mean and the variance of the video shots associated with node  $R$ , respectively. The node seeds,  $ST_1, \dots, ST_m$ , which are the principal video

shots (cover all the potential concepts of its sublevel nodes), are used for representing the high-level semantic concept assigned to the corresponding node because of the lack of feature support at the higher semantic level.  $\mathcal{R}_R$  is the set of access rules for the video concept associated with node  $R$ .

#### 4. VIDEO SHOT AND SALIENT OBJECT DETECTION

As mentioned in Section 1, more effective video analysis and feature extraction are very attractive for supporting efficient video database access control. Many pioneering video analysis approaches have been proposed in the past decade [Aslandogan et al. 1997; Meng and Chang 1996; Meng et al. 1995; Liu and Zick 1995; Shen and Sethi 1998; Yeo and Liu 1995; Fan et al. 2000; Zhang et al. 1993; Kobla and Doermann 1998]. Some of these techniques can also work very well on MPEG compressed videos. However, few of the papers describing those techniques mention how the threshold for shot detection is automatically selected and adapted to the activities of various videos or even the different video shots in the same video sequence. In general, threshold setting plays a critical role in automatic video shot detection because the thresholds should be adapted to the activities of video contents. It is impossible to use a universal threshold satisfying various conditions because the activities for various video sequences or even different video shots within the same sequence should be different. To support a more efficient bit rate allocation, we have developed an efficient scene cut detection technique to determine the size of GOP in an adaptive MPEG video encoder [Fan et al. 2000]. Our scene cut detection technique can adapt the thresholds for video shot detection according to the activities of various video sequences, and this technique has been developed to work on MPEG compressed videos [Fan et al. 2001a]. Unfortunately, such a technique still cannot adapt the thresholds for different video shots within the same sequence. In order to adapt the thresholds to the *local activities* of different video shots within the same sequence, we use a small window (i.e., 20 frames in our current work) and the threshold for each window is adapted to its local visual activity by using our automatic threshold detection technique and local activity analysis. The video shot detection results shown in Figures 3, 4, 5, and 6 are obtained from several video data sources used in our system, such as movies and medical videos. The average performance of our adaptive video shot detection techniques are given in Tables I and II. The average performance comparison between the traditional techniques and this proposed method are given in Table III. The *accuracy ratio* is defined as

$$accuracy\_ratio = \frac{Break\_shots + Gradual\_Shots}{Break\_shots + Gradual\_Shots + Missed\_shots}$$

One can find that our video shot detection techniques can obtain better results by adapting the thresholds for shot detection according to the local activities of video contents. Physical video shots are too general to characterize the associated semantic visual concepts. In order to support more efficient video database access control based on the semantic visual concepts, we have developed several object-specific functions to detect the well-defined salient objects from these physical video shots [Fan et al. 2001b, 2001d]. In the domain of medical



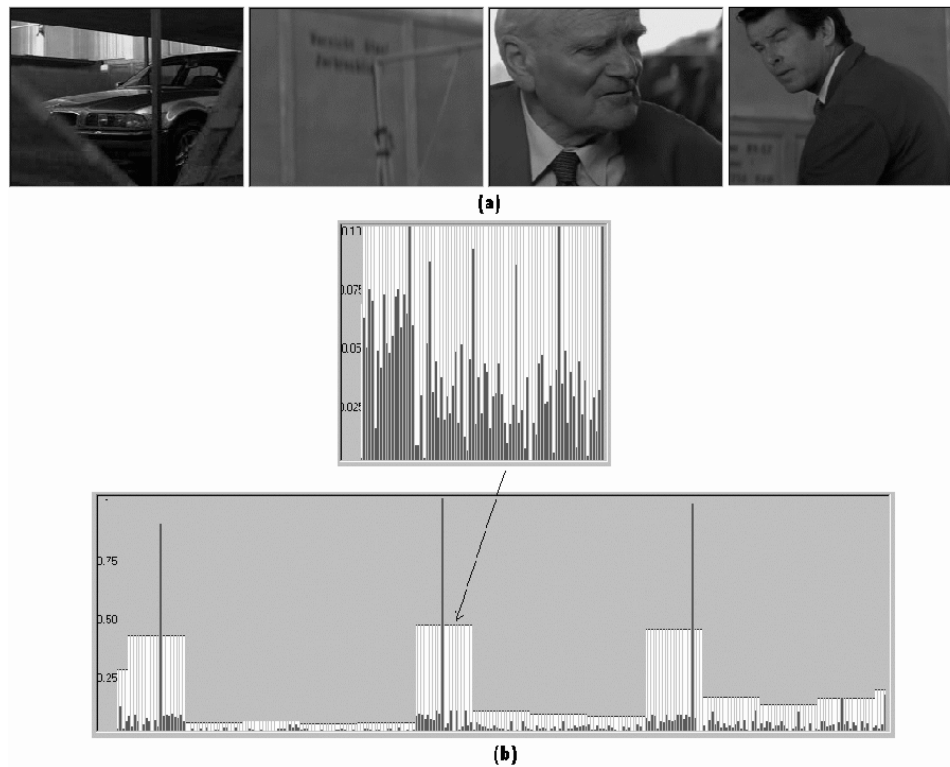


Fig. 3. Video shot detection results from a movie: (a) part of the detected scene cut frames; (b) the corresponding color histogram difference and the determined thresholds for different video shots, where the values of color histogram difference in a small window are also given.

education videos, the presence of the semantic scenes, such as *presentation*, *dialog*, *surgery*, and *diagnosis*, can be determined by the presence of the salient objects, such as *slides*, *human faces*, *blood-red regions*, and *skin regions*. Salient object detection results, human faces and blood-red regions from a news video and a medical education video, are given in Figures 7 and 8, respectively. Physical video shots, which consist of any type of these well-defined salient objects, are taken as the principal video shots for characterizing the semantic visual concepts associated with the corresponding MPEG video. Therefore, the semantic visual concepts in a database are characterized and indexed via these principal video shots by using their shot-based and object-based visual features.

## 5. SEMANTIC VIDEO SHOT CLASSIFICATION

After the principal video shots and their visual features are obtained, we focus on generating semantic scenes and higher-level visual concepts such as semantic clusters, so that more effective database indexing and access control scheme can be supported. The semantic classifier is built in a bottom-up fashion as shown in Figure 9. As mentioned in Section 3, the hierarchical structure of the classifier, that is, *levels* and *nodes*, is first determined according to the concept

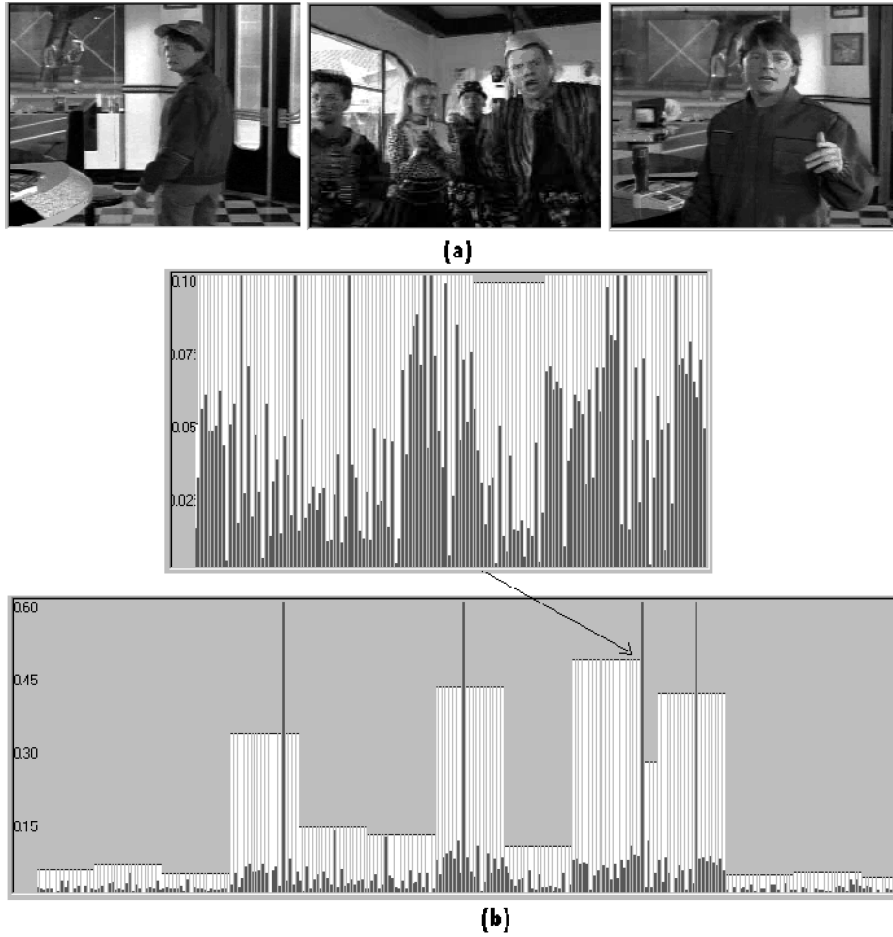


Fig. 4. Video shot detection results from a movie: (a) part of the detected scene cut frames; (b) the corresponding color histogram difference and the determined thresholds for different video shots, where the values of color histogram difference in a small window are also given.

hierarchy of video contents and is given by the domain experts or obtained via WordNet [Miller et al. 1990; Aslandogan et al. 1997]. Once such video database management structure is given, we use a set of labeled training examples to determine the discriminating visual features for each node. A labeled training example is in terms of a set of low-level visual features and the semantic label for the corresponding node. There are two measures for defining the similarity among the labeled training examples: (a) *visual similarity* via comparing their low-level visual features; (b) *semantic similarity* via comparing their high-level semantic labels. The feature-based similarity distance  $D_F(T_\delta, T_\gamma)$  between two video shots  $T_\delta$  and  $T_\gamma$  is defined as

$$D_F(T_\delta, T_\gamma) = \sum_{F_l \in \Xi} \frac{1}{\alpha_l} \cdot D_{F_l}(T_\delta, T_\gamma), \quad \sum_{l=1}^n \frac{1}{\alpha_l} = 1, \quad (1)$$

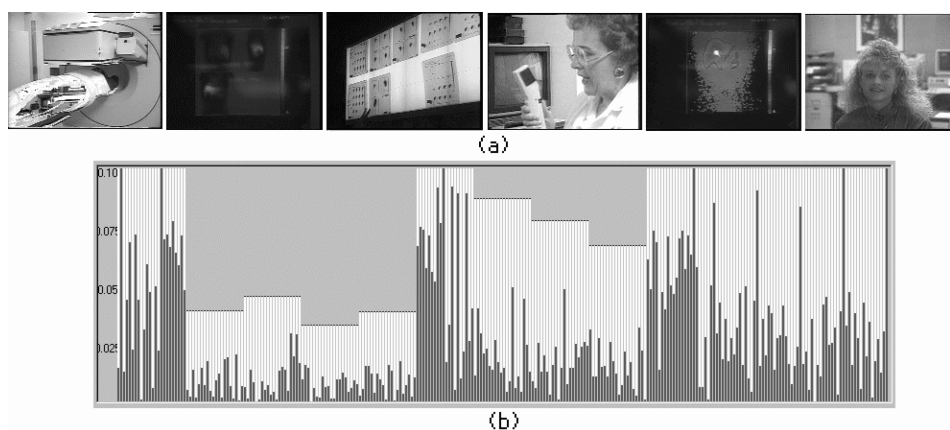


Fig. 5. Video shot detection results from a medical video: (a) part of the detected scene cut frames; (b) the corresponding color histogram difference and the determined thresholds for different video shots.

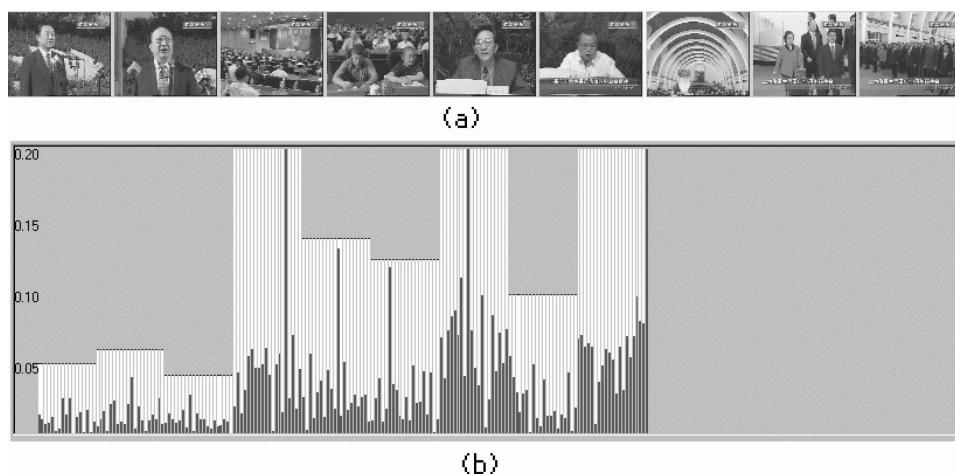


Fig. 6. Video shot detection results from a video News: (a) part of the detected scene cut frames; (b) the corresponding color histogram difference and the determined thresholds for different video shots.

where  $D_{F_l}(T_\delta, T_\gamma)$  denotes the similarity distance between  $T_\delta$  and  $T_\gamma$  according to their  $l$ th feature  $F_l$ ,  $\alpha_l$  is the weight for the  $l$ th feature,  $\Xi$  is the set of original visual features, and  $n$  is the total number of features which are initially selected for video shot representation. The semantic similarity distance  $D_S(T_\delta, T_\gamma)$  between two principal video shots  $T_\delta$  and  $T_\gamma$  can be defined as

$$D_S(T_\delta, T_\gamma) = \begin{cases} 0, & L_\delta = L_\gamma \\ 1, & \text{otherwise,} \end{cases} \quad (2)$$

where  $L_\delta$  and  $L_\gamma$  are the semantic labels for the principal video shots  $T_\delta$  and  $T_\gamma$ , respectively. Our hierarchical semantic video classifier focuses on bridging

Table I. The Average Performance of Our Video Shot Detection Technique for News Video Sequences

Test videos	news1.mpg	news2.mpg	news3.mpg
Frame numbers	517	6288	7024
Break shots	86	98	129
Gradual shots	6	11	7
Missed shots	4	7	15
Accuracy ratio	97.8%	93.9%	91.3%

Table II. The Average Performance of Our Video Shot Detection Technique for Medical Video Sequences

Test videos	med1.mpg	med2.mpg
Frame numbers	33200	15420
Break shots	116	57
Gradual shots	21	48
Missed shots	6	9
Accuracy ratio	95.8%	92.1%

Table III. The Average Performance Comparison

Test videos	med1.mpg	news2.mpg
Frame numbers	33200	6288
Zhang [Zhang et al. 1993]	92.5%	96.9%
Yeo [Yeo and Liu 1995]	91.8%	94.7%
Proposed method	95.8%	97.8%



Fig. 7. The detected salient objects (i.e., human faces) from a video news: original image versus salient object.

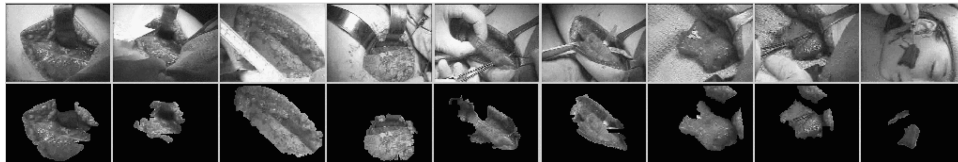


Fig. 8. The detected salient objects (i.e., blood-red regions) from a medical education video: original image versus salient object.

the gap between these two measures, so that the visual similarity can correspond to the semantic similarity by selecting the discriminating visual features with suitable relevance. We first use a set of labeled training examples to select the discriminating visual features and their relevance for each node. The Lagrangian optimization technique is then used for obtaining these

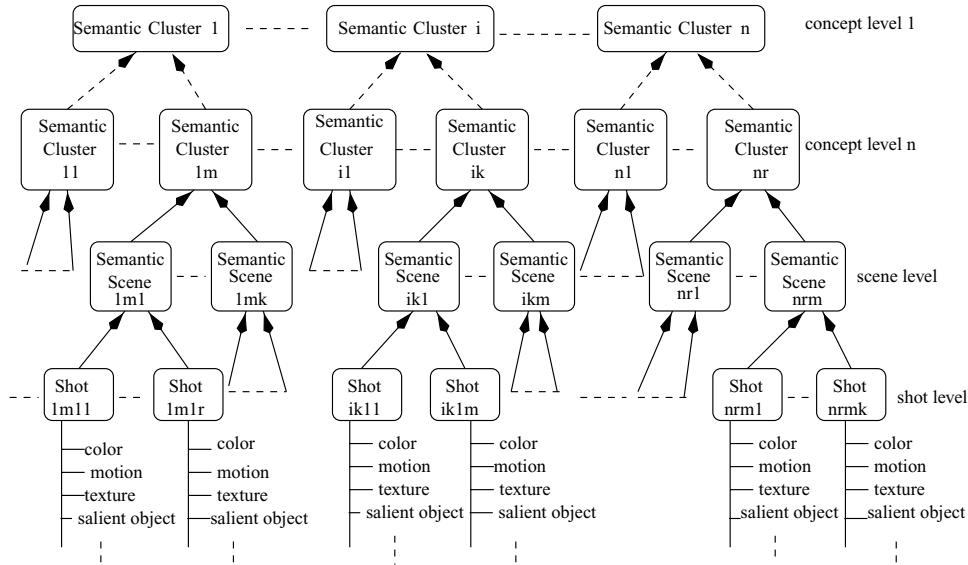


Fig. 9. The bottom-up procedure for building the hierarchical video classifier, where the semantic cluster may include  $n$  levels according to the concept hierarchy.

dimensional weights over the training data set  $\Omega$  [Minka and Picard 1997; Ortega et al. 1998; Rui and Huang 1999]:

$$\min \left\{ D_F(T_\delta, T_\gamma) = \sum_{F_l \in \Xi} \frac{1}{\alpha_l} \cdot D_{F_l}(T_\delta, T_\gamma) \right\}, \text{ subject to: } \sum_{l=1}^n \frac{1}{\alpha_l} = 1. \quad (3)$$

Given the dimensional weighting coefficients  $\{\frac{1}{\alpha_1}, \frac{1}{\alpha_2}, \dots, \frac{1}{\alpha_n}\}$  for a semantic cluster, the degree of importance of its  $n$ -dimensional visual features is also given. Higher-dimensional weighting coefficients mean that the associated visual features are more important in predicting the judgment of similarity by humans. On the other hand, cluster variance is a good quality measure of the distributions and independence of the similar principal video shots in the same cluster. Small value of variance means that all the similar principal video shots in the same cluster are distributed more densely. Large value of the variance indicates that the similar principal video shots in the same cluster are sparsely distributed. From the quality of video classification point view, a small cluster variance is preferred so that the cluster only consists of the similar principal video shots. From the indexing point of view, a sparse distribution of similar principal video shots in the same cluster is preferred so that they can be distinguished efficiently. A good trade-off between these two issues should then be found. Our approach is to first reduce the cluster variance below a threshold so that the cluster only consists of the similar principal video shots, and then to select the “principal” dimensions (i.e., with high-dimensional weights) with large variance so that the similar principal video shots in the same cluster can be efficiently separated to support more effective indexing. Therefore, the

discriminating visual features, which have larger weighting coefficients (more important) and larger dimensional variance (data points are distributed more sparsely with respect to them), can be selected. Once the semantic video classifier is in place, it is used to semantically classify the unlabeled videos. The principal video shots are first extracted from the unlabeled videos by using our adaptive video shot detection and salient object extraction techniques. The task of video shot classification can then be summarized as follows: Given an unlabeled principal video shot  $q$  (obtained from the unlabeled video) and its  $n$ -dimensional feature values  $X$ , it is first assigned to the best matched semantic cluster  $C_i$  that corresponds to the minimum similarity distance  $D_F(X, \mu_{c_i}, \sigma_{c_i})$ . The similarity distance  $D_F(X, \mu_{c_r}, \sigma_{c_r})$  between the unlabeled principal video shot  $q$  and the semantic cluster  $C_r$  in its feature subspace is calculated as

$$D_F(X, \mu_{c_i}, \sigma_{c_i}) = \sum_{l=1}^{D_{c_r}} \frac{1}{\alpha_{c_{r_l}}} \cdot D_{F_l}(x_l, \mu_{c_{r_l}}), \quad (4)$$

where  $D_{F_l}(x_l, \mu_{c_{r_l}})$  is the similarity distance between the unlabeled principal video shot and the semantic cluster  $C_r$  according to their  $l$ th visual feature. If

$$D_F(X, \mu_{c_r}, \sigma_{c_r}) \leq \sum_{l=1}^{D_{c_r}} \sigma_{c_{r_l}}, \quad (5)$$

then the semantic cluster  $C_r$  is selected as a candidate. Similarly, we can also get other potential semantic clusters to which the unlabeled principal video shot also belongs to. The unlabeled principal video shot is classified into the semantic cluster  $C_i$  which has the minimum similarity distance. Subsequently, the unlabeled principal video shot is further classified into the best matched semantic sublevel clusters and scene in a top-down fashion. Along with each step of classification of a new video shot, for each corresponding high-level unit (i.e., a cluster) in the classifier, we update the means and variances. High-level semantic concepts (i.e., cluster levels) may have lower feature support; thus it is still hard to use the cluster mean and variance to characterize the average properties of the principal video shots in the same cluster. In order to avoid this problem, we have identified a number of principal video shots for each high-level node and these principal video shots are used for node representation and are also taken as seeds for video classification and retrieval. The basic requirement of seed selection is that the selected principal video shots (i.e., seeds) for the higher level node should cover all the potential visual concepts in its sublevel nodes. Given an unlabeled principal video shot  $T = \{x_1, x_2, \dots, x_n\}$  characterized by  $n$ -dimensional visual features, the classification is performed as

- (a) First, the classifier tries to find the best matched seed from each semantic cluster. Since there are several node seeds for each semantic cluster, the weighted feature-based similarity distance  $D_F(T, ST_{C_r}^j)$  between the query shot  $T$  and the  $j$ th seed of the cluster  $C_r$  (i.e.,  $ST_{C_r}^j$ ) is calculated:

$$D_F^i(T, ST_{C_r}^j) = \sum_{h=1}^{D_r} \frac{1}{\beta_h} D_{F_h}(T, ST_{C_r}^j), \quad (6)$$



Fig. 10. The semantic scene *Bone* generated from medical education videos.

where  $D_{F_h}(T, ST_{C_r}^j)$  is the similarity distance between the query shot  $T$  and the  $j$ th seed of cluster  $C_r$  on the basis of their  $h$ th dimensional features. The best matched seed in the corresponding cluster  $C_r$  can be determined as

$$D_F(T, ST_{C_r}^l) = \min\{D_F^r(T, ST_{C_r}^j) | ST^j \in \{ST_1, \dots, ST_m\}\}. \quad (7)$$

(b) Second, the classifier finds the best matched cluster:

$$D_F(T, C_i) = \min\{D_F(T, ST_{C_r}^l) | C_r \in \{C_{11}, \dots, C_{1n}\}\}. \quad (8)$$

(c) Third, the classifier can find the best matched sublevel clusters, and the best matched group by using an approach similar to the approach used in step (b). Examples of video classification results are shown in Figures 10, 11, and 12. The performance of our video shot classifier also depends on the size of the training data set. A large training data set often increases

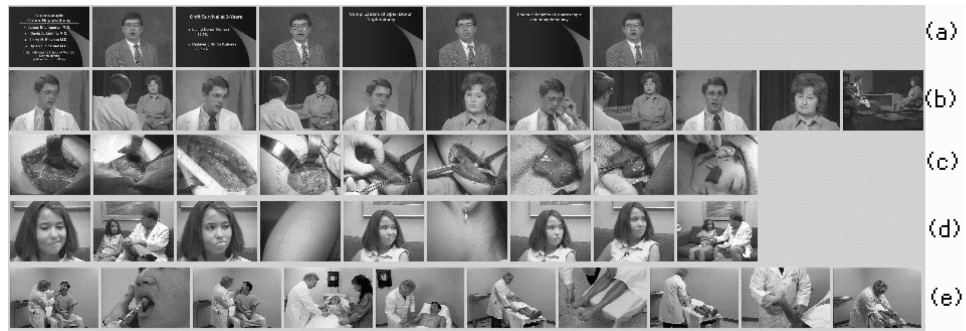


Fig. 11. The semantic scenes generated from *medical video*: (a) presentation; (b) dialog; (c) surgery; (d) diagnosis; (e) diagnosis.



Fig. 12. The semantic scene of *surgery* generated from *medical video*.

the accuracy of the classification as shown in Figure 13. The limited size of the training data set for a node depends on the dimensions of its discriminating visual features which are used for characterizing the corresponding visual concept. The average performance of our semantic video classifier is also been tested for various video sources. The average performance of our semantic video classifier is given in Table IV. The classification *accuracy*



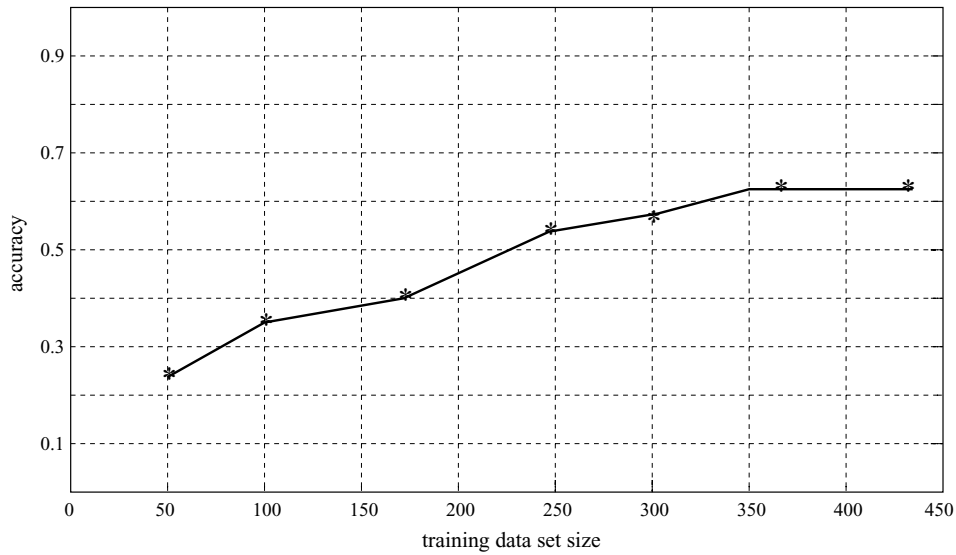


Fig. 13. Classification accuracy based on different training data sizes.

Table IV. The Average Performance of Our Semantics-Sensitive Video Classifier

Test data types	Test data numbers (shots)	Concept levels	Accuracy ratio
Medical Videos	1508	3	64.5%
News Videos	1800	3	60.3%
Movie Videos	4620	4	58.4%

*ratio* is defined as

$$accuracy\_ratio = \frac{total\_shots - misclassified\_shots}{total\_shots},$$

where *total\_shots* indicates the total number of video shots which are relevant to the corresponding visual concept, and *misclassified\_shots* denotes the number of video shots which are relevant to the corresponding visual concept but classified into other visual concept and the video shots which are irrelevant to the corresponding visual concept but classified into the corresponding visual concept. One-level and two-state image classification techniques [Vailaya et al. 1999; Minka and Picard 1997; Ortega et al. 1998; Rui and Huang 1999; Ishikawa et al. 1998; Huang et al. 1998; Sheikholeslami et al. 1998; Manolopoulos et al. 2000; Guttman 1984; Berchtold et al. 1996; White and Jain 1996; Lin et al. 1995; Chakrabarti and Mehrotra 1999; Quinlan 1986; Li et al. 2000] can reach an accuracy higher than 90%. As compared with these traditional one-level and two-state semantic image classification techniques, one can find that the accuracy ratio for our semantic video classifier is not good as we expected. The reasons are: (a) There is a semantic gap between the low-level visual features and the high-level semantic visual concepts, and the selected visual features may be unsuitable for characterizing the associated semantic

visual concepts. (b) The accuracy of our semantic video classifier also depends on the data distribution of the training set and some selected training samples may be irrelevant to the semantic visual concept or the semantic labels for these training samples are incorrect. (c) The accuracy of our semantic video classifier also depends on the size of the training data set but it is too expensive to obtain large-scale training samples. (d) Our semantic video classifier focuses on the problem for multiple levels (i.e., concept hierarchy) and multiple states (i.e., each cluster includes multiple subclusters); thus the variance of the low-level visual features for the semantically similar principal video shots should be large and thus it decreases the performance. (f) The semantically similar principal video shots may consist of the similar salient objects with very different background.

## 6. VIDEO INDEXING TOWARD MULTILEVEL ACCESS CONTROL

After the semantic clusters with their discriminating visual features have been obtained, another key issue is how to support cluster-based hierarchical video database indexing, so that multilevel access control can be efficiently performed. Both video classification and indexing have been intensively researched, but these two topics have been investigated separately for different optimization objectives. We now investigate how video classification and indexing can be effectively combined to support more efficient multilevel access control. Note that we are not proposing a new indexing technique. Instead, we show how our semantic video classification technique can lead to an extremely simple indexing structure that performs very well for multilevel access control. Video contents are first partitioned into a set of semantic clusters as shown in Figure 14; each semantic cluster is then partitioned into a set of subclusters and each subcluster may consist of a set of subregions. Such hierarchical partition of a semantic cluster ends when the number of multidimensional data points in each subregion is less than a predefined threshold:  $\log N \ll D_i$ , where  $N$  is the total number of multidimensional data points in the subregion, and  $D_i$  is the number of dimensions of the discriminating visual features for the corresponding subregion. The indexing structure consists of a set of separate multidimensional indices for the clusters and each cluster is connected to a single root node as shown in Figure 14. The indexing structure includes: a root hash table for keeping track of the information about all the clusters in the database, a leaf hash table for each cluster for recording the information about all its subclusters, a second-leaf hash table for each subcluster for recording the information about all its subregions, and a hash table for each subregion for mapping all its data points to the disk pages where the videos reside. The root hash table keeps the information about all the semantic clusters and each root node may consist of a set of leaf nodes for accessing its subclusters. Moreover, the representative features associated with each root node are centroid, radius, meta features, visual features, semantic features, dimensional weighting coefficients, number of leaf nodes, and representative icons. Each leaf node is also represented by a set of parameters as described in Section 3. The indexing structure also consists of

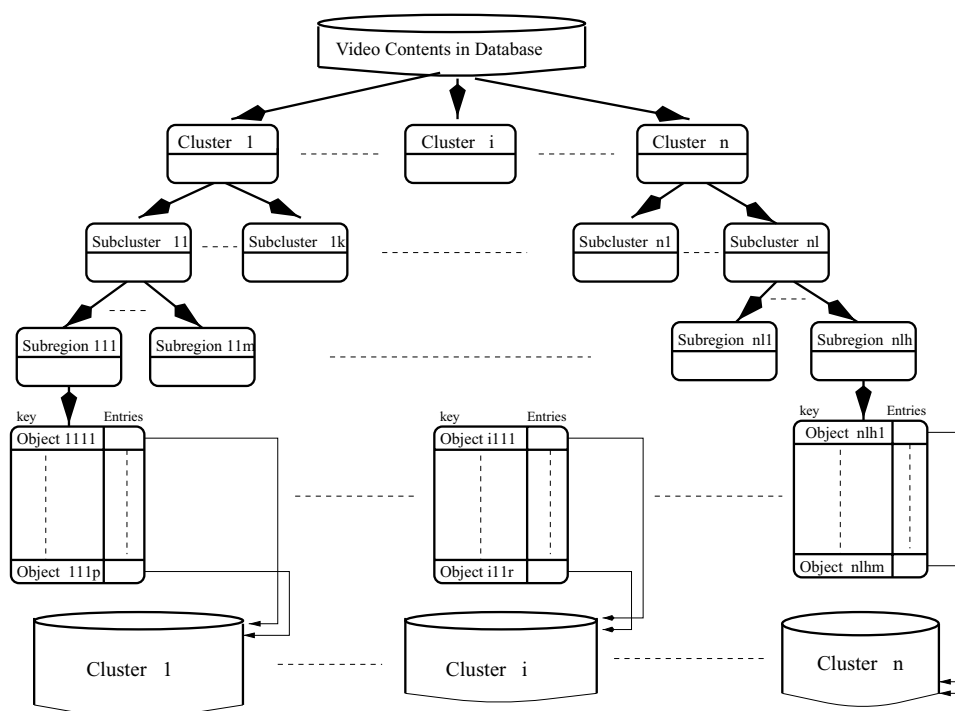


Fig. 14. The hierarchical video database partition and cluster-based indexing structure. The pages in a hash table are kept at about 80 percent occupancy and leave some space for future insertions.

hash tables for the clusters, subclusters, and subregions, where the keys are their discriminating visual features used for characterizing their centroids and radiuses and the entries are the pointers to their lower-level components in the hierarchy. The hash table for each subregion is built by mapping all its videos to the associated disk pages; an indexing pointer is assigned to each video as shown in Figure 14. 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 each subregion are also maintained, where the keys are their discriminating visual 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 into a set of independent disks as shown in Figure 14. Since the sizes (number of data points) of clusters may be very different, the paths from the root nodes to their leaf-nodes may not have exactly the same length. To answer a query, only the semantic clusters that are near the query object are retrieved. Traditional high-dimensional indexing trees, such as R-tree, X-tree, SS-tree, SR-tree, can also be used for indexing these multidimensional data points in the same subregion. However, it has been shown that if  $D_i \gg \log N$ , then no nearest neighbor algorithm can be significantly faster than a linear search. Therefore, a multidimensional scanning technique is used for obtaining the pointers for the data points in the same subregion [Fan et al. 2001a].

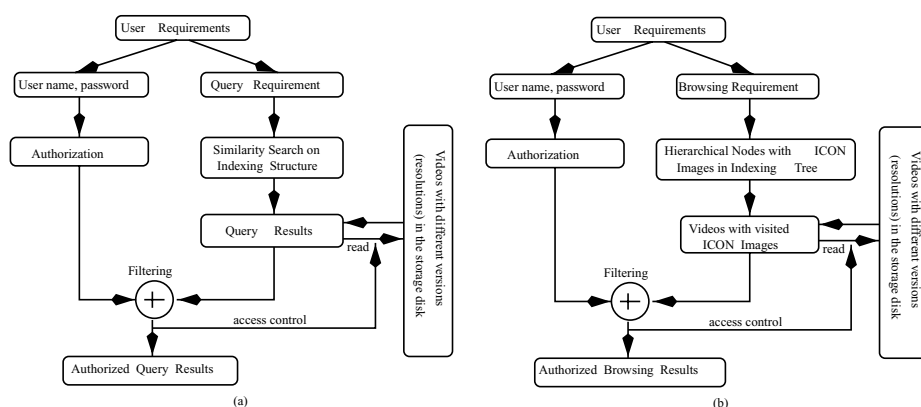


Fig. 15. The video access control architecture: (a) access control for an authorized query-by-example procedure; (b) access control for an authorized browsing procedure. The ICON images are the representative frames that are selected for visualizing the content summary.

## 7. ACCESS CONTROL

Integrating access control into video database systems is achieved by specifying a set of filtering rules and access control procedures. Filtering rules describe *who* is allowed to access *what* in the video database and under *which* mode. Filtering rules are specified according to an access control model that takes into account the peculiar protection requirements of video data. Access control procedures deploy these rules on database transactions. Users can require access to video elements according to two different modes: *querying* and *browsing*. Under the querying mode, users require specific video shots (e.g., through a query-by-example approach). By contrast, under the browsing mode, users browse and navigate the video database through its semantic categories. For both browsing and querying, the access procedures can be organized according to two phases as shown in Figures 15(a) and 15(b), respectively. During the first phase, the relevant video elements are selected. Then a filtering phase is performed against the retrieved objects. In this phase, authorization information is used for filtering the results to be returned to the users according to their authorizations. In the following, we first illustrate the main characteristics of the proposed access control model. We then present the access control procedure for both querying and browsing mode.

### 7.1 The Access Control Model

The key characteristic of the access control model we propose is the flexibility it provides in specifying the video elements to which filtering rules apply. The model supports a wide range of protection granularity levels, in that it is possible to specify filtering rules that apply to a whole semantic cluster, a sub-cluster with special contents, a video stream, a video segment (e.g., video shot or video object), a video frame, or even a salient object. Moreover, the model supports both content-dependent and content-independent access control to video elements. Indeed, a filtering rule can contain both a set of video element

identifiers or a *content expression*, implicitly denoting a set of video elements. Content expressions are built by exploiting the set of features which are associated with video elements. Video elements denoted by a content expression are those whose features satisfy the expression. An additional feature of the proposed access control model is the possibility of easily expressing *exceptions* to the access authorizations granted through filtering rules. This is a relevant characteristic, since different users may have different permissions to obtain different details or layers of the same video element. For example, a user may be permitted to access the *sport* cluster, but he/she may not be permitted to access the lower layer of the same cluster (e.g., subcluster) such as *soccer*, or a user may be permitted to access a video stream, but he/she may not have permission to access the lower layer of the same video stream such as some video shots, some frames, or even some parts of frames. The hierarchical video database model, representation, and indexing structures we propose in this paper are very suitable for addressing such multilevel accessing control problems. Additionally, the proposed access control model allows one to grant an access privilege only for selected time instants and with specific quality levels. In the following, we first present the filtering rules that can be specified in our model. Then we discuss how to deal with possible conflicts among filtering rules.

**7.1.1 Filtering Rules.** To formalize the main components of the access control model, we denote with  $\mathcal{VD}$  the target video database, that is, the database to be protected, and with  $\mathcal{U}$  the set of users authorized to access  $\mathcal{VD}$  services. Elements in  $\mathcal{VD}$  are hierarchically organized according to the video model previously introduced, and therefore can be organized into a *hierarchy*, which is a partial order  $<_{\mathcal{VD}}$ .<sup>1</sup> Given two video elements  $ve_1, ve_2 \in \mathcal{VD}$ ,  $ve_1 <_{\mathcal{VD}} ve_2$  if and only if  $ve_1$  is a component of  $ve_2$ . For instance, given a cluster  $c$  and one of its subcluster  $sc$ ,  $sc <_{\mathcal{VD}} c$  holds. As far as privileges are concerned, we can classify them into two main categories, which correspond to play and edit operations. However, in our current work, we consider only play operations and we assume that only the database provider is permitted to edit (i.e., insert, delete, and update) video elements in the video database. The play privilege can be granted for different time periods and with different display qualities. In the following, we denote with  $\mathcal{N}$  the set of natural numbers, and with  $\mathcal{Q}$  a partially ordered set of quality levels. We assume that  $\mathcal{Q}$  always contains element  $\top$  such that, for all  $x \in \mathcal{Q}$ ,  $x \preceq \top$ .

*Definition 7.1 (Mode Specification).* A mode specification is a pair  $(d, q)$ , where  $d \in \mathcal{N}$  denotes a temporal duration, and  $q \in \mathcal{Q}$  is a quality level. Both the duration and the quality level may be omitted. In that case, by default,  $q = \top$ , and  $d$  is set equal to the duration of the video element with which the mode specification is associated.

For instance, the mode specification  $(60, \text{low})$  states a play permission with low quality for 60 seconds.<sup>2</sup> Video elements to which a filtering rule applies

<sup>1</sup>By *hierarchy* we mean a poset  $(S, <)$ , where  $S$  is a set and  $<$  is a partial order over  $S$ .

<sup>2</sup>We assume that video duration is expressed in seconds.

can be either explicitly specified, by listing their identifiers, or can be implicitly denoted by imposing conditions on their associated features. Such conditions can be specified through content expressions, formally defined as follows. In the following, we denote with  $\mathcal{FT}$  a set of names of the features associated with elements in  $\mathcal{VD}$ , with  $\mathcal{V}$  a set of values for features in  $\mathcal{FT}$ , and with  $\mathcal{OP}$  a set of operators for values in  $\mathcal{V}$  (e.g.,  $<$ ,  $>$ ,  $=$ , etc.).

*Definition 7.2 (Content Expression).* The set  $\mathcal{CE}$  of content expressions is defined as follows:

- (1) each element in  $\mathcal{FT}$  is a content expression;
- (2) if  $ft \in \mathcal{FT}$ ,  $v \in \mathcal{V}$ , and  $op \in \mathcal{OP}$ , then  $ft \text{ OP } v$  is a content expression<sup>3</sup>;
- (3) if  $ce_1$  and  $ce_2$  are content expressions, then  $ce_1 \wedge ce_2$ , and  $ce_1 \vee ce_2$  are content expressions.

*Example 7.1.* The following are examples of content expressions:

- $\text{Soccer} \wedge \text{Date} > 1/1/2002$ : it denotes all the video elements dealing with soccer and which have been produced after January, 1, 2002.
- $\{c_1, \dots, c_4, f_1\}$ : it denotes clusters  $c_1, \dots, c_4$  and frame  $f_1$ .

In the following, given a content expression  $ce \in \mathcal{CE}$ , we denote with  $Eval(ce)$  the set of identifiers of elements in  $\mathcal{VD}$  that satisfy  $ce$ . A video element specification denotes the set of video elements to which a filtering rule applies. We provide a wide range of protection granularity levels, in that filtering rules may apply either to specific video elements (denoted by their identifiers) or to all clusters, subclusters, subregions, and so on. Alternatively, the video elements to which a filtering rule applies can be implicitly specified through a content expression: those video elements are the ones which satisfy the content expression. All these possibilities are summarized by the following definition.

*Definition 7.3 (Video Element Specification).* A video element specification can assume one of the following forms:

- (1) a set  $\{id_1, \dots, id_n\}$  of video elements identifiers in  $\mathcal{VD}$ , or
- (2) the keywords `all_clusters`, `all_subclusters`, `all_regions`, `all_subregions`, `all_scenes`, `all_shots`, `all_frames`, `all_salient_objects`, denoting all the cluster, subcluster, region, subregion, scene, shot, frame, salient object identifiers in  $\mathcal{VD}$ , respectively, or
- (3) a content expression in  $\mathcal{CE}$ , implicitly denoting a set of video element identifiers in  $\mathcal{VD}$ .

One of the key characteristics of the proposed access control model is the possibility of specifying exceptions to the access authorizations implied by a filtering rule. To this purpose, a filtering rule contains two distinct video element specifications: one denoting the set of video elements to which the rule applies, and

<sup>3</sup>Note that the operators that can be used in a content expression  $ft \text{ OP } v$  are actually a subset of the ones in  $\mathcal{OP}$ , since they depend on the type of the feature  $ft$ . We assume that there is some mechanism in place that verifies that only meaningful operators are used in a content expression.

one denoting a subset (or portions) of the video elements specified by the first specification which denotes selected components that the user is not allowed to access. We refer to the video elements denoted by this second specification as *censored elements*. Censored elements are used to specify exceptions to the accesses granted by a filtering rule. We are now ready to introduce the definition of filtering rule.

*Definition 7.4 (Filtering Rule).* A filtering rule is a tuple:  $(\text{user}, \text{target\_elements}, \text{censored\_elements}, \text{mode\_spec})$ , where  $\text{user} \in \mathcal{U}$  is the user to which the filtering rule applies,  $\text{target\_elements}$  and  $\text{censored\_elements}$  are video element specifications, specified according to Definition 7.3, whereas  $\text{mode\_spec}$  is a mode specification. The  $\text{censored\_elements}$  and  $\text{mode\_spec}$  components may be omitted. If  $\text{censored\_elements}$  is omitted, then, by default,  $\text{censored\_elements} = \emptyset$ , whereas if  $\text{mode\_spec}$  is omitted we assume that the rule authorizes the denoted video elements to be displayed for their whole duration at the highest-quality level.

Thus, the filtering rule  $(u, te, ce, (d, ql))$  authorizes user  $u$  to play for  $d$  seconds and with quality level  $ql$  all the video elements denoted by  $te$  which are not contained in the set of video elements denoted by  $ce$ .

*Example 7.2.* The filtering rule  $F_1 = \langle \text{Bob}, \text{Soccer} \rangle$  authorizes Bob to access all the video elements dealing with soccer. In this case, since the mode specification is omitted, the authorization is for playing at the highest-quality level and for the whole duration of the denoted video elements. By contrast, the filtering rule  $F_2 = \langle \text{Ann}, \text{all\_clusters}, \text{Drugs}, (300, \text{low}) \rangle$  authorizes Ann to play for five minutes with a low-quality level all the clusters contained in the video database apart from those reporting information about drugs. Finally, the filtering rule  $F_3 = \langle \text{Ann}, f_1, so_5 \rangle$  authorizes Ann to play the frame whose identifier is  $f_1$ , except for the salient object  $so_5$ .

In the following, we denote with the term *Filtering Base (FB)* the set of filtering rules specified for video elements in the target video database. Moreover, given a filtering rule  $F = \langle \text{user}, \text{target\_elements}, \text{censored\_elements}, \text{mode\_spec} \rangle$ , we denote with  $\text{user}(F)$ ,  $\text{target\_elements}(F)$ ,  $\text{censored\_elements}(F)$ , and  $\text{mode\_spec}(F)$  the user, the target elements, the censored elements, and the mode specification of  $F$ , respectively.

**7.1.2 Filtering Rules Conflicts.** The possibility of specifying exceptions in the filtering rules may introduce potential conflicts. For instance, consider two filtering rules  $F_1$  and  $F_2$ , and suppose that  $F_1$  allows a user  $u$  to access a whole cluster  $c$ , except one of its subcluster, say  $sc_1$ . Suppose moreover that the filtering rule  $F_2$  authorizes the same user  $u$  to access subcluster  $sc_1$ , apart from two video shots  $vs_1$  and  $vs_2$ . Clearly, there is a conflict between  $F_1$  and  $F_2$  on subcluster  $sc_1$ , since according to  $F_1$   $u$  must be prevented to access  $sc_1$ , whereas according to  $F_2$ ,  $u$  can access subcluster  $sc_1$  apart from video shots  $vs_1$  and  $vs_2$ . Several approaches can be devised to deal with such conflicting filtering rules. In our approach, we do not consider the simultaneous presence of conflicting filtering rules as an inconsistency; rather we define a conflict resolution policy to

deal with conflicting rules. Our conflict resolution policy is based on the concept of *more specific filtering rule*. The idea is that filtering rules specified on lower levels of our hierarchical video model prevail over filtering rules specified on upper-level elements. This means that in the example above user  $u$  is allowed to access the whole cluster  $c_1$  apart from video shots  $vs_1$  and  $vs_2$  in subcluster  $sc_1$ , since rule  $F_2$  prevails over rule  $F_1$  on subcluster  $sc_1$ , in that it is specified on lower-level video elements with respect to  $F_2$ . To formalize the conflict resolution policy, we need to introduce some preliminary definitions. First, given a video element  $ve$ , we need to identify the filtering rules in  $\mathcal{FB}$  which apply to  $ve$ . To this purpose, we first introduce the notion of projection of a video database with respect to a video element.

*Definition 7.5 (Video Database Projection).* Let  $ve \in \mathcal{VD}$  be a video element. The projection of  $\mathcal{VD}$  with respect to  $ve$ , denoted  $\Pi_{ve}(\mathcal{VD})$ , contains all and only the video elements  $ve' \in \mathcal{VD}$  such that:  $ve' \prec_{\mathcal{VD}} ve$  or  $ve \prec_{\mathcal{VD}} ve'$ .

Thus, the projection of  $\mathcal{VD}$  wrt a video element  $ve$  contains all video elements which precede or follow  $ve$  in the  $\prec_{\mathcal{VD}}$  hierarchy. The above definition is exploited to determine the set of filtering rules which apply to a given video element. This is captured by the notion of *filtering base projection*, formally defined as follows.

*Definition 7.6 (Filtering Base Projection).* Let  $ve \in \mathcal{VD}$  be a video element. The projection of  $\mathcal{FB}$  with respect to  $ve$ , denoted  $\Pi_{ve}(\mathcal{FB})$ , is the subset of  $\mathcal{FB}$  such that:  $\forall F \in \mathcal{FB}, F \in \Pi_{ve}(\mathcal{FB})$  iff  $\text{target\_elements}(F)^4 \cap \Pi_{ve}(\mathcal{VD}) \neq \emptyset$ .

Our conflict resolution policy is formalized by the concept of *more specific filtering rule*, formally introduced as follows.

*Definition 7.7 (More Specific Filtering Rule).* Let  $u \in \mathcal{U}$  be a user, and  $ve \in \mathcal{VD}$  a video element. Let  $F_1, F_2 \in \mathcal{FB}$  be two filtering rules such that  $\text{user}(F_1) = \text{user}(F_2) = u$ , and  $F_1, F_2 \in \Pi_{ve}(\mathcal{FB})$ .  $F_1$  is more specific than  $F_2$  with respect to  $ve$ , written  $F_1 \succ_{ve} F_2$ , iff one of the following conditions holds:

Rule 1.  $\exists ve' \in (\text{target\_elements}(F_1) \cap \Pi_{ve}(\mathcal{VD}))$  such that  $\nexists ve'' \in (\text{target\_elements}(F_2) \cap \Pi_{ve}(\mathcal{VD}))$ ,  $ve'' \prec_{\mathcal{VD}} ve'$ ;

Rule 2. Rule 1 does not hold and  $\exists ve' \in (\text{censored\_elements}(F_1) \cap \Pi_{ve}(\mathcal{VD}))$ , such that  $\nexists ve'' \in (\text{censored\_elements}(F_2) \cap \Pi_{ve}(\mathcal{VD}))$ ,  $ve'' \prec_{\mathcal{VD}} ve'$ .

Thus, when conflicts among two filtering rules arise, the more specific one is considered as prevailing.

*Example 7.3.* Consider the filtering rules  $F_2$  and  $F_3$  of Example 7.2 and suppose that frame  $f_1$  deals with Drugs. Suppose now that user Ann requires a video shot contained in frame  $f_1$ . Suppose moreover that the requested video shot does not contain the salient object  $so_5$ . Clearly, a conflict exists between rules  $F_2$  and  $F_3$  since, according to rule  $F_2$ , Ann must be forbidden to access the requested shot, whereas according to rule  $F_3$ , Ann can access the requested shot. By applying the conflict resolution policy (cf. Rule 1 of Definition 7.7), rule

<sup>4</sup>For simplicity, here and in the following, by abuse of notation,  $\forall F \in \mathcal{FB}$ , we use  $\text{target\_elements}(F)$  as a shorthand for  $\text{Eval}(\text{target\_elements}(F))$ , if  $\text{target\_elements}(F) \in \mathcal{CE}$ .



$F_3$  prevails over  $F_2$ , since it is more specific. Indeed,  $F_3$  is specified on frame  $f_1$ , whereas  $F_2$  is specified at the cluster level. As a result, Ann can access the requested video shot.

## 7.2 Querying Mode

Under the querying mode of access control, a user explicitly submits a query to the video database to require the access to a (set of) video elements. Queries are submitted against video shots, where each video shot is characterized by  $n$ -dimensional representative features. Thus, a video shot  $vs = \{x_1, x_2, \dots, x_n\}$  can be modeled as a point in an  $n$ -dimensional space, where  $x_1, x_2, \dots, x_n$  are the representative features of  $vs$ . The access control procedure consists of three main steps:

- (1) First, the weighted feature-based similarity distances between the requested video shot  $vs$  and the centroids of the semantic clusters are calculated. The query processor then returns the cluster  $A_k$ , which has the smallest similarity distance with respect to  $vs$  or such that the associated similarity distance  $d_F(vs, A_k)$  is not greater than the radius  $\varphi_c^k$  of  $A_k$ . If such a cluster  $A_k$  exists, the query processor finds the associated sub-cluster in  $A_k$  which is most relevant to the requested video shot  $vs$ , and then finds the most relevant subregion by invoking an analogous searching procedure.
- (2) Second, the weighted feature-based similarity distances between the requested video shot  $vs$  and all the objects in the selected subregion are calculated. The search engine then returns a set of ranked video elements which are more relevant to  $vs$ .
- (3) Third, the filtering rules are used for selecting the suitable videos with suitable versions from these obtained query results.

These steps are formalized by the algorithm in Figure 17, whose strategy is graphically illustrated in Figure 16. The algorithm receives as input a pair  $(u, vs)$ , where  $u$  is the user submitting the access request and  $vs$  is the requested video shot, and returns all video elements that are relevant to the user query and that  $u$  is allowed to access according to the filtering rules in the  $\mathcal{FB}$ . For each returned video element, the algorithm also returns the allowed display time and quality level. Such information are maintained in set *Authorized.elements*. The algorithm makes use of function *Solve\_Query* that receives as input a target video shot  $vs$  and returns a set of ranked video elements which are more relevant to  $vs$ . The strategy used by function *Solve\_Query()* is the one reported in steps 1 and 2 above. If *Solve\_Query()* does not return any video shot then the access is denied. Otherwise, the filtering step is applied to each video element  $ve$  returned by *Solve\_Query()*. First (step 3), a query is executed on the  $\mathcal{FB}$  to extract all the filtering rules specified for user  $u$  that apply to  $ve$ . Such rules are collected into set *Target\_Rules*. Then, the algorithm computes a restriction of  $\Pi_{ve}(\mathcal{VD})$  which contains only  $ve$  and its direct and indirect children, and extracts the most specific rules from *Target\_Rules*. For each of these rules, it identifies the shot/frames/salient objects in the restriction of the projection to

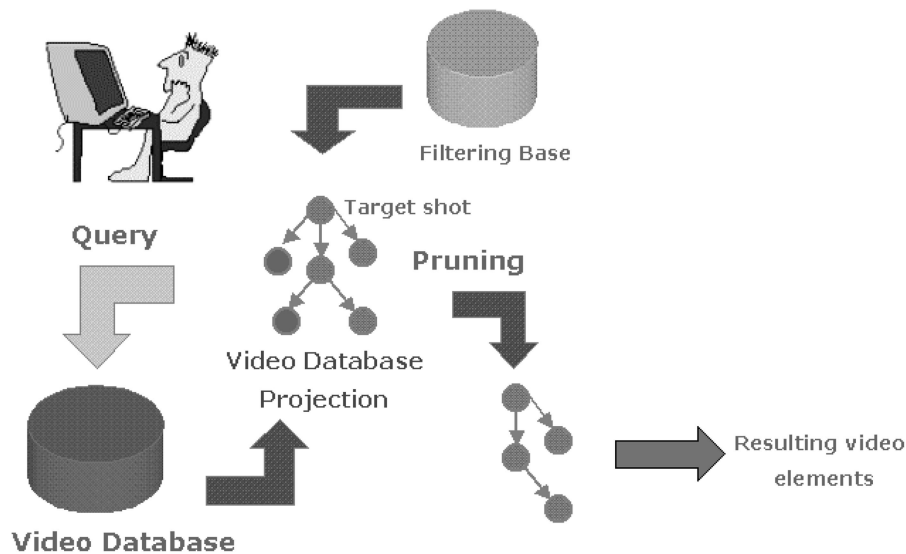


Fig. 16. Access Control Strategy.

which the rule applies. All the unmarked shot/frames/salient objects denoted by the *censored\_elements* component of the rule, and all the children of such elements, are marked as not accessible by the requesting user, whereas all the unmarked shot/frames/salient objects denoted by the *target\_elements* component of the rule, and all the children of such elements, are marked as accessible by the requesting user. Additionally, the video elements marked as accessible are marked with the display quality level and the temporal duration, if any, specified in the filtering rule. Then, the considered rules are removed from set *Target\_Rules* and the process is iterated till *Target\_Rules* is empty or all the video elements in the projection have been marked. The resulting set of video elements is then computed by pruning from the projection all video elements which are marked as nonaccessible. This task is done by function *Prune()*. The result of function *Prune()* is added to set *Authorized\_elements* and the same process is then iterated for all the video elements returned by *Solve\_Query()*. If, at the end of this process, *Authorized\_elements* is empty, then the access is denied; otherwise the algorithm halts by returning set *Authorized\_elements*. Figure 18(a) shows the ranked results for a *free* feature-based query procedure in which the filtering procedure is not activated. The icon video contents for these query results are visualized. Users can thus first get a rough sense of the video contents and select more relevant query results for browsing. For a video database complemented with a filtering base, the filtering interface is activated when users push the button to get the details of these query results as shown in Figure 18(b). The relevant query results with suitable versions will be delivered to the user according to the result of Algorithm 7.1. This often leads to the generation of multiple copies of the same video element stored in the disk and censored at different levels for different classes of users.

ALGORITHM 7.1. **Querying Mode Access Control Algorithm**


---

**INPUT:** 1) An access request  $(u, vs)$ , where  $u$  is a user and  $vs$  is a video shot  
 2) The Filtering Base  $FB$

**OUTPUT:** 1) *Authorized.elements*, i.e., the set of  $vs$  components  $u$  is authorized to access,  
 if *Authorized.elements*  $\neq \emptyset$ ,  
 2) ACCESS DENIED, otherwise

**METHOD:**

- (1) *Authorized.elements* is initialized to be empty
- (2) *Interesting.elements* = *Solve\_Query*( $vs$ )
- (3) **If** *Interesting.elements* =  $\emptyset$ : **return** ACCESS DENIED
  - else**
    - For** each  $ie \in$  *Interesting.elements*:
      - Let *Target.Rules* =  $\{F \mid F \in \Pi_{ie}(FB), \text{ and } \text{user}(F) = u\}$
      - Let  $\Pi_{ie}(\mathcal{VD}) \parallel_{ie} = \{ie\} \cup \{ve \mid ve \in \Pi_{ie}(\mathcal{VD}) \text{ and } (ve \prec_{\mathcal{VD}} ie \text{ or } ve = ie)\}$
      - While** *Target.Rules*  $\neq \emptyset$  and  $\exists ve \in \Pi_{ie}(\mathcal{VD}) \parallel_{ie}$  which is not marked:
        - Let *Strongest.Target.Rules* =  $\{F \mid F \in \text{Target.Rules and } \nexists F' \in \text{Target.Rules with } F' \succ_{ie} F\}$
        - For** each  $R \in$  *Strongest.Target.Rules*:
          - Let  $UM$  be the set of unmarked video elements in  $\Pi_{ie}(\mathcal{VD}) \parallel_{ie}$
          - For** each  $ve \in UM$  which is denoted by *censored.elements*( $R$ ):
            - Mark  $ve$  and each  $ve' \in UM$  such that  $ve' \prec_{\mathcal{VD}} ve$  with NA
            - Remove the marked elements from  $UM$
          - endfor**
          - For** each  $ve \in UM$  which is denoted by *target.elements*( $R$ ):
            - Mark  $ve$  and each  $ve' \in UM$  such that  $ve' \prec_{\mathcal{VD}} ve$  with  $(A, \text{mode\_spec}(R))$
            - Remove the marked elements from  $UM$
          - endfor**
          - Remove *Strongest.Target.Rules* from *Target.Rules*
        - endwhile**
        - Add *Prune*( $\Pi_{ie}(\mathcal{VD}) \parallel_{ie}$ ) to *Authorized.elements*
      - endfor**
      - If** *Authorized.elements*  $\neq \emptyset$ : **return** *Authorized.elements*
      - else:** **return** ACCESS DENIED

**endif**

---

Fig. 17. An access control algorithm for querying mode.

### 7.3 Browsing Mode

Users, however, are not only interested in searching for specific video shots or key objects (e.g., query-by-example). They would also like to browse and navigate the video database through its semantic categories. Such requirements have created great demands for effective and flexible systems to manage digital videos. Browsing refers to a technique or a process where users skip information rapidly and decide whether the content is relevant to their needs. Browsing video databases should be like scanning the table of contents and indices of a book, or flipping through the pages, to quickly get a rough sense of content and gradually focus on particular chapters or sections of interest. We believe that our semantic clustering technique and cluster-based hierarchical indexing structure are very suitable for providing fast browsing. Moreover, a semantic manual text title and a set of icon images are visualized for each cluster, and different semantic titles or icon images are then categorized in the form of a table



(a)



(b)

Fig. 18. An example of content-based video retrieval: (a) an example of video query result; (b) filtering interface.



Fig. 19. The cluster-based browsing system: (a) browsing of whole database; (b) authorization for browsing the selected semantic cluster; (c) authorized browsing of the selected semantic cluster; (d) authorization for browsing an interesting video sequence in the selected cluster.

of video contents for providing an overview of video contents in the database. This categorization of video contents into semantic clusters can be seen, among other things, as a solution to bridge the gap between low-level visual features and high-level semantic concepts, and it can be helpful both in organizing video database contents and in obtaining the automatic annotation of video contents. After the semantic categorization of video contents is obtained, 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) as shown in Figure 19(a). One can find that users can get a rough sense of the video contents in a cluster without moving down to a lower level of the hierarchy. Filtering rules can be used for controlling this database browsing procedure, using the same methods employed in Algorithm 7.1 (see Figure 19(b)). Browsing the selected semantic cluster is supported by partitioning the video contents in the same cluster into a set of

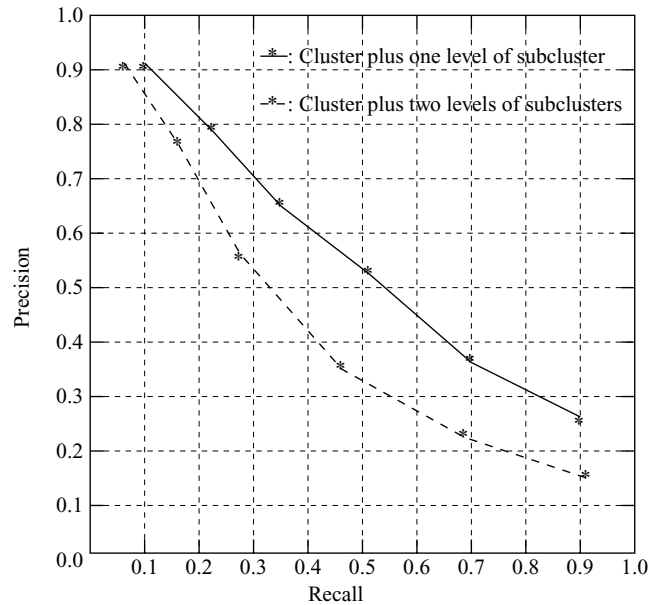


Fig. 20. The average performance of hierarchical video database indexing technique. Each point is obtained via 810 queries from three different video sources.

subclusters. The icon video content for each subcluster is also visualized as shown in Figure 19(c). 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. Moreover, the user can expand or contract the tree and preview the listed videos. The filtering procedure for the same semantic cluster is shown in Figure 19(d).

#### 7.4 Performance Analysis

The average performance of our hierarchical video database indexing technique is given in Figure 20. The efficiency of our video database access control strategies is measured by (a) the average query cost from user point of view; (b) the access control accuracy ratio. The query cost of our hierarchical video database access control scheme is given in Figure 22. The query cost includes the time for hierarchical authorization and query processing over the proposed database indexing structure. Another relevant performance measure when dealing with video access control, is the *access control accuracy*. We define the access control accuracy ratio as

$$\text{control\_ratio} = \frac{\text{authorized\_access}}{\text{total\_access}}.$$

The access control accuracy ratio defines which is the percentage of database accesses which are authorized according to the specified filtering rules. The relevance of this measure is due to the fact that different semantic interpretations may be associated with the same video frame. Consider as an example



Fig. 21. Different interpretations for the same image.

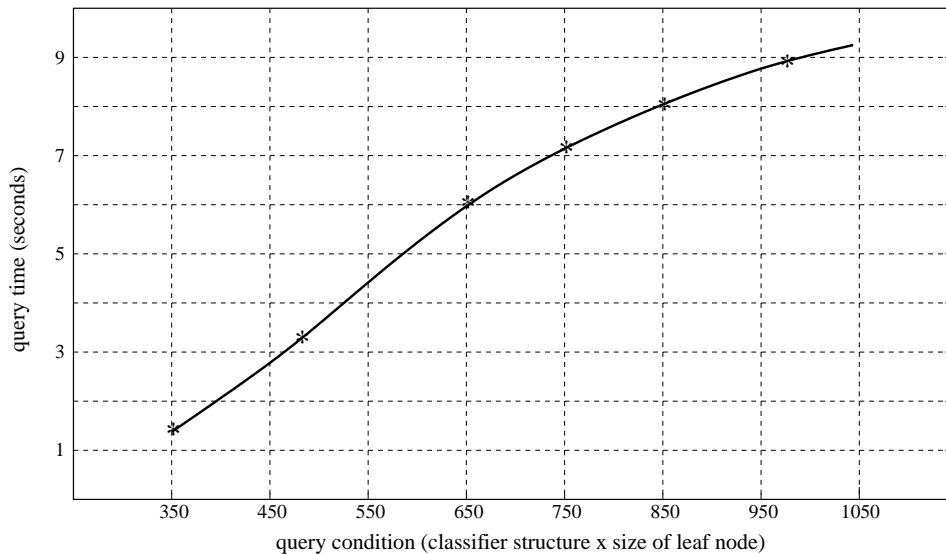


Fig. 22. The average query cost of our hierarchical video database access control scheme.

the image in Figure 21. Some people may see people in touch, some people may see a cup. If some do not have permission to access cup images, but they have permission to access “two people in touch” images, they can eventually get this image, which results in a failure of the access control mechanism, due to the semantic gap problem. In our current implementation, we observe that more than 95% database accesses are successfully authorized. The average testing results are given in Table V. One can find that the access control accuracy ratio depends on two parameters: (a) video data types; (b) database management levels (i.e., visual concept levels). If the video data type includes a large population of various visual concepts and complex relationships among the visual concepts, such as *movies*, the performance of our hierarchical video database access control scheme will decrease. Our hierarchical video database access

Table V. The Average Performance of Our Hierarchical Video Database Access Control Scheme

Test data types	Accuracy on cluster level	Accuracy on shot level	Accuracy on object level
Medical videos	97.3%	91.2%	89.7%
News videos	98.02%	91.6%	90.23%
Movie videos	90.2%	88.9%	87.8%

control can provide more effective protection for the higher-level semantic visual concepts because it is easy to define the access control tools for these higher-level database management units and access control can be easily implemented by hiding the related database indexing units. This is also a major advantage of our video database access control scheme as compared with existing ones [Bertino et al. 2000; Kumar and Babu 1998].

### 7.5 Access Control Strategies

In our system, checking whether a request to access a given video element can be authorized or should be denied may require the evaluation of several video element specifications, which may be expensive at run-time. To further reduce the cost of access control, we are investigating the use of precomputation strategies for enforcing access control. The idea is that whenever a filtering rule is inserted into the filtering base, we calculate and store the set of video element identifiers to which it applies. This set is computed by identifying the set of video elements denoted by the *target\_elements* component of the rule and by removing from this set the identifiers of video elements denoted by the *censored\_elements* component of the rule. By using this strategy, access control becomes very efficient, since there is no difference in costs between filtering rules in which video elements are explicitly specified and filtering rules in which they are implicitly specified through content expressions. Note that the drawback of a precomputation strategy is that administrative operations, such as the creation, deletion, or modification of a video element, become more expensive. However, these operations are considerably less frequent than access requests and, additionally, efficient strategies can be devised to limit as much as possible the overhead introduced by the precomputation on the execution of administrative operations. In the following, we briefly discuss a possible strategy for managing administrative operations when a precomputation strategy is applied. When a new video element is acquired by the system, it is first processed by a module which extracts all its associated features. At this point, all the rules in the filtering base which contain a feature relevant for the new object must be considered to check whether they apply to the new video element. By contrast, the deletion of a video element only requires the removal of the deleted element from all the sets of video elements associated by the precomputation step to the filtering rules in the filtering base. The modification of a video element can be handled using the same strategy employed for video element acquisition.

## 8. APPLICATION TO MPEG-7

To define exchangeable formats, MPEG has initiated a new work item, formally called “Multimedia Content Description Interface,” better known as MPEG-7.



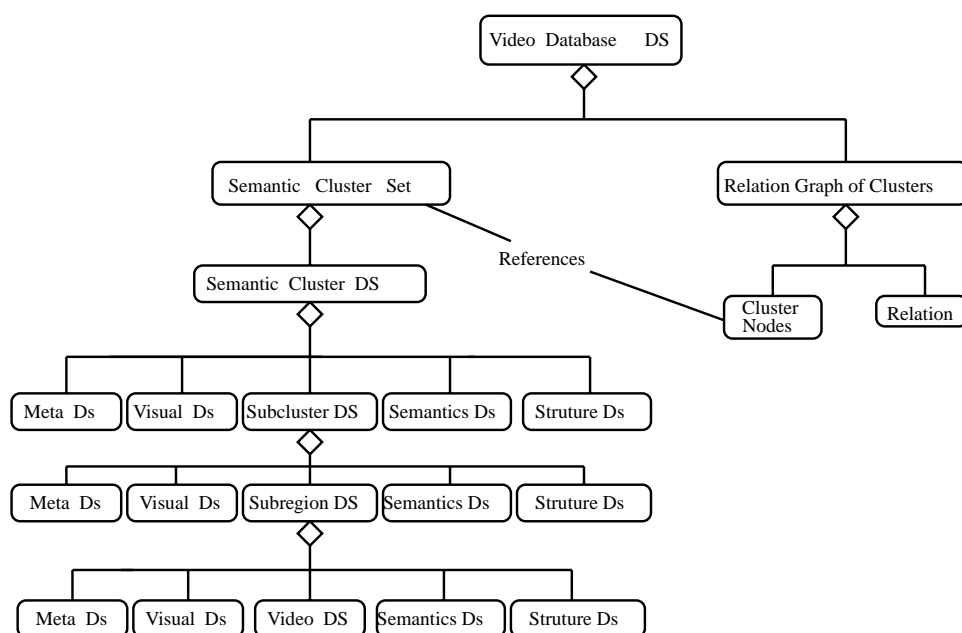


Fig. 23. The video database description scheme, where the diamond represents the aggregation of sub-DSs.

MPEG-7 aims at developing a multimedia content description standard in order to facilitate multimedia searching and filtering application [Li et al. 2000; Beritez et al. 2000]. In the context of MPEG-7, a description of an audiovisual (AV) document includes descriptors (termed *Ds*), which specify the syntax and semantics of a representation entity for a feature of the AV data, and description schemes (termed *DSs*) which specify the structures and semantics of a set of *Ds* and *DSs*. The MPEG-7 *DS* consists of two parts: one is a set of *Ds*, another is the contextual and logical relationships among these *Ds* that indicate how these *Ds* can be integrated to generate a *DS*. Descriptions are expressed in a common description definition language (DDL) to allow their exchange and access.

In order to support video content description in MPEG-7, many approaches have been proposed in the past [Li et al. 2000; Beritez et al. 2000]. However, all these approaches focus on describing a single video sequence. With the explosive growth of video data, description schemes (*DSs*) for video database become an urgent need [Salembier et al. 2000]. In Salembier [2000], the concept of cluster-based approach is introduced, but no details are given. In Section 3, we have proposed a semantic-sensitive video database model which can express the contextual and logical relationships among different database levels (i.e., different concept levels). Each node of our video database model is also characterized by visual features and semantic label, and many techniques have been proposed to describe these visual features. Based on this observation, a hierarchical video database description scheme can be supported in our multilevel video database system by using our hierarchical video database model. Our video database

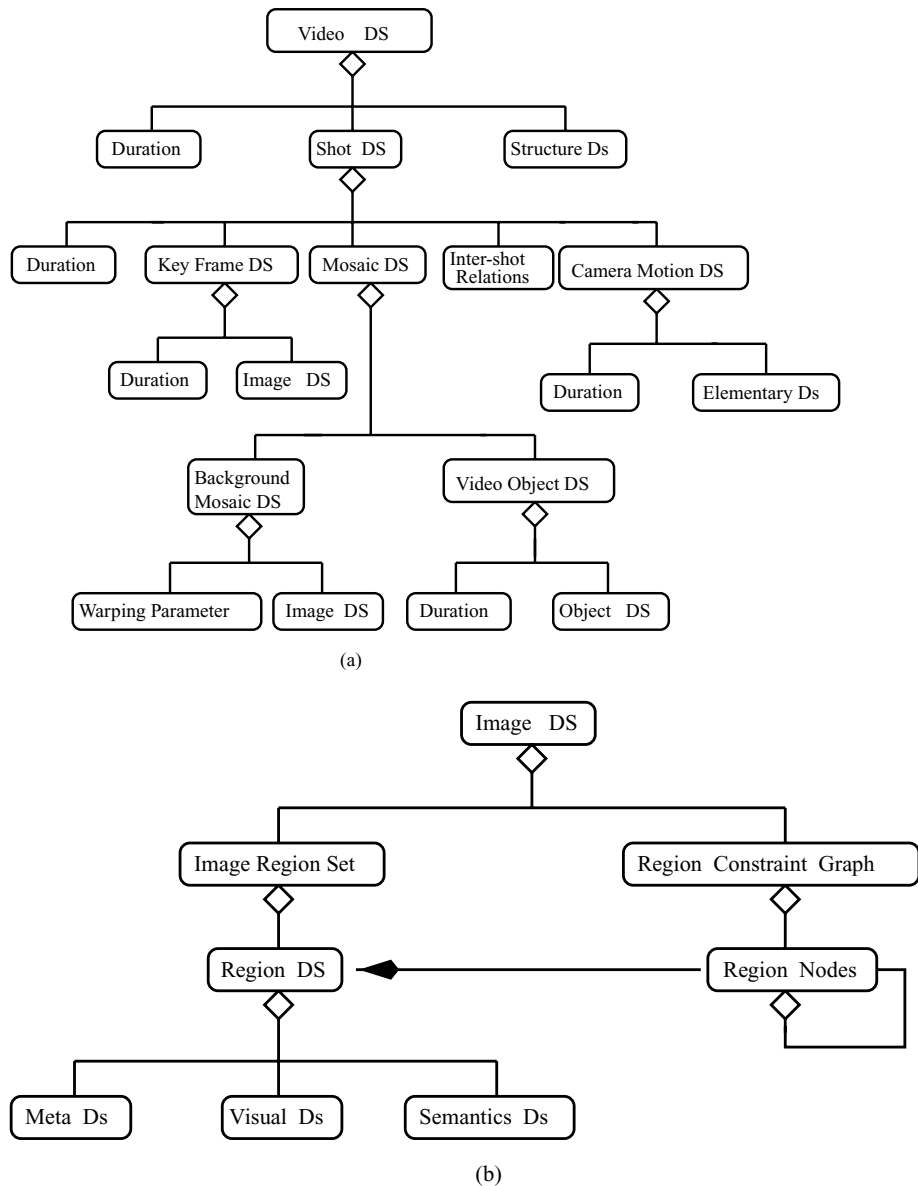


Fig. 24. (a) The shot-based video content description scheme; (b) the image (video frame) description scheme.

description schemes (DSs) at each level, such as the semantic cluster level, include two parts: one is a set of description schemes or descriptors for its sublevel database units, another is the contextual and logical relationships among these sublevel components (i.e., subclusters) that indicate how they are integrated for generating the high-level description scheme (i.e., DS for the corresponding semantic cluster). By this approach, we can get the description schemes for

video database because the contextual and logical relationships among different database levels are represented in our semantic video database model.

The hierarchical video database DS, shown in Figure 23, describes the physical organization (structure) of the database and its semantic properties. The semantic relationship represented in the hierarchy in Figure 23, is of the type “is-made-of” to address the high-level description of database. One can find that the higher-level DS is an aggregation of a set of lower-level DSs. Our system supports the shot-based approach for accessing video contents in database. The shot-based video DS as shown in Figure 24(a) is to define the shot-based temporal and spatial organization structures of a video and to describe its visual properties. The image DS as shown in Figure 24(b) is used to define the spatial organization structure of an image and to represent the relationships among the regions.

## 9. CONCLUSIONS

In this paper, we have proposed a novel approach to support multilevel access control in video databases. Our technique combines video indexing mechanisms with a hierarchical organization of video contents, so that different classes of users can access different video elements or even the same video element with different qualities on the basis of their permissions. We have also proposed a hierarchical video database model to enable multilevel access control in video databases. We have also proposed a cluster-based indexing technique to support this multilevel video access control mechanism. The access control model developed in this paper is specifically tailored to the characteristics of video data, and provides a very flexible mechanism for specifying the video elements to which a filtering rule applies.

Future work includes a comprehensive testing of the proposed models and the implementation of the materialization approach for performing access control.

## REFERENCES

- ADAM, N., ATLURI, V., BERTINO, E., AND FERRARI, E. 2002. A content-based authorization model for digital libraries. *IEEE Trans. Knowl. Data Eng.* 14, 2, 296–315.
- ASLANDOGAN, Y. A., THIER, C., YU, C. T., ZOU, J., AND RISHE, N. 1997. Using semantic contents and WorldNet™ in image retrieval. In *Proceedings of the International Conference on Research and Development in Information Retrieval (SIGIR'97)*. 286–295.
- BARAANI-DASTJERDI, A., PIEPRZYK, J., AND SAFAVI-NAINI, R. 1997. A multi-level view model for secure object-oriented databases. *Data Knowl. Eng.* 23, 2, 97–117.
- BERCHTOLD, S., KEIM, D. A., AND KRIEGEL, H. P. 1996. The X-tree: an index structure for high-dimensional data. In *Proceedings of the International Conference on Very Large Databases (VLDB'96)*. 28–39.
- BERITEZ, A., PAEK, S., CHANG, S.-F., PURI, A., HUANG, Q., SMITH, J. R., LI, C.-S., BERGMAN, L. D., AND JUDICE, C. N. 2000. Object-based multimedia content description schemes and applications for MPEG-7. *Signal Processing: Image Commun.* 16, 235–269.
- BERTINO, E., BETTINI, C., FERRARI, E., AND SAMARATI, P. 1998. An access control model supporting periodicity constraints and temporal reasoning. *ACM Trans. Database Syst.* 23, 3, 231–285.
- BERTINO, E., HAMMAD, M., AREF, W., AND ELMAGARMID, A. K. 2000. An access control model for video database systems. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM'00)*.
- BERTINO, E., JAJODIA, S., AND SAMARATI, P. 1999. A flexible authorization mechanism for data management systems. *ACM Trans. Inform. Syst.* 17, 2, 101–140.

- CHAKRABARTI, K. AND MEHROTRA, S. 1999. The Hybrid tree: an index structure for high dimensional feature spaces. In *Proceedings of the IEEE International Conference on Data Engineering (ICDE'99)*.
- CHANG, S. F., CHEN, W., MENG, H. J., SUNDARAM, H., AND ZHONG, D. 1998. A fully automatic content-based video search engine supporting spatiotemporal queries. *IEEE Trans. Circ. Syst. Video Tech.* 8, 602–615.
- DEL BIMBO, A., VICARIO, E., AND ZINGONI, D. 1995. Symbolic description and visual querying of image sequences using spatio-temporal logic. *IEEE Trans. Knowl. Data Eng.* 7, 4, 609–622.
- DENG, Y. AND MANJUNATH, B. S. 1998. NeTra-V: toward an object-based video representation. *IEEE Trans. Circ. Syst. Video Tech.* 8, 616–627.
- FAN, J., YAU, D. K. Y., AREF, W. G., AND REZGUI, A. 2000. Adaptive motion-compensated video coding scheme towards content-based bit rate allocation. *J. Electron. Imag.* 9, 521–533.
- FAN, J., AREF, W. G., ELMAGAMID, A. K., HACID, M.-S., MARZOUK, M. S., AND ZHU, X. 2001a. Multi-View: multi-level video content representation and retrieval. *J. Electron. Imag.* (Special Issue on Multimedia Database) 10, 4, 895–908.
- FAN, J., YAU, D. K. Y., ELMAGARMID, A. K., AND AREF, W. G. 2001b. Image segmentation by integrating color edge detection and seeded region growing. *IEEE Trans. Image Process.* 10, 10, 1454–1466.
- FAN, J., YU, J., FUJITA, G., ONOYE, T., SHIRAKAWA, I., AND WU, L. 2001c. Spatiotemporal segmentation for compact video representation. *Signal Process.: Image Commun.* 16, 553–566.
- FAN, J., ZHU, X., AND WU, L. 2001d. An automatic model-based semantic object extraction algorithm. *IEEE Trans. Circ. Syst. Video Tech.* 11, 10, 1073–1084.
- FERNANDEZ, E., GUEDES, E., AND SONG, H. 1994. A model for evaluation and administration of security in object-oriented database. *IEEE Trans. Knowl. Data Eng.* 6, 2, 275–292.
- FLICKNER, M., SAWHNEY, H., NIBLACK, W., ASHLEY, J., HUANG, Q., DOM, B., GORKANI, M., HAFNER, J., LEE, D., PETKOVIC, D., STEELE, D., AND YANKER, P. 1995. Query by image and video content: the QBIC system. *IEEE Comput.* 38, 1, 23–31.
- GUTTMAN, A. 1984. R-trees: a dynamic index structure for spatial searching. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD'84)*. 47–57.
- HOLOWCZAK, R. 1997. Extractors for digital libraries objects. Ph.D. dissertation, MS/CIS Department, Rutgers University.
- HUMRAPUR, A., GUPTA, A., HOROWITZ, B., SHU, C. F., FULLER, C., BACH, J., GORKANI, M., AND JAIN, R. 1997. Virage video engine. In *Proceedings of the 5th Conference on Storage and Retrieval for Image and Video Databases* (San Jose, CA, Feb.). 188–197.
- HUANG, J., KUMAR, S. R., AND ZABIH, R. 1998. An automatic hierarchical image classification scheme. In *Proceedings of the ACM Multimedia Conference* (Bristol, U.K.).
- ISHIKAWA, Y., SUBRAMANYA, R., AND FALOUTSOS, C. 1998. MindReader: querying databases through multiple examples. In *Proceedings of the International Conference on Very Large Databases (VLDB'98)*. 218–227.
- JAIN, A. K., VAILAYA, A., AND WEI, X. 1999. Query by video clip. *ACM Multimedia Syst.* 7, 369–384.
- KOBLA, V. AND DOERMANN, D. 1998. Indexing and retrieval of MPEG compressed video. *J. Electron. Imag.* 7, 294–307.
- KUMAR, P. S. AND BABU, G. P. 1998. Intelligent multimedia data: data+indices+inference. *ACM Multimedia Syst.* 6, 395–407.
- LI, J., WANG, J. Z., AND WIEDERHOLD, G. 2000. SIMPLiCity: semantic-sensitive integrated matching for picture libraries. In *Proceedings of the VISUAL Conference* (Lyon, France).
- LIN, K., JAGADISH, H. V., AND FALOUTSOS, C. 1995. The TV-tree: an index structure for high-dimensional data. *VLDB J.* 3, 517–542.
- LIU, H. AND ZICK, G. 1995. Scene decomposition of MPEG compressed video. In *Proceedings of the IS&T/SPIE Symposium on Electronic Imaging: Science and Technology*. Vol. 2119, pp. 26–37.
- MANOLOPOULOS, Y., THEODORIDIS, Y., AND TSOTRAS, V. J. 2000. *Advanced Database Indexing*, Kluwer, Dordrecht, The Netherlands.
- MENG, J. AND CHANG, S.-F. 1996. CVEPS-A compressed video editing and parsing system. In *Proceedings of the ACM Multimedia Conference* (Boston, MA, Nov.)
- MENG, J., JUAN, Y., AND CHANG, S.-F. 1995. Scene change detection in a MPEG compressed video sequence. In *Proceedings of the IS&T/SPIE Symposium on Electronic Imaging: Science and Technology*. Vol. 2419, pp. 14–25.

- MILLER, G. A., BECKWITH, R., FELLBAUM, C., GROSS, D., AND MILLER, K. 1990. Introduction to Word-Net: an on-line lexical database. *Int. J. Lexicography* 3, 235–244.
- MINKA, T. P. AND PICARD, R. W. 1997. Interactive learning with a society of models. *Patt. Recog.* 30, 4, 565–581.
- ORTEGA, M., RUI, Y., CHAKRABARTI, K., PORKAEW, K., MEHROTRA, S., AND HUANG, T. S. 1998. Supporting ranked boolean similarity queries in MARS. *IEEE Trans. Knowl. Data Eng.* 10, 6, 905–925.
- PENTLAND, A., PICARD, R. W., AND SCLAROFF, S. 1996. Photobook: content-based manipulation of image databases. *Int. J. Comput. Vis.* 18, 233–254.
- QUINLAN, J. 1986. Induction of decision trees. *Machine Learn.* 1, 1, 81–106.
- RUI, Y. AND HUANG, T. S. 1999. A novel relevance feedback technique in image retrieval. In *Proceedings of the ACM Multimedia Conference*. 67–70.
- RUI, Y., HUANG, T. S., ORTEGA, M., AND MEHROTRA, S. 1998. Relevance feedback: a power tool for interactive content-based image retrieval. *IEEE Trans. Circ. Syst. Video Tech.* 8, 644–655.
- SALEMBIER, P., QIAN, R., O'CONNOR, N., CORREIA, P., SEZAN, I., VAN BEEK, P. 2000. Description schemes for video programs, users and devices. *Signal Process.: Image Commun.* 16, 211–234.
- SAMARATI, P., BERTINO, E., AND JAJODIA, S. 1996. An authorization model for a distributed hypertext system. *IEEE Trans. Knowl. Data Eng.* 8, 4, 555–562.
- SATOH, S. AND KANADE, T. 1997. Name-It: association of face and name in video. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*.
- SHEIKHOESLAMI, G., CHANG, W., AND ZHANG, A. 1998. Semantic clustering and querying on heterogeneous features for visual data. In *Proceedings of the ACM Multimedia Conference* (Bristol, U.K.).
- SHEN, B. AND SETHI, I. K. 1998. Block-based manipulations on transform-compressed image and videos. *ACM Multimedia Syst.* 6, 113–124.
- TAMURA, H. ET AL. 1978. Texture feature corresponding to visual perception. *IEEE Trans. Syst. Man, Cybern.* 8.
- VAILAYA, A., FIGUEIREDO, M., JAIN, A. K., AND ZHANG, H. J. 1999. A Bayesian framework for semantic classification of outdoor vacation images. In *Proceedings of the IS&T/SPIE Symposium on Electronic Imaging: Science and Technology*. Vol. 3656, pp. 415–426.
- WANG, H. AND CHANG, S.-F. 1999. A highly efficient system for automatic face region detection in MPEG video. *IEEE Trans. Circ. Syst. Video Tech.* 9.
- WHITE, D. A. AND JAIN, R. 1996. Similarity indexing with the SS-tree. In *Proceedings of the IEEE International Conference on Data Engineering* (ICDE'96). 516–523.
- ZHANG, H. J., KANKANHALLI, A., AND SMOLIAR, S. W. 1993. Automatic partitioning of full-motion video. *Multimedia Syst.* 2, 10–28.
- ZHANG, H. J., WU, J., ZHONG, D., AND SMOLIAR, S. 1997. An integrated system for content-based video retrieval and browsing. *Patt. Recog.* 30, 643–658.
- ZHONG, D., ZHANG, H. J., AND CHANG, S.-F. 1996. Clustering methods for video browsing and annotation. In *Proceedings of the IS&T/SPIE Symposium on Electronic Imaging: Science and Technology*. 239–246.
- ZHONG, Y., ZHANG, H. J., AND JAIN, A. K. 2000. Automatic caption localization in compressed video. *IEEE Trans. Patt. Anal. Mach. Intell.* 22, 385–392.
- YEO, B. L. AND LIU, B. 1995. Rapid scene change detection on compressed video. *IEEE Trans. Circ. Syst. Video Tech.* 5, 533–544.
- YEO, B. L. AND YEUNG, M. M. 1997. Classification, simplification and dynamic visualization of scene transition graphs for video browsing. In *Proceedings of the IS&T/SPIE Symposium on Electronic Imaging: Science and Technology*. Vol. 3312, pp. 60–70.

Received August 2001; revised March 2002, August 2002; accepted November 2002