Real-Time Compressed-Domain Spatiotemporal Segmentation and Ontologies for Video Indexing and Retrieval

Vasileios Mezaris, *Student Member, IEEE*, Ioannis Kompatsiaris, *Member, IEEE*, Nikolaos V. Boulgouris, *Member, IEEE*, and Michael G. Strintzis, *Fellow, IEEE*

***Abstract*—In this paper, a novel algorithm is presented for the real-time, compressed-domain, unsupervised segmentation of image sequences and is applied to video indexing and retrieval. The segmentation algorithm uses motion and color information directly extracted from the MPEG-2 compressed stream. An iterative rejection scheme based on the bilinear motion model is used to effect foreground/background segmentation. Following that, meaningful foreground spatiotemporal objects are formed by initially examining the temporal consistency of the output of iterative rejection, clustering the resulting foreground macroblocks to connected regions and finally performing region tracking. Background segmentation to spatiotemporal objects is additionally performed. MPEG-7 compliant low-level descriptors describing the color, shape, position, and motion of the resulting spatiotemporal objects are extracted and are automatically mapped to appropriate intermediate-level descriptors forming a simple vocabulary termed *object ontology*. This, combined with a relevance feedback mechanism, allows the qualitative definition of the high-level concepts the user queries for (*semantic objects*, each represented by a *keyword*) and the retrieval of relevant video segments. Desired spatial and temporal relationships between the objects in multiple-keyword queries can also be expressed, using the *shot ontology*. Experimental results of the application of the segmentation algorithm to known sequences demonstrate the efficiency of the proposed segmentation approach. Sample queries reveal the potential of employing this segmentation algorithm as part of an object-based video indexing and retrieval scheme.**

***Index Terms*—Compressed-domain segmentation, object-based video indexing, ontologies, real-time segmentation, relevance feedback, spatiotemporal video segmentation, support vector machines.**

I. Introduction

THE manipulation of digital video is an integral part of many emerging multimedia applications, particularly in the area of personalized user-interactive services, such as sophisticated query and retrieval from video databases [1]–[3]. Such indexing schemes employ descriptors ranging from low-level features to higher level semantic concepts [4]. Low-level features are machine-oriented and can be automatically extracted (e.g., MPEG-7 compliant descriptors [5]), whereas high-level concepts require manual annotation of the medium or are restricted to specific domains. In all cases, preprocessing of video data is necessary as the basis on which indices are extracted. The preprocessing is of *coarse granularity* if it involves processing of video frames as a whole, whereas it is of *fine granularity* if it involves detection of objects within a video frame [6]. In this paper, a segmentation algorithm is developed to support a novel fine granularity approach to video indexing and retrieval. The low-level features automatically extracted from the resulting spatiotemporal objects are mapped to high-level concepts using an object ontology, combined with a relevance feedback mechanism.

Several approaches have been proposed in the literature for video segmentation [7]. Most of these operate in the uncompressed pixel domain [8]–[10], which provides them with the potential to estimate object boundaries with pixel accuracy but requires that the processed sequence be fully decoded before segmentation can be performed. As a result, the usefulness of such approaches is usually restricted to nonreal-time applications; this is due to the high computational complexity resulting from the large number of pixels that have to be processed. Often the need also arises for motion feature extraction [11]–[13] using block matching algorithms. Real-time pixel-domain methods [14] are usually applicable only on head-and-shoulder sequences (e.g., video-conference applications) or are based on the assumption that the background is uniformly colored, an assumption not always valid in practice.

To counter these drawbacks of pixel-domain approaches, compressed-domain methods have been proposed for spatiotemporal segmentation. However, some of them, although significantly faster than most pixel-domain algorithms, cannot operate in real time [15], [16]. In [17], translational motion vectors are accumulated over a number of frames and the magnitude of the displacement is calculated for each macroblock; macroblocks are subsequently assigned to regions by uniformly quantizing the magnitude of the displacement. In [18] and [19], translational motion vectors and dc coefficients are clustered. In [20], segmentation is performed using ac/dc discrete cosine transform (DCT) coefficients only; foreground/background classification is based on thresholding the average temporal change of each region, while the macroblock motion vectors are

Manuscript received April 28, 2003; revised November 20, 2003. This work was supported by the EU projects SCHEMA "Network of Excellence in Content-Based Semantic Scene Analysis and Information Retrieval" (IST-2001-32795) and aceMedia "Integrating knowledge, semantics and content for user centred intelligent media services" (FP6-001765).

V. Mezaris and M. G. Strintzis are with the Information Processing Laboratory, Electrical and Computer Engineering Department, Aristotle University of Thessaloniki, Thessaloniki 54124, Greece, and with the Informatics and Telematics Institute (ITI)/Centre for Research and Technology Hellas (CERTH), Thessaloniki 57001, Greece (e-mail: strintzi@eng.auth.gr).

I. Kompatsiaris is with the Informatics and Telematics Institute (ITI)/Centre for Research and Technology Hellas (CERTH), Thessaloniki 57001, Greece.

N. V. Boulgouris was with the Informatics and Telematics Institute (ITI)/Centre for Research and Technology Hellas (CERTH), Thessaloniki 57001, Greece. He is now with the Electrical and Computer Engineering department, University of Toronto, Toronto, ON M5S 3G4 Canada.

Digital Object Identifier 10.1109/TCSVT.2004.826768

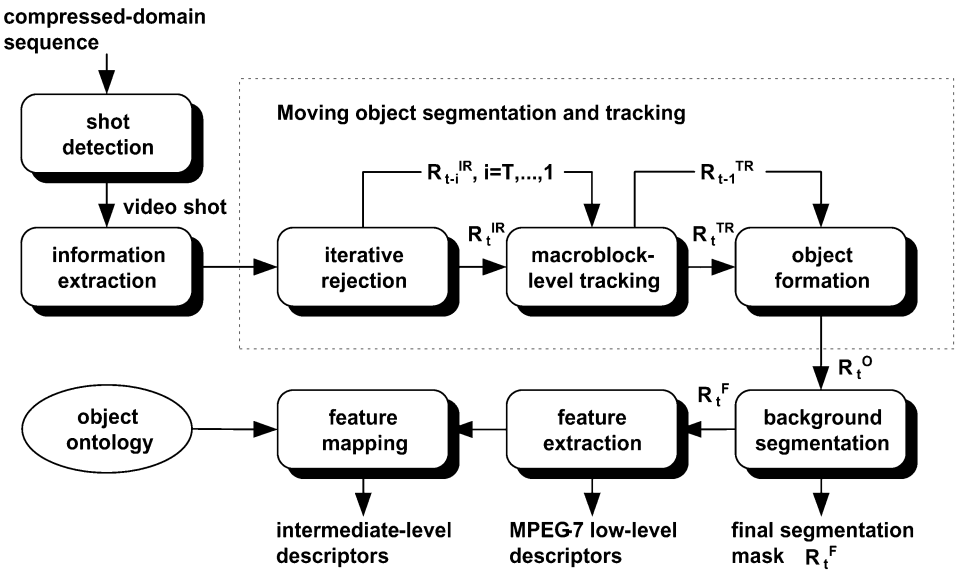


Fig. 1. Overview of the compressed-domain spatiotemporal segmentation algorithm and the feature extraction procedure.

not used. In [21], a method is developed for tracking manually identified moving objects in the compressed stream based on macroblock motion vectors.

To allow efficient indexing of large video databases, an algorithm for the real-time, unsupervised spatiotemporal segmentation of video sequences in the compressed domain is proposed in this paper. Only I- and P-frames are examined, since they contain all information that is necessary for the proposed algorithm; this is also the case for most other compressed-domain algorithms. The bilinear motion model [22] is used to model the motion of the camera (equivalently, the perceived motion of static background) and, wherever necessary, the motion of the identified moving objects. This is seen to produce much better results than the simple clustering techniques using pure translational motion, which have been used by previous methods. An iterative rejection scheme [23] and temporal consistency constraints are employed to deal with the fact that motion vectors extracted from the compressed stream may not represent accurately the actual object motion. Both foreground and background spatiotemporal objects are identified. This is useful, for example, in retrieval applications, where, instead of querying for a compound background, one is allowed to query for its constituent objects, such as sky, sea, or mountain. The proposed spatiotemporal segmentation algorithm is applied to shots; shot detection is performed using the method of [24], chosen because of its computational simplicity.

Furthermore, this paper presents an attempt to bridge the gap between the low-level features extracted from the spatiotemporal objects and the high-level concepts used for querying. This is usually restricted to domain-specific applications [25], [26], where exact mapping of low-level features to objects using *a priori* knowledge is feasible. In contrast to that, the proposed scheme attempts to address the problem of retrieval in generic video collections, where no possibility of structuring a domain-specific knowledge base exists. In such generic collections, the *query-by-example* paradigm [27] is usually employed. This is based on the assumption that the user has access to a clip which represents what the user seeks, which is not very realistic [2] and, for this reason, other query strategies have recently been proposed, such as the *query-by-sketch* paradigm presented in [1]. In [2], the problem of bridging the gap between low-level representation and high-level semantics is formulated as a probabilistic pattern recognition problem. In [28] and [29], hybrid methods extending the query-by-example strategy are developed.

In the proposed indexing and retrieval scheme, instead of adopting the query-by-example strategy, the spatiotemporal segmentation algorithm is combined with simple *ontologies* [30]–[33] and a *relevance feedback* mechanism [34]–[36] based on support vector machines [37], [38]. This scheme (Fig. 1) allows for MPEG-7 compliant low-level indexing features to be extracted for the spatiotemporal objects and subsequently be associated with higher level descriptors that humans are more familiar with; these are used to restrict the search to a set of potentially relevant spatiotemporal objects. Final query results are produced after one or more rounds of relevance feedback, which result in those shots containing relevant objects being ranked higher than others initially identified as potentially relevant.

The paper is organized as follows. In Section II, the extraction of information from the compressed stream is discussed. In Section III, moving object segmentation and tracking methods are developed. Section IV deals with background segmentation. The indexing and retrieval scheme making use of the proposed segmentation algorithm is discussed in Section V. Section VI contains experimental evaluation of the developed methods, and, finally, conclusions are drawn in Section VII.

II. Compressed-Domain Information Extraction

The information used by the proposed segmentation algorithm is extracted from MPEG-2 [39] sequences during the decoding process. Specifically, motion vectors are extracted from the P-frames and are used for foreground/background segmentation and for the subsequent identification of different foreground objects. Since P-frames are coded using motion information from I-frame to P-frame or from P-frame to P-frame, their motion information provides a clearer indication of the motion

TABLE I
MASKS USED DURING FOREGROUND OBJECT SEGMENTATION

Symbol	Description
R_t^{IR}	foreground/background mask, output of the iterative rejection process of section III-B
R_t^{TR}	foreground/background mask, output of the macroblock-level tracking of section III-C
R_{t-i}^{temp}, $i = 0, \ldots, T$	temporary foreground/background masks, used during macroblock-level tracking
R_t^I	mask created from R_t^{IR} by clustering foreground macroblocks to connected regions s_k^t, $k = 1, \ldots, \kappa^t$ (section III-D)
R_t^O	foreground spatiotemporal object mask, output of the spatiotemporal object formation process of section III-D

of an object compared to motion information derived from temporally adjacent frames. In order to derive motion information for the I-frames, averaging of the motion vectors of the P-frames that are temporally adjacent to the given I-frame is performed, rather than block matching employed in previous work on compressed-domain tracking [21].

In order to further segment the background to its constituent objects (e.g., sky and grass), the use of color information is essential; this is due to the fact that the background has already been identified, using the motion information, as a nonconnected region of uniform motion. The color information extracted for the purpose of background segmentation is restricted to the dc coefficients of the macroblocks, corresponding to the Y, Cb, and Cr components of the MPEG color space. A single dc coefficient is used to describe luminance (Y) information for every macroblock. DC coefficients are present in I-frames and intracoded macroblocks of P-frames but not in intercoded macroblocks. However, they may be conveniently found in the latter by extrapolation using the method in [40]. Alternatively, motion information can be used for temporal tracking in P-frames of the background regions formed in I-frames using color information. The latter technique is employed in this work.

III. MOVING OBJECT SEGMENTATION AND TRACKING

A. Overview

The extraction of spatiotemporal moving objects is the key challenge in any video segmentation algorithm. Before proceeding with the detailed discussion of each step of the moving object segmentation and tracking algorithm, the notion of *spatiotemporal objects* is defined.

Definition: A spatiotemporal object o_q is a set of temporally adjacent spatial regions o_q^t, $t = t_1, t_1 + 1, \ldots, t_2, t_1 < t_2$, all of which are nonempty ($o_q^t \neq \emptyset, \forall\, t \in [t_1, t_2]$) and which for $t \in [t_1 + 1, t_2]$ have been associated with o_q by temporal tracking of spatial region $o_q^{t_1}$, using the framework described in Sections III-D and IV for foreground and background spatiotemporal objects, respectively:

$$o_q = \{o_q^{t_1}, \ldots, o_q^{t_2}\}.$$

The proposed algorithm for moving object extraction is based on exploiting the motion information (motion vectors) of the macroblocks and consists of three main steps.

Step 1) Iterative macroblock rejection is performed in a frame-wise basis to detect macroblocks with motion vectors deviating from the single rigid plane assumption. As a result, certain macroblocks of the current frame are activated (marked as possibly belonging to the foreground).

Step 2) The temporal consistency of the output of iterative rejection over a number of frames is examined, to detect activated macroblocks of the current frame that cannot be tracked back to activated macroblocks for a number of previous frames. These are excluded from further processing (deactivated). This process is based on temporal tracking of activated macroblocks using their motion vectors.

Step 3) Macroblocks still activated after step 2 are clustered to connected regions that are in turn assigned to either preexisting or newly appearing spatiotemporal objects, based on the motion vectors of their constituent macroblocks. Spatial and temporal constraints are also applied to prevent the creation of spatiotemporal objects inconsistent with human expectation (e.g., single-macroblock objects or objects with undesirably small temporal duration).

The above steps are explained in more detail in the sequel. An overview of the different segmentation masks used in this process is shown in Table I.

B. Iterative Macroblock Rejection

Iterative rejection is a method originally proposed in [41] for global motion estimation using the output of a block matching algorithm (BMA) and a four-parameter motion model. In [23], the method was extended to the estimation of the eight parameters of the bilinear motion model, used in turn for the retrieval of video clips based on their global motion characteristics. This method is based on iteratively estimating the parameters of the global-motion model using least-square estimation and rejecting those blocks whose motion vectors result in larger than average estimation errors. The iterative procedure is terminated when one iteration leaves the set of rejected blocks unaltered. The underlying assumption is that the background is significantly larger than the area covered by the moving objects; thus, the application of the iterative rejection scheme to the entire frame results in motion vectors

affected by local (object) motion being rejected, and global motion (camera motion or, equivalently, the perceived motion of still background) being estimated.

In this study, iterative rejection based on the bilinear motion model is used to generate the mask R_t^{IR}, indicating which macroblocks have been rejected at time t (or activated, from the segmentation objective's point of view). This is the first step of foreground/background segmentation. Rejected (activated) macroblocks are treated as potentially belonging to foreground objects. Compared to classical methods based on examining the temporal change of color features [42] for purposes of fast raw-domain foreground/background segmentation, the employed method of iterative rejection enjoys high efficiency, especially when dealing with sequences captured by a moving camera.

Although this is a fast and relatively simple method for detecting macroblocks that belong to the foreground, several macroblocks may be falsely activated. This may be due to inaccurate estimation of motion vectors from the compressed stream or to inability of the motion model to accurately capture the undergoing global motion. In order to identify and discard falsely activated macroblocks, the temporal consistency of the output of iterative rejection over a number of previous frames is examined, as discussed in Section III-C.

C. Macroblock-Level Tracking

In order to examine the temporal consistency of the output of iterative rejection, activated macroblocks are temporally tracked using the compressed-domain motion vectors. The temporal tracking is based upon the work presented in [21], where objects are manually marked by selecting their constituent macroblocks and these objects are subsequently tracked in the compressed domain using the macroblock motion vectors. However, in contrast to the method in [21], the proposed method requires no human intervention for the selection of the macroblocks to be tracked. A shortcoming of the method in [21] is the need for block matching in order to extract motion features for the I-frames. This is avoided in the present study by averaging the motion vectors of the P-frames that are temporally adjacent to the given I-frame, as already discussed in Section II.

More specifically, let $\tau(\cdot)$ be the tracking operator realizing the tracking process of [21], whose input is a macroblock at time t and its output is the corresponding macroblock or macroblocks at time $t+1$. This correspondence is established by estimating the overlapping of the examined macroblock with its spatially adjacent ones, determined using the displacement indicated by its motion vector. Then, the operator $\mathcal{T}(\cdot)$ is defined as having a mask (such as R_t^{IR}) as input, applying the $\tau(\cdot)$ operator to the set of all foreground macroblocks of that mask, and outputting the corresponding mask at time $t+1$.

Let R_t^{TR} denote the output foreground/background mask derived via macroblock-level tracking, using masks R_{t-i}^{IR}, $i = T, \ldots, 0$. The derivation of mask R_t^{TR}, using the operator $\mathcal{T}(\cdot)$ to evaluate and enforce the temporal consistency of the output of iterative rejection over T frames, can be expressed as

$$\begin{aligned} R_{t-T}^{\text{temp}} &= R_{t-T}^{IR} \\ \text{for } i = T, \ldots, 1, \quad R_{t-i+1}^{\text{temp}} &= \mathcal{T}(R_{t-i}^{\text{temp}}) \cap R_{t-i+1}^{IR} \\ R_t^{TR} &= R_t^{\text{temp}} \end{aligned}$$

where $\cap$ denotes the intersection of foreground macroblocks and R_{t-i}^{temp}, $i = T, \ldots, 0$, is a set of temporary foreground/background segmentation masks.

It is important to observe that the above process does not lead to infinite error propagation: if a macroblock is falsely assigned to the background, this will affect at most the T subsequent frames. Further, it may affect only the current frame, since the tracking process, as explained in [21], results in the inflation of tracked regions (in this case, the foreground part of the foreground/background mask). The efficiency of macroblock-level tracking in rejecting falsely activated macroblocks is demonstrated in Fig. 2 for frame 220 of the "penguin" sequence.

D. Spatiotemporal Object Formation

After the rejection of falsely activated macroblocks, as described in Section III-C, the remaining macroblocks are clustered to connected foreground regions and are subsequently assigned to foreground spatiotemporal objects. Clustering to connected regions s_k^t, $k = 1, \ldots, \kappa^t$, is performed using a four-connectivity component labeling algorithm [43]; this results in the creation of mask R_t^I. As will be discussed in the sequel, this does not necessarily imply that each of these connected spatial regions in R_t^I belongs to a single spatiotemporal object o_q. Only for the first frame of the shot, in the absence of a previous object mask R_t^O (i.e., the output of the region formation and tracking step, expressing the spatiotemporal object membership of each macroblock), each connected region is assumed to correspond to a single object as follows:

$$\text{for } t = 0, \quad R_t^O = R_t^I.$$

To determine whether a given spatial region belongs to one or more preexisting spatiotemporal objects or to a newly appearing one and to eventually create the object mask R_t^O, motion projection is performed by applying the tracking operator $\tau(\cdot)$ to the macroblocks of each spatiotemporal object of mask R_{t-1}^O. Thus, every connected region s_k^t of mask R_t^I can be assigned to one of the following three categories.

1) A number of macroblocks $M_{k,q}$, $M_{k,q} \geq S_{k,q}$, of s_k^t have been assigned to spatiotemporal object o_q in mask $\mathcal{T}(R_{t-1}^O)$, and no macroblock of s_k^t has been assigned to a spatiotemporal object o_m, $m \neq q$.
2) A number of macroblocks $M_{k,q}$, $M_{k,q} \geq S_{k,q}$, of s_k^t have been assigned to spatiotemporal object o_q in mask $\mathcal{T}(R_{t-1}^O)$, and one or more macroblocks of s_k^t have been assigned to different spatiotemporal objects, namely o_m, $m = 1, \ldots, M$.
3) There is no spatiotemporal object o_q in mask $\mathcal{T}(R_{t-1}^O)$ having $M_{k,q}$ macroblocks of s_k^t, $M_{k,q} \geq S_{k,q}$, assigned to it.

The parameter $S_{k,q}$ in the definition of the above categories is estimated for every pair of a spatial region s_k^t of mask R_t^I and the projection of a spatiotemporal object o_q in mask $\mathcal{T}(R_{t-1}^O)$. Let M_k^s, M_q^o denote the size in macroblocks of the examined pair (s_k^t and motion projection of object o_q in mask $\mathcal{T}(R_{t-1}^O)$ respectively). Then, the parameter $S_{k,q}$ is calculated as follows:

$$S_{k,q} = a \cdot \frac{M_k^s + M_q^o}{2}. \tag{1}$$

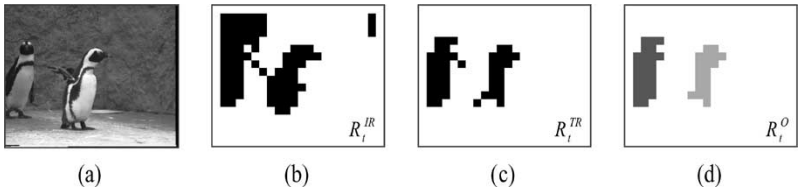


Fig. 2. Frame 220 of the "penguin" sequence. (a) Original image. (b) Output of iterative rejection R_t^{IR}. (c) Activated macroblocks after macroblock-level tracking for $T = 4$ (mask R_t^{TR}). (d) Final results showing the two spatiotemporal objects present in this frame (mask R_t^O). The usefulness of macroblock-level tracking in rejecting falsely activated macroblocks is evident.

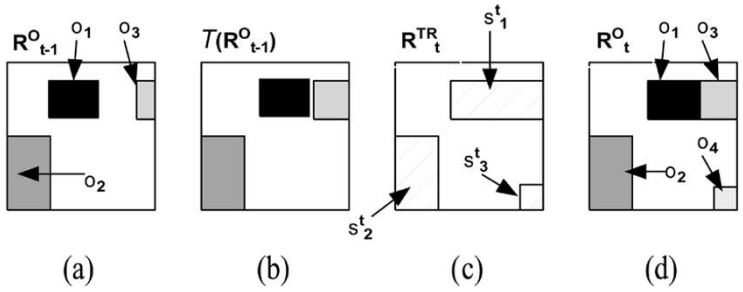


Fig. 3. Synthetic example of the spatiotemporal object formation process of Section III-D. Using the object mask for the previous frame, (a) R^O_{t-1}, (b) mask $\mathcal{T}(R^O_{t-1})$ is created. Comparison of the latter with (c) mask R_t^{TR} reveals the presence of one spatial region (s_1^t) that belongs to the second category (can be associated with spatiotemporal objects o_1 and o_3), one that belongs to the first category (s_2^t, can be associated only with o_2) and one that belongs to the third category (s_3^t). Treating the three different cases as described in Section III-D, (d) the object mask R_t^O is formed.

The value of the parameter a was set to 0.5 on the basis of experimentation.

Obviously, the spatial regions s_k^t classified in the third category cannot be associated with an existing spatiotemporal object; therefore, each region of this category forms a new spatiotemporal object.

Similarly, the spatial regions s_k^t classified in the first category can only be associated with a single spatiotemporal object o_q. However, more than one spatial regions may be associated with the same spatiotemporal object. In this case, the larger spatial region becomes part of o_q, while the rest are discarded (their macroblocks are assigned to the background). This procedure is intended to deal with objects breaking up: the fragments that are discarded at time t, if they actually correspond to moving objects, will clearly be assigned to category 3) at time $t+1$ and in turn be identified as new spatiotemporal objects.

As for the regions s_k^t classified in the second category, the initial correspondence of specific macroblocks belonging to s_k^t with spatiotemporal objects is employed so as to estimate the parameters of the bilinear motion model for each of the competing objects. Subsequently, each macroblock is assigned to the object for which the motion estimation error in minimized. This process elegantly handles moving objects that become spatially adjacent. The possibility of merging two adjacent objects, leaving unaltered any masks created at time $t-i$, $i>0$, is also examined by estimating the parameters of their common motion model (at time t) and comparing the corresponding mean-square error with those of the motion models estimated for each object separately.

The process of moving object segmentation and tracking is terminated by imposing application-oriented restrictions regarding the size and temporal duration of valid moving objects, if any such restrictions exist. Generally, the removal of very small objects and objects of very small temporal duration is beneficial, since they are most likely to be false objects and are of little use in the proposed indexing and retrieval application. The above procedure is illustrated using real and synthetic masks in Figs. 2 and 3, respectively.

E. Pixel-Domain Boundary Refinement

In specific applications, the information that can be extracted from a segmentation mask of macroblock-level accuracy may be insufficient (e.g., for the extraction of shape descriptors of high accuracy). In this case, pixel-domain processing of a partially decompressed sequence may be required to extract object masks of pixel accuracy. This can be achieved using the color features of pixels in the area of each moving object and a Bayes classifier for two-class separation (moving object/background) to reclassify all pixels in that area or a portion of them, in a fashion similar to that of [44]. Pixel-accuracy masks created using this refinement method are presented in the experimental results section.

IV. Background Segmentation

After foreground spatiotemporal objects have been extracted, background segmentation is performed based on classifying the remaining macroblocks (assigned to the background) to one of a number of background spatiotemporal objects. This task is performed using two distinct steps, each dealing with one of the different types of the examined frames. These steps are preceded at the beginning of the shot, by a procedure for the estimation of the number of background objects that should be created. The result of the background segmentation is a final segmentation mask R_t^F.

Background segmentation begins by applying the *maximin* algorithm [45] to the color dc coefficients of the first frame, which is an I-frame. The maximin algorithm employs the Euclidean distance in the YCrCb colorspace to identify radically different

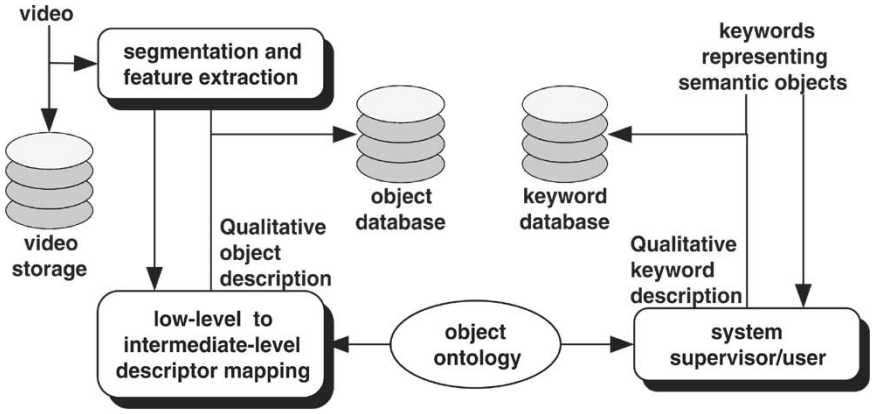


Fig. 4. Indexing system overview. Low-level and intermediate-level descriptor values for the spatiotemporal objects are stored in the object database; intermediate-level descriptor values for the user-defined keywords are stored in the keyword database.

colors; these indicate the presence of different background spatiotemporal objects. Its output is the number of estimated objects and their corresponding colors, which are used to initiate the clustering process.

In I-frames, background macroblocks are clustered to background objects using the K-means algorithm [45], [46], where K is the number of objects estimated by the maximin algorithm. For the first frame of the shot, the color centers are initialized using the output of the maximin algorithm, while for the subsequent I-frames the color centers of the resulting objects in the previous I-frame are used for initialization. The connectivity of the K objects is enforced using a recursive component labeling algorithm [43] to identify small nonconnected parts, which are subsequently assigned to a spatially adjacent object on the basis of color similarity. The connectivity constraint is useful in accurately estimating the position of each object, which could otherwise comprise nonconnected parts scattered in the entire frame.

In P-frames, the absence of color information can be dealt with by using the macroblock motion vectors and a previous final mask R^F_{t-1}. Temporal tracking is then performed as discussed in Sections II and III: macroblocks that are associated via the tracking process with more than one background objects are assigned to the one for which the motion information indicates a stronger association (i.e., a higher degree of overlapping with it, as in Section III-C), while those not associated with any object are assigned to one on the basis of spatial proximity.

V. Object-Based Indexing and Retrieval Using a Shot Ontology

A. Overview

The proposed segmentation algorithm is suitable for introducing object-based functionalities to video indexing and retrieval applications, due to the formation of both foreground and background spatiotemporal objects, for which object-based descriptors in the context of the MPEG-7 Visual standard [47] can be extracted. Examples of such standardized descriptors include the *dominant color descriptor*, the *scalable color descriptor*, *contour-based* and *region-based* shape descriptors, and *motion trajectory* and *parametric motion* descriptors. The use of such object-based descriptors permits the processing of more expressive queries and makes indexing and retrieval more efficient, compared to key-frame based indexing.

With the exception of a few MPEG-7 descriptors, such as *Motion Activity*, which are fairly high level, most standardized descriptors are low-level arithmetic ones, chosen so as to ensure their usefulness in a wide range of possible applications. These descriptors, however, are not suitable for being directly manipulated by the user of an indexing and retrieval scheme, e.g., for defining the color of a desired object. When examining the specific application of object-based video indexing, it is possible to alleviate this problem by translating certain low-level arithmetic values to intermediate-level descriptors qualitatively describing the object attributes; the latter are preferable, since humans are more familiar with manipulating qualitative descriptors than arithmetic values.

Extending the approach in [32] and [33], the values of the intermediate-level descriptors used for this qualitative description form a simple vocabulary, the *object ontology*. Ontologies are tools for structuring knowledge, defined as the specification of a representational vocabulary for a shared domain of discourse which may include definitions of classes, relations, functions, and other objects. In the proposed scheme, ontologies are used to facilitate the mapping of low-level descriptor values to higher level semantics. An object ontology and a shot ontology are employed to enable the user to form, respectively, a simple qualitative description of the desired objects and their relationships in the shot; in parallel, a qualitative description of each spatiotemporal object in the database is automatically estimated using the object ontology, as will be discussed in Section V-C.

Under the proposed scheme, a query is initiated by the user qualitatively describing the semantic objects and their relations in the desired shot. By comparing the user-supplied qualitative description with the one automatically estimated for each spatiotemporal object, clearly irrelevant ones can be discarded; the remaining, potentially relevant ones are presented to the user at random order. The user then evaluates a subset of them, marking relevant ones simply by checking the appropriate "relevant" box. By submitting this relevance feedback, one or two support vector machines are trained and subsequently rank according to relevance all potentially relevant spatiotemporal objects, using their low-level descriptor values; the shots containing these objects are then presented to the user, ordered by rank. This relevance feedback process can then be repeated, to further enhance the output of the query. The architecture of the indexing scheme and the query procedure are graphically illustrated in Figs. 4 and 5.

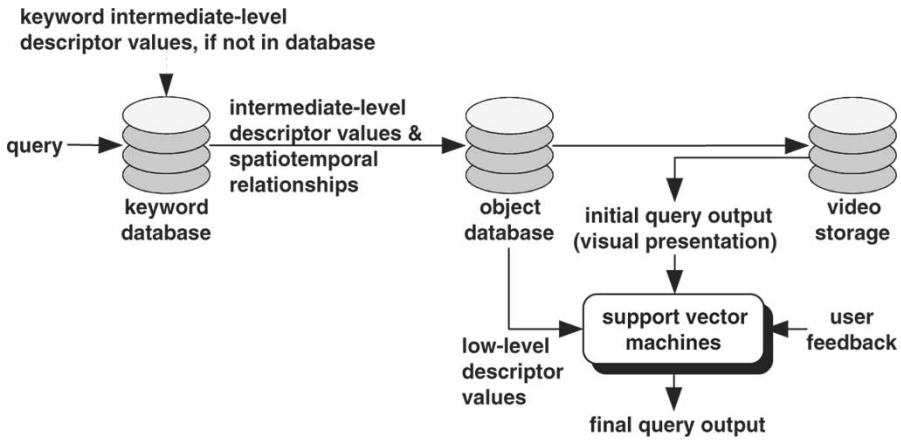


Fig. 5. Query process overview.

TABLE II
SET OF USED MPEG-7 DESCRIPTORS

Descriptor	Video entity described
Motion Activity	shot
Dominant Color	spatiotemporal object
GoF/GoP Color	spatiotemporal object
Contour Shape	spatiotemporal object
Motion Trajectory using "Local" coordinates	spatiotemporal object
Motion Trajectory using "Integrated" coordinates	foreground spatiotemporal object

B. MPEG-7 Descriptors

As soon as a sequence of segmentation masks is produced for each video shot, a set of descriptor values useful in querying the database are calculated for each spatiotemporal object. Standardized MPEG-7 descriptors are used, to allow for flexibility in exchanging indexing information with other MPEG-7 compliant applications. The different MPEG-7 descriptors used in this work are summarized in Table II.

As can be seen from Table II, each object property need not be associated with a single descriptor. For example, in the case of object motion, two different motion trajectories are calculated for each foreground object, as the result of the use of two different coordinate systems (values "Local" and "Integrated" of *Spatial 2D Coordinates* descriptor). In the latter case, the use of a fixed, with respect to the camera, reference point for the coordinate system allows the categorization (e.g., fast or slow, direction) of foreground object motion even in the presence of a moving camera. Regarding background objects, using "Local" coordinates is more appropriate, since the goal is not to extract speed characterization or motion direction but rather their qualitative space-localization in the frame. To facilitate the policing of spatial relations between foreground and background objects, a trajectory using "Local" coordinates is calculated for the former, as well.

Two MPEG-7 "color" descriptors are also used; unlike motion descriptors, they both apply to all objects. This duality serves the purpose of satisfying the diverse requirements set by the general architecture of the retrieval scheme: low-level descriptors should be easy to map to intermediate-level qualitative descriptors (e.g., names of basic colors) and still permit accurate retrieval. A few most-dominant colors of the *Dominant Color* descriptor are most appropriate for associating with color-names, whereas when using the low-level descriptors directly, color histograms (*GoF/GoP Color*) demonstrate better retrieval performance [48]; they also have the advantage of being compatible with the L2 norm used as part of the employed relevance feedback mechanism.

C. Object and Shot Ontologies

1) Object Ontology: In this study, ontologies [49], [50] are employed to allow the user to query a video collection using semantically meaningful concepts (semantic objects), without the need for performing manual annotation of visual information. A simple *object ontology* is used to enable the user to describe semantic objects, like "tiger", using a vocabulary of intermediate-level descriptor values. These are automatically mapped to the low-level descriptor values calculated for each spatiotemporal object in the database, thus allowing the association of keywords representing semantic objects (e.g., the "tiger" keyword) and potentially relevant spatiotemporal objects. The simplicity of the employed object ontology permits its applicability to generic video collections without requiring the correspondence between spatiotemporal objects and relevant descriptors to be defined manually. This object ontology can be expanded so as to include additional descriptors corresponding either to low-level properties (e.g., texture) or to higher level semantics

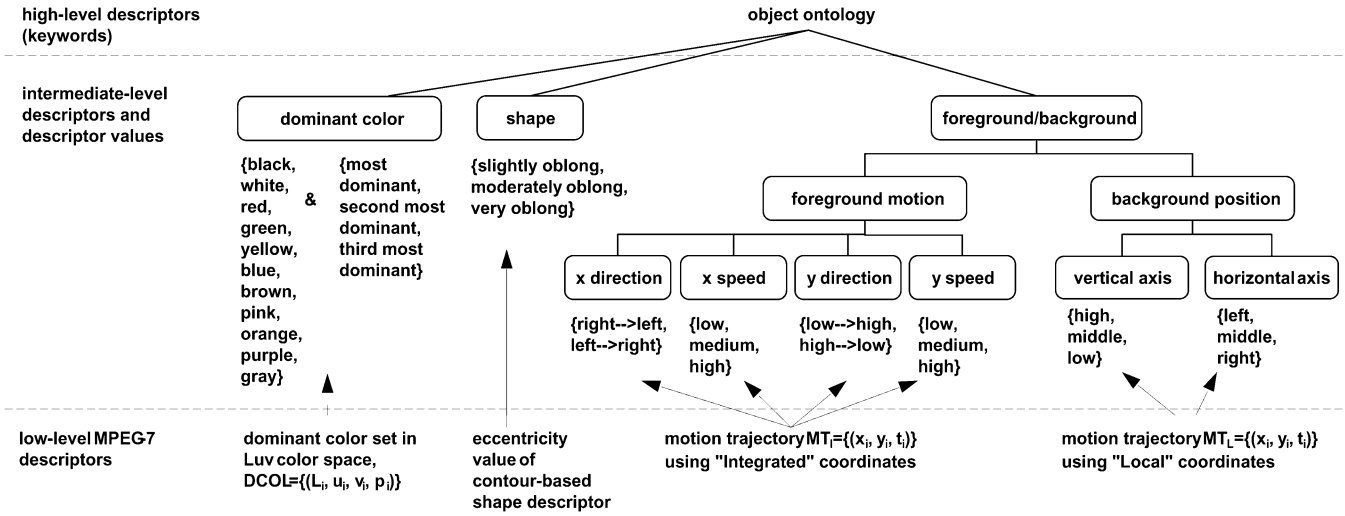


Fig. 6. Object ontology. The correspondence between low-level MPEG-7 descriptors and intermediate-level descriptors is shown.

which, in domain-specific applications, could be inferred either from the visual information itself or from associated information (e.g., subtitles).

The object ontology is presented in Fig. 6, where the possible intermediate-level descriptors and descriptor values are shown. Each value of these intermediate-level descriptors is mapped to an appropriate range of values of the corresponding low-level, arithmetic descriptor. With the exception of color (e.g., "black") and direction (e.g., "low → high") descriptor values, the value ranges for every low-level descriptor are chosen so that the resulting intervals are equally populated. This is pursued so as to prevent an intermediate-level descriptor value from being associated with a plurality of spatiotemporal objects in the database, since this would render it useless in restricting a query to the potentially most relevant ones. Overlapping, up to a point, of adjacent value ranges, is used to introduce a degree of fuzziness to the descriptor values; for example, both "slightly oblong" and "moderately oblong" values may be used to describe a single object.

Let $D_{q,i}$ be the qth descriptor value (e.g., "slightly oblong") of intermediate-level descriptor D_i (e.g., "shape") and $R_{q,i} = [L_{q,i}, H_{q,i}]$ be the range of values of the corresponding arithmetic descriptor z_i. Given the probability density function $pdf(z_i)$ and the factor V expressing the degree of overlapping of adjacent value ranges, the requirement that value ranges should be equally populated defines lower and upper bounds $L_{q,i}$, $H_{q,i}$ which are easily calculated by

$$\begin{aligned} L_{1,i} &= L_i, \\ \int_{L_{q-1,i}}^{L_{q,i}} pdf(z_i)dz_i &= \frac{1-V}{Q_i - V\cdot(Q_i-1)}, \\ q &= 2,\ldots,Q_i \end{aligned} \tag{2}$$

$$\int_{L_{1,i}}^{H_{1,i}} pdf(z_i)dz_i = \frac{1}{Q_i - V\cdot(Q_i-1)} \tag{3}$$

$$\begin{aligned} \int_{H_{q-1,i}}^{H_{q,i}} pdf(z_i)dz_i &= \frac{1-V}{Q_i - V\cdot(Q_i-1)}, \\ q &= 2,\ldots,Q_i \end{aligned} \tag{4}$$

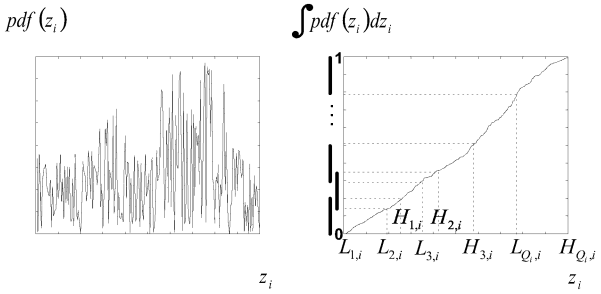


Fig. 7. Graphical illustration of the derivation of (2)–(4).

where Q_i is the number of descriptor values defined for the examined descriptor D_i (for example, for "shape," $Q_i = 3$), and L_i is the lower bound of the values of variable z_i. The overlapping factor V was selected equal to $V = 0.25$ in our experiments. The derivation of (2)–(4) is graphically illustrated in Fig. 7. Since all value ranges $R_{q,i}$, $q = 1,\ldots,Q_i$, are equally populated, the quantity $\int_{L_{q,i}}^{H_{q,i}} pdf(z_i)dz_i$, which equals the length of the boldfaced linear segments on the left side of the second diagram of Fig. 7, is independent of q. For each pair of adjacent linear segments, the length of their overlapping part is by definition $V\int_{L_{q,i}}^{H_{q,i}} pdf(z_i)dz_i$. Adding the length of all (Q_i) linear segments and then subtracting the length of the parts that, due to overlapping, were counted twice in the initial sum, provides the quantity 1, as illustrated in Fig. 7. Thus, $(Q_i - V(Q_i-1))\int_{L_{q,i}}^{H_{q,i}} pdf(z_i)dz_i = 1$. Equation (3) is directly derived from this for $q = 1$; (2) and (4) are similarly derived.

Regarding color, a correspondence between the 11 basic colors [51], used as color descriptor values, and the values of the hue saturation value (HSV) color space is heuristically defined. More accurate correspondences based on the psychovisual findings of, e.g., [51] and others are possible, as in [52] and [53]; however, this investigation is beyond the scope of the present work. Regarding the direction of motion, the mapping between values for the descriptors "x direction," "y direction," and the MPEG-7 *Motion Trajectory* descriptor is based on the sign of the cumulative displacement of the foreground spatiotemporal objects.

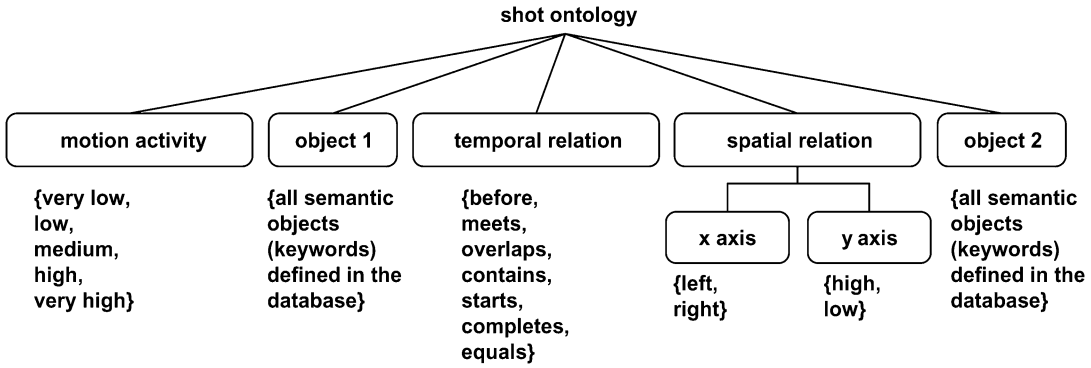


Fig. 8. Shot ontology.

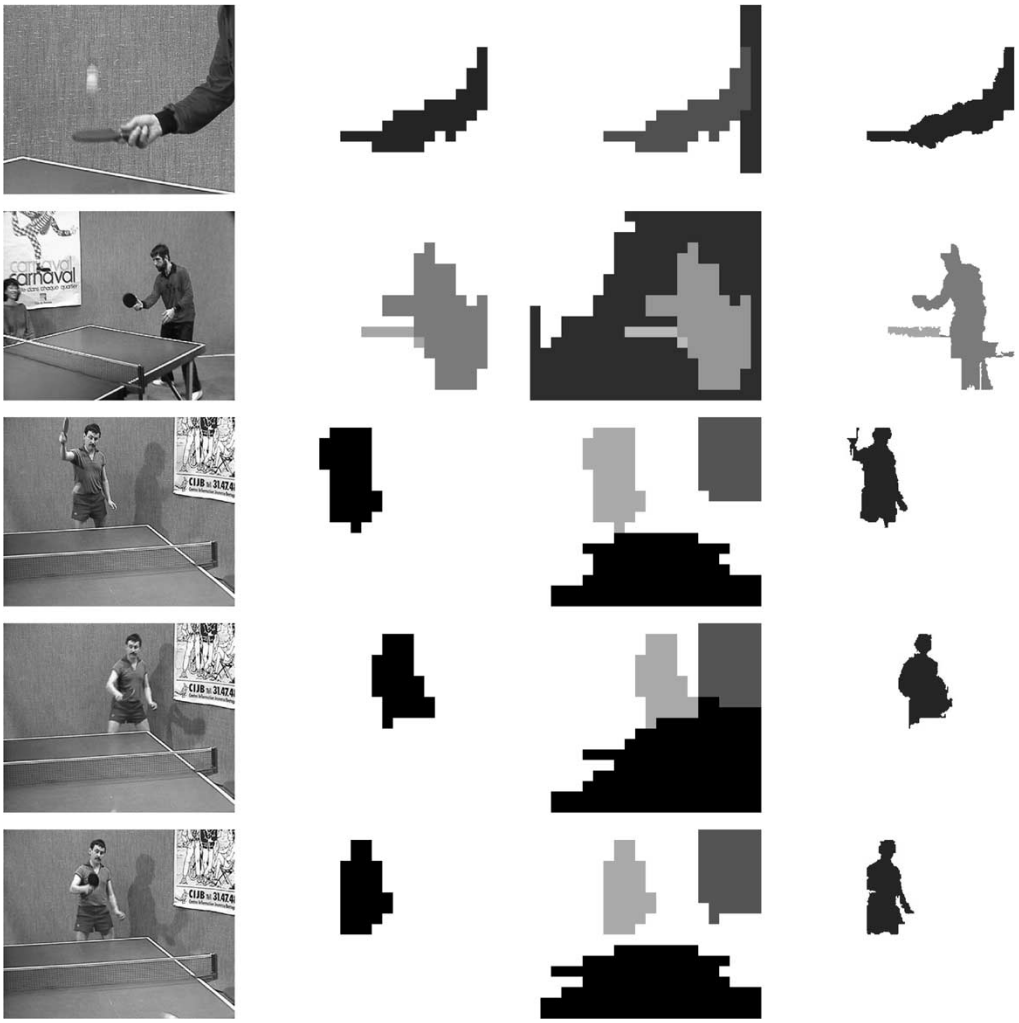


Fig. 9. Results of moving-object detection, final mask after background segmentation, and moving objects after pixel-domain boundary refinement for the "Table-tennis" sequence, frames 10, 88, 154, 214, and 265.

2) Shot Ontology: In order to enable the formulation of descriptive queries, a simple *shot ontology* is defined. As described in Fig. 8, the definition of the shot ontology allows the submission of either single-keyword or dual-keyword queries; however, one can easily extend the shot ontology to allow for multiple-keyword queries. Desired temporal and spatial relationships between the objects can be expressed, along with the specification of the desired motion activity of the shot according to the MPEG-7 *Motion Activity* descriptor. The temporal relations defined in [54] are employed in this study, along with simple spatial relationship descriptors defining the desired position of the second object with respect to the first.

As soon as a query is formulated by describing the desired shot using the shot ontology, the intermediate-level descriptor values associated with each desired semantic object/keyword are compared to those of each spatiotemporal object contained in the database. Descriptors for which no values have been associated with the desired semantic object are ignored; for

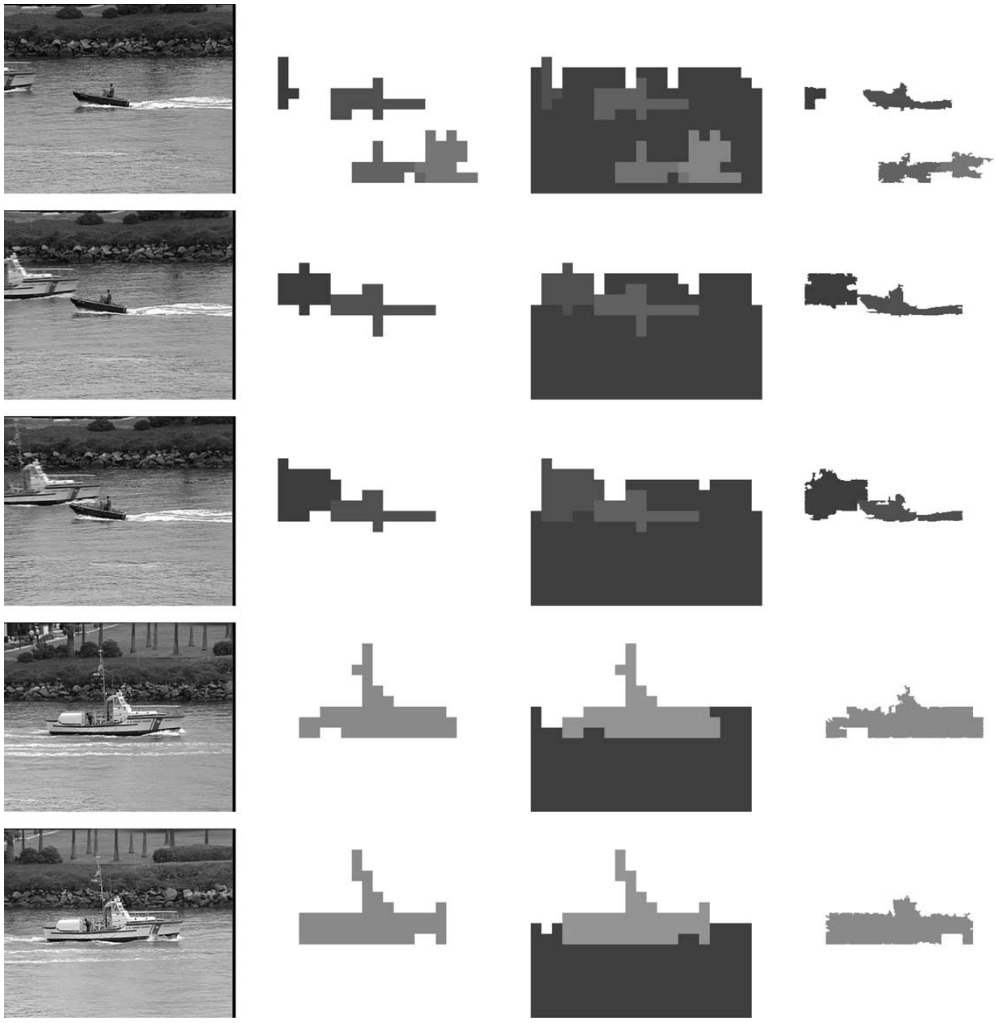


Fig. 10. Results of moving-object detection, final mask after background segmentation, and moving objects after pixel-domain refinement for the "Coast-guard" sequence, frames 10, 28, 37, 202, and 250.

each remaining descriptor, spatiotemporal objects not sharing at least one descriptor value with those assigned to the desired semantic object are deemed irrelevant. In the case of dual-keyword queries, the above process is performed for each desired semantic object separately and only shots containing at least two distinct potentially relevant spatiotemporal objects, one for each keyword, are returned; if desired spatial or temporal relationships between the semantic objects have been defined, compliance with these constraints is checked using the corresponding low-level descriptors, in order to further reduce the number of potentially relevant shots returned to the user.

D. Relevance Feedback

After narrowing down the search to a set of potentially relevant spatiotemporal objects, relevance feedback is employed to produce a qualitative evaluation of the degree of relevance of each spatiotemporal object. The employed mechanism is based on a method proposed in [38], where it is used for image retrieval using global image properties under the query-by-example scheme. This method combines support vector machines (SVM) [37] with a constrained similarity measure (CSM) [38]. Support vector machines employ the user-supplied feedback (training samples) to learn the boundary separating the two classes (positive and negative samples, respectively). Each sample (in our case, spatiotemporal object) is represented by its low-level descriptor vector $\mathbf{F}$. Following the boundary estimation, the CSM is employed to provide a ranking; in [38], the CSM employs the Euclidean distance from the key-image used for initiating the query for images inside the boundary (images classified as relevant) and the distance from the boundary for those classified as irrelevant. Under the proposed scheme, no key-image is used for query initiation; the CSM is therefore modified so as to assign to each spatiotemporal object classified as relevant the minimum of the Euclidean distances between it and all positive training samples (i.e., spatiotemporal objects marked as relevant by the user during relevance feedback).

The above relevance feedback technique was realized using the SVM software libraries of [55]. The Gaussian radial basis function is used as a kernel function by the SVM, as in [38], to allow for nonlinear discrimination of the samples. The low-level descriptor vector $\mathbf{F}$ is composed of the 256 values of the histogram (*GoF/GoP Color* descriptor) along with the *eccentricity* value of the *Contour Shape* descriptor and either the position or the speed in the x and y axes, depending on whether the examined spatiotemporal object belongs to the background or the

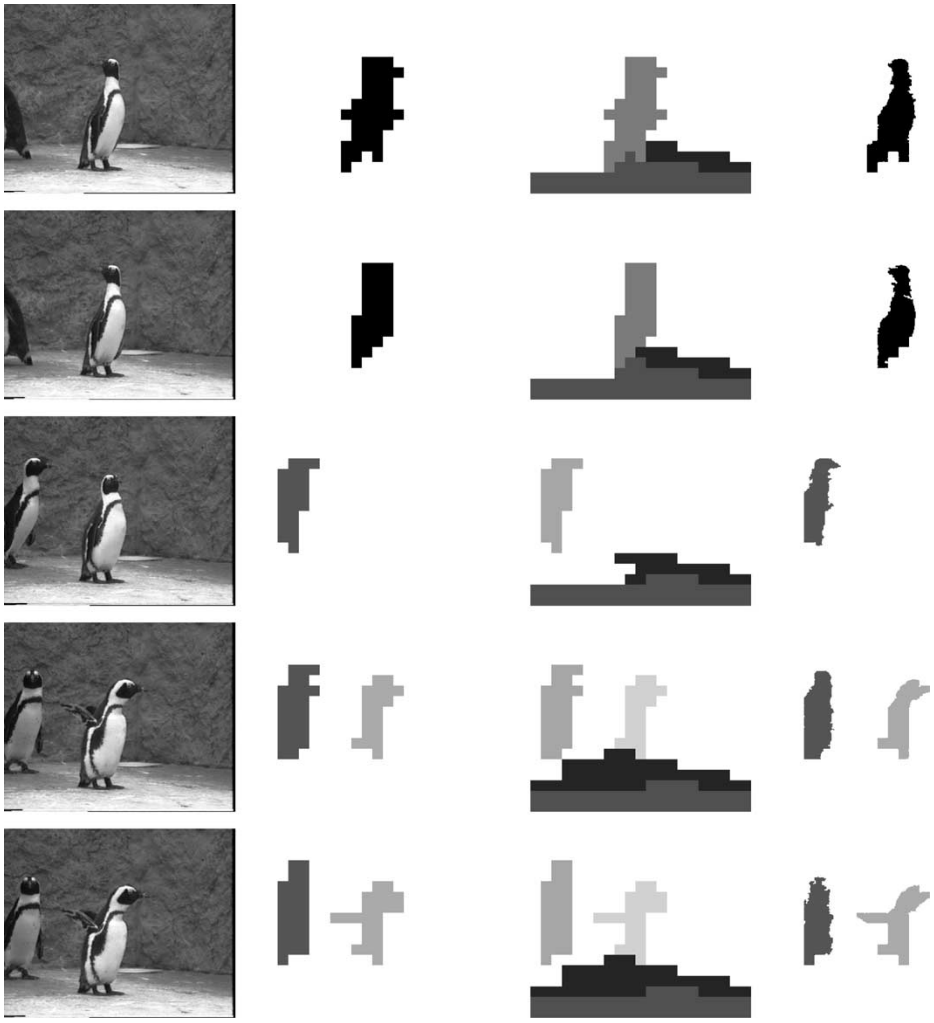


Fig. 11. Results of moving-object detection, final mask after background segmentation, and moving objects after pixel-domain refinement for the "Penguin" sequence, frames 1, 7, 175, 220, and 223.

foreground respectively. In the case of dual-keyword queries, two different SVMs are independently trained and the shot rank is calculated as the sum of the two ranks. This relevance feedback process can be repeated as many times as necessary, each time using all previously supplied training samples.

Furthermore, it is possible to store the parameters of the trained SVM and the corresponding training set for every keyword that has already been used in a query at least once. This endows the system with the capability to respond to anticipated queries without initially requiring any feedback; in a multiuser (e.g., web-based) environment, it additionally enables different users to share knowledge, either in the form of semantic object descriptions or in the form of results retrieved from the database. In either case, further refinement of retrieval results is possible by additional rounds of relevance feedback.

VI. Experimental Results

The proposed algorithms and methodologies were tested on known test sequences, as well as a collection of 812 video shots created by digitizing parts of movies and collecting video clips available on the internet.

Results of the real-time compressed-domain segmentation algorithm are presented for the "Table-tennis" (Fig. 9), "Coast-guard" (Fig. 10), "Penguin" (Fig. 11), and "Stair" (Fig. 12) sequences in CIF format. Segmentation masks both before (R_t^O, second column of Figs. 9–12) and after the background segmentation (R_t^F, third column of Figs. 9–12) are presented, to clearly demonstrate the foreground and background objects identified in the compressed stream by the proposed algorithm. Results after the application of pixel-domain boundary refinement (Section III-E) to the moving objects are also presented in the aforementioned figures. It is seen that the proposed algorithm succeeds in extracting the actual foreground objects depicted in the sequences. No oversegmentation is caused by the proposed approach and thus the formation of meaningful spatiotemporal objects is facilitated. Additionally, very few false objects are created. Moving objects that have halted, as the rightmost penguin in Fig. 11, are assigned new labels when they resume moving.

The proposed segmentation approach imposes little additional computational burden to the computational complexity of a standard MPEG decoder. Excluding any processes of the MPEG decoder, the proposed compressed-domain segmentation algorithm requires on the average 5.02 ms per processed CIF-format I/P-frame on an 800-MHz Pentium III. This translates to almost 600 frames per second considering the presence of two consecutive B-frames between every two I/P-frames,

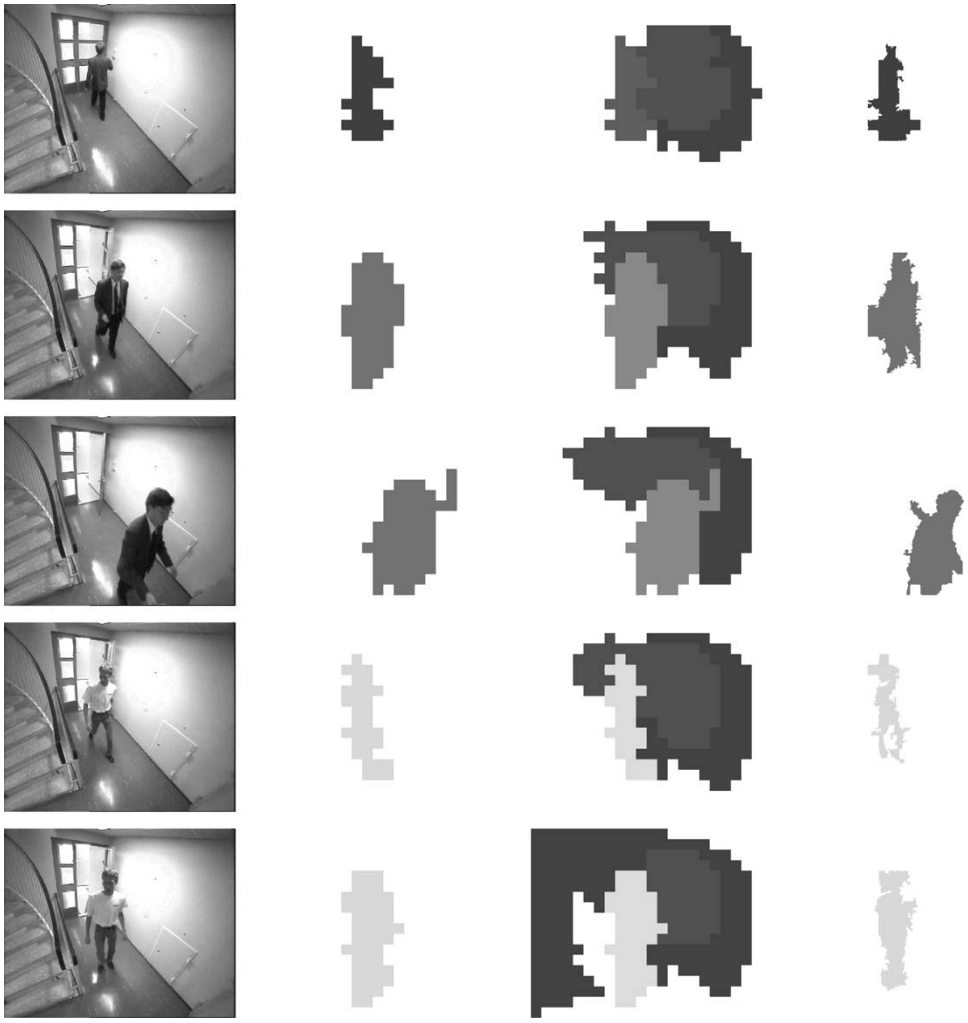


Fig. 12. Results of moving-object detection, mask after background segmentation, and moving objects after pixel-domain refinement for the "Stair" sequence, frames 238, 661, 679, 1330, and 1336.

which is typical for MPEG-2 sequences and is the case for the employed test media. The pixel-domain boundary refinement of Section III-E requires on the average 0.48 seconds per processed I/P-frame. The latter procedure is not necessary for applications like the indexing and retrieval scheme proposed in this study.

The segmentation algorithm of Sections III and IV was subsequently applied to a collection of 812 video shots used for indexing and retrieval experiments, resulting to the creation of 3058 spatiotemporal objects. MPEG-7 low-level descriptors were calculated for each of the created objects, as described in Section V-B. Following that, the mapping between these low-level descriptors and the intermediate-level descriptors defined by the object ontology was performed; this was done by estimating the low-level-descriptor lower and upper boundaries corresponding to each intermediate-level descriptor value, as discussed in Section V-C. Since a large number of heterogeneous spatiotemporal objects was used for the initial boundary calculation, future insertion of heterogeneous video clips to the database is not expected to significantly alter the proportion of spatiotemporal objects associated with each descriptor value; thus, the mapping between low-level and intermediate-level descriptors is not to be repeated, regardless of future insertions.

The next step in the experimentation with the proposed system was to use the object ontology to define, using the available intermediate-level descriptors, high-level concepts, i.e., semantic objects. Since the purpose of the first phase of each query is to employ these definitions to reduce the data set by excluding obviously irrelevant spatiotemporal objects, the definitions of semantic objects need not be particularly restrictive; this is convenient from the users' point of view, since the user can not be expected to have perfect knowledge of the color, shape and motion characteristics of the object sought in the database [56]. Two such definitions, namely for the "red car" and "cheetah" keywords, are illustrated in Fig. 13. Subsequently, the shot ontology was employed to form and submit the query. Several experiments were conducted using single-keyword or dual-keyword queries. Ontology-based querying resulted in initial query results produced by excluding the majority of spatiotemporal objects in the database, which were found to be clearly irrelevant.

Finally, one or more pages of fifteen randomly selected, potentially relevant spatiotemporal objects were presented to the user for manual evaluation; the user checked the "relevant" check-box for those that were actually relevant. As a rule, evaluating one or two such pages was found to be sufficient.

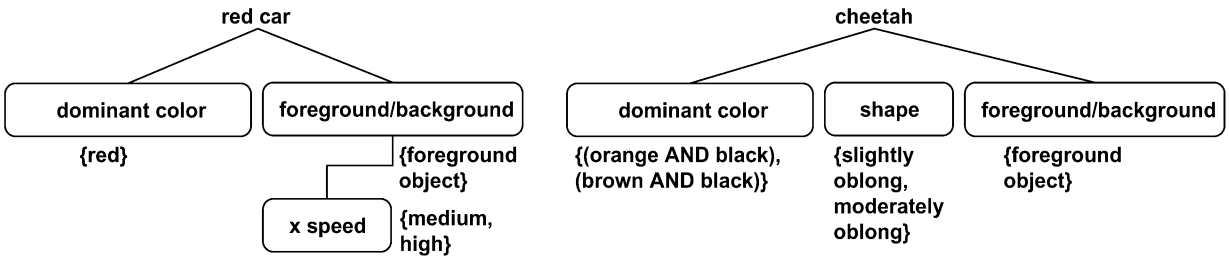
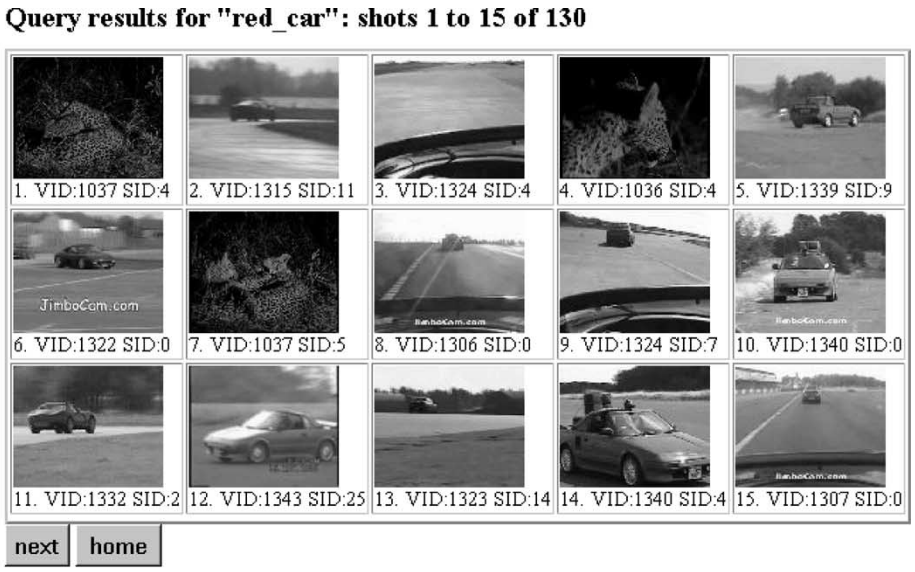
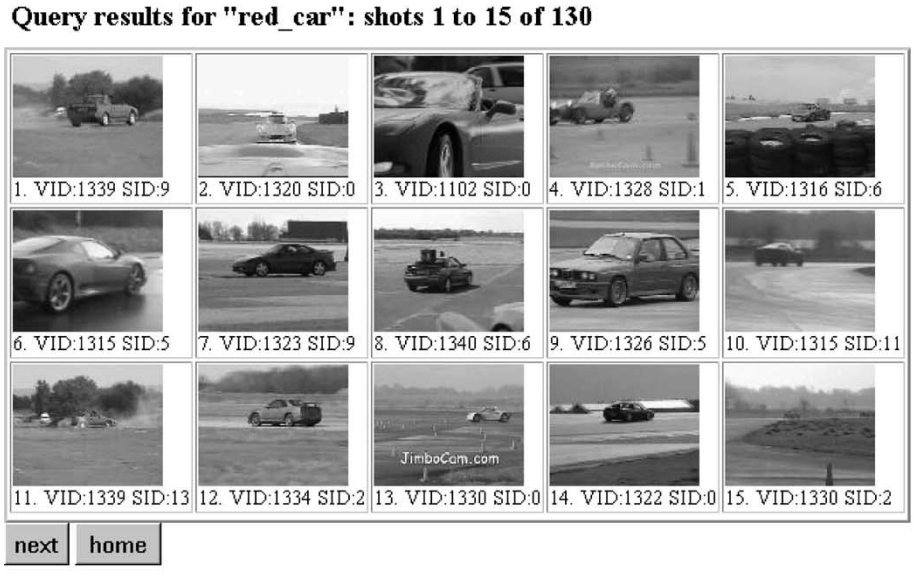


Fig. 13. Exemplary definitions of semantic objects using the object ontology, employed in retrieval experiments.



(a)



(b)

Fig. 14. Results for a "red car" query: (a) shots containing potentially relevant objects, identified using the intermediate-level descriptors and (b) results after one round of relevance feedback.

The average time required for the SVM training and the subsequent object ranking was 0.44 s, on an 800-MHz Pentium III. The fifteen most relevant spatiotemporal objects, according to rank, after the application of one or two rounds of relevance feedback are presented in Figs. 14 and 15 accordingly. Initial query results before the application of relevance feedback are also presented in these figures, demonstrating the improvement achieved during the relevance feedback stage.

Fig. 15. Results for a "cheetah" query: (a) shots containing potentially relevant objects, identified using the intermediate-level descriptors and (b) results after two rounds of relevance feedback.

VII. Conclusion

An algorithm for the unsupervised segmentation of compressed image sequences was presented in this paper, along with an associated video indexing and retrieval scheme. The proposed segmentation algorithm was shown to operate in real-time on a PC, producing semantically meaningful spatiotemporal objects both for the foreground and the background of the shot. Due to its real-time, unsupervised operation, the proposed algorithm is very suitable for content-based multimedia applications requiring the manipulation of large volumes of visual data. The proposed video indexing and retrieval scheme, based on the combination of the proposed segmentation algorithm with ontologies and relevance feedback tools, enabled the formulation of descriptive queries and demonstrated efficient retrieval of visual information.

Acknowledgment

The authors would like to gratefully acknowledge the assistance of COST211 quat.

References

[1] S.-F. Chang, W. Chen, H. J. Meng, H. Sundaram, and D. Zhong, "A fully automated content-based video search engine supporting spatiotemporal queries," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 602–615, Sept. 1998.

[2] M. R. Naphade, I. V. Kozintsev, and T. S. Huang, "A factor graph framework for semantic video indexing," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 40–52, Jan. 2002.

[3] E. Izquierdo, J. R. Casas, R. Leonardi, P. Migliorati, N. O'Connor, I. Kompatsiaris, and M. G. Strintzis, "Advanced content-based semantic scene analysis and information retrieval: The schema project," in *Proc. Workshop on Image Analysis for Multimedia Interactive Services*, London, U.K., Apr. 2003, pp. 519–528.

[4] A. Yoshitaka and T. Ichikawa, "A survey on content-based retrieval for multimedia databases," *IEEE Trans. Knowledge Data Eng.*, vol. 11, pp. 81–93, Jan.–Feb. 1999.

[5] S.-F. Chang, T. Sikora, and A. Puri, "Overview of the MPEG-7 standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 688–695, June 2001.

[6] W. Al-Khatib, Y. F. Day, A. Ghafoor, and P. B. Berra, "Semantic modeling and knowledge representation in multimedia databases," *IEEE Trans. Knowledge Data Eng.*, vol. 11, pp. 64–80, Jan.–Feb. 1999.

[7] P. Salembier and F. Marques, "Region-based representations of image and video: Segmentation tools for multimedia services," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 1147–1169, Dec. 1999.

[8] I. Kompatsiaris, G. Mantzaras, and M. G. Strintzis, "Spatiotemporal segmentation and tracking of objects in color image sequences," in *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS 2000)*, vol. 5, Geneva, Switzerland, May 2000, pp. 29–32.

[9] E. Sifakis and G. Tziritas, "Moving object localization using a multi-label fast marching algorithm," *Signal Processing: Image Commun.*, vol. 16, pp. 963–976, 2001.

[10] N. O'Connor, S. Sav, T. Adamek, V. Mezaris, I. Kompatsiaris, T. Y. Lui, E. Izquierdo, C. F. Bennstrom, and J. R. Casas, "Region and object segmentation algorithms in the qimera segmentation platform," in *Proc. 3rd Int. Workshop Content-Based Multimedia Indexing (CBMI03)*, 2003, pp. 381–388.

[11] A. A. Alatan, L. Onural, M. Wollborn, R. Mech, E. Tuncel, and T. Sikora, "Image sequence analysis for emerging interactive multimedia services-The European COST 211 framework," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 19–31, Nov. 1998.

[12] E. Izquierdo, J. Xia, and R. Mech, "A generic video analysis and segmentation system," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 4, 2002, pp. 3592–3595.

[13] V. Mezaris, I. Kompatsiaris, and M. G. Strintzis, "Video object segmentation using bayes-based temporal tracking and trajectory-based region merging," *IEEE Trans. Circuits Syst. Video Technol.*, to be published.

[14] E. Izquierdo and M. Ghanbari, "Key components for an advanced segmentation system," *IEEE Trans. Multimedia*, vol. 4, pp. 97–113, Mar. 2002.

[15] R. Wang, H.-J. Zhang, and Y.-Q. Zhang, "A confidence measure based moving object extraction system built for compressed domain," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 5, 2000, pp. 21–24.

[16] R. V. Babu and K. R. Ramakrishnan, "Compressed domain motion segmentation for video object extraction," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 4, 2002, pp. 3788–3791.

[17] M. L. Jamrozik and M. H. Hayes, "A compressed domain video object segmentation system," in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, 2002, pp. 113–116.

[18] H.-L. Eng and K.-K. Ma, "Spatiotemporal segmentation of moving video objects over MPEG compressed domain," in *Proc. IEEE Int. Conf. Multimedia and Expo*, vol. 3, 2000, pp. 1531–1534.

[19] N. V. Boulgouris, E. Kokkinou, and M. G. Strintzis, "Fast compressed-domain segmentation for video indexing and retrieval," in *Proc. Tyrrhenian Int. Workshop Digital Communications (IWDC 2002)*, Sept. 2002, pp. 295–300.

[20] O. Sukmarg and K. R. Rao, "Fast object detection and segmentation in MPEG compressed domain," in *Proc. IEEE TENCON*, vol. 3, 2000, pp. 364–368.

[21] L. Favalli, A. Mecocci, and F. Moschetti, "Object tracking for retrieval applications in MPEG-2," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, pp. 427–432, Apr. 2000.

[22] S. Mann and R. W. Picard, "Video orbits of the projective group: A simple approach to featureless estimation of parameters," *IEEE Trans. Image Processing*, vol. 6, pp. 1281–1295, Sept. 1997.

[23] T. Yu and Y. Zhang, "Retrieval of video clips using global motion information," *Electron. Lett.*, vol. 37, no. 14, pp. 893–895, July 2001.

[24] V. Kobla, D. S. Doermann, and K. I. Lin, "Archiving, indexing, and retrieval of video in the compressed domain," in *Proc. SPIE Conf. Multimedia Storage and Archiving Systems*, vol. 2916, 1996, pp. 78–89.

[25] G. Tsechpenakis, G. Akrivas, G. Andreou, G. Stamou, and S. D. Kollias, "Knowledge-assisted video analysis and object detection," presented at the Eur. Symp. Intelligent Technologies, Hybrid Systems and Their Implementation on Smart Adaptive Systems, Algarve, Portugal, Sept. 2002.

[26] R. Visser, N. Sebe, and M. S. Lew, "Detecting automobiles and people for semantic video retrieval," in *Proc. 16th Int. Conf. Pattern Recognition*, vol. 2, 2002, pp. 733–736.

[27] M. R. Naphade, M. M. Yeung, and B. L. Yeo, "A novel scheme for fast and efficient video sequence matching using compact signatures," *Proc. SPIE*, vol. 3972, pp. 564–572, Jan. 2000.

[28] W. Chen and S.-F. Chang, "VISMap: An interactive image/video retrieval system using visualization and concept maps," in *Proc. IEEE Int. Conf. Image Processing*, vol. 3, 2001, pp. 588–591.

[29] S. S. M. Chan, L. Qing, Y. Wu, and Y. Zhuang, "Accommodating hybrid retrieval in a comprehensive video database management system," *IEEE Trans. Multimedia*, vol. 4, pp. 146–159, June 2002.

[30] V. Kashyap, K. Shah, and A. P. Sheth, "Metadata for building the multimedia patch quilt," in *Multimedia Database System: Issues and Research Direction*. Berlin, Germany: Springer-Verlag, 1995, pp. 297–319.

[31] B. Chandrasekaran, J. R. Josephson, and V. R. Benjamins, "What are ontologies, and why do we need them?," *IEEE Intell. Syst.*, vol. 14, pp. 20–26, Jan.–Feb. 1999.

[32] V. Mezaris, I. Kompatsiaris, and M. G. Strintzis, "Ontologies for object-based image retrieval," in *Proc. Workshop Image Analysis For Multimedia Interactive Services*, London, U.K., Apr. 2003, pp. 96–101.

[33] ——, "An ontology approach to object-based image retrieval," in *Proc. IEEE Int. Conf. Image Processing (ICIP03)*, Barcelona, Spain, Sept. 2003, pp. 511–514.

[34] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra, "Relevance feedback: A power tool for interactive content-based image retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 644–655, Sept. 1998.

[35] S. Tong and E. Chang, "Support vector machine active learning for image retrieval," in *Proc. ACM Int. Conf. on Multimedia*, 2001, pp. 107–118.

[36] F. Jing, M. Li, H.-J. Zhang, and B. Zhang, "Region-based relevance feedback in image retrieval," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 4, 2002, pp. 145–148.

[37] V. N. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.

[38] G.-D. Guo, A. K. Jain, W.-Y. Ma, and H.-J. Zhang, "Learning similarity measure for natural image retrieval with relevance feedback," *IEEE Trans. Neural Networks*, vol. 13, pp. 811–820, July 2002.

[39] "Generic Coding of Moving Pictures and Associated Audio Information," MPEG-2, ISO/IEC 13 818, 1996.

[40] B.-L. Yeo, "Efficient processing of compressed images and video," Ph.D. dissertation, Princeton Univ., Princeton, NJ, 1996.

[41] G. B. Rath and A. Makur, "Iterative least squares and compression based estimations for a four-parameter linear global motion model and global motion compensation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 1075–1099, Oct. 1999.

[42] I. Kompatsiaris and M. G. Strintzis, "Spatiotemporal segmentation and tracking of objects for visualization of videoconference image sequences," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, pp. 1388–1402, Dec. 2000.

[43] R. Jain, R. Kasturi, and B. G. Schunck, *Machine Vision*. New York: McGraw-Hill, 1995.

[44] V. Mezaris, I. Kompatsiaris, and M. G. Strintzis, "A framework for the efficient segmentation of large-format color images," in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, 2002, pp. 761–764.

[45] N. V. Boulgouris, I. Kompatsiaris, V. Mezaris, D. Simitopoulos, and M. G. Strintzis, "Segmentation and content-based watermarking for color image and image region indexing and retrieval," *EURASIP J. Appl. Signal Processing*, vol. 2004, no. 4, pp. 418–431, Apr. 2002.

[46] J. McQueen, "Some methods for classification and analyis of multivariate observations," in *Proc. 5th Berkely Symp. on Math. Stat. and Prob.*, vol. 1, 1967, pp. 281–296.

[47] T. Sikora, "The MPEG-7 visual standard for content description—An overview," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 696–702, June 2001.

[48] B. S. Manjunath, J.-R. Ohm, V. V. Vasudevan, and A. Yamada, "Color and texture descriptors," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 703–715, June 2001.

[49] P. Martin and P. W. Eklund, "Knowledge retrieval and the World Wide Web," *IEEE Intell. Syst.*, vol. 15, pp. 18–25, May–June 2000.

[50] A. T. Schreiber, B. Dubbeldam, J. Wielemaker, and B. Wielinga, "Ontology-based photo annotation," *IEEE Intell. Syst.*, vol. 16, pp. 66–74, May-June 2001.

[51] B. Berlin and P. Kay, *Basic Color Terms: Their Universality and Evolution*. Berkeley, CA: Univ. of California, 1969.

[52] J. M. Lammens, "A computational model of color perception and color naming," Ph.D. dissertation, Univ. of Buffalo, Buffalo, NJ, 1994.

[53] A. Mojsilovic, "A method for color naming and description of color composition in images," in *Proc. IEEE Int. Conf. Image Processing*, vol. 2, Rochester, NY, Sept. 2002, pp. 789–792.

[54] Y. F. Day, S. Dagtas, M. Iino, A. Khokhar, and A. Ghafoor, "Spatio-temporal modeling of video data for on-line object-oriented query processing," in *Proc. Int. Conf. Multimedia Computing and Systems*, May 1995, pp. 98–105.

[55] C.-C. Chang and C.-J. Lin. (2001) LIBSVM: A library for support vector machines. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[56] M. R. Naphade and T. S. Huang, "Extracting semantics from audio-visual content: The final frontier in multimedia retrieval," *IEEE Trans. Neural Networks*, vol. 13, pp. 793–810, July 2002.

Vasileios Mezaris (S'98) was born in Athens, Greece, in 1979. He received the Diploma degree in electrical and computer engineering from Aristotle University of Thessaloniki, Greece, in 2001 where he is currently working toward the Ph.D. degree.

He is also a Graduate Research Assistant with the Informatics and Telematics Institute, Thessaloniki, Greece. His research interests include still image segmentation, video segmentation and object tracking, content-based indexing and retrieval.

Mr. Mezaris is a member of the Technical Chamber of Greece.

Ioannis Kompatsiaris (S'94–M'02) received the Diploma degree in electrical engineering and the Ph.D. degree in 3-D model-based image sequence coding from Aristotle University of Thessaloniki (AUTH), Thessaloniki, Greece, in 1996 and 2001, respectively.

He is a Senior Researcher (Researcher D') with the Informatics and Telematics Institute, Thessaloniki. Prior to his current position, he was a Leading Researcher on 2-D and 3-D Imaging at AUTH. His research interests include 2-D and 3-D monoscopic and multiview image sequence analysis and coding, semantic annotation of multimedia content, multimedia information retrieval and knowledge discovery, MPEG-4 and MPEG-7 standards. His involvement with those research areas has led to the coauthoring of one book chapter, 11 papers in refereed journals, and more than 30 papers in international conferences. He has served as a regular reviewer for a number of international journals and conferences. Since 1996, he has been involved in more than 11 projects in Greece, funded by the EC, and the Greek Ministry of Research and Technology. Currently he is the coordinator of the SCHEMA NoE in Content-Based Semantic Scene Analysis and Information Retrieval.

Dr. Kompatsiaris is a member of the Technical Chamber of Greece.

Nikolaos V. Boulgouris (S'96–M'04) received the Diploma and Ph.D. degrees from the University of Thessaloniki, Thessaloniki, Greece, in 1997 and 2002, respectively.

Since September 2003, he has been a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, Univeristy of Toronto, Toronto, ON, Canada. Formerly, he was a Researcher with the Informatics and Telematics Institute, Thessaloniki. During his graduate studies, he held several research and teaching assistantship positions. Since 1997, he has participated in research projects in the areas of image/video communication, pattern recognition, multimedia security, and content-based indexing and retrieval.

Michael G. Strintzis (M'70–SM'80–F'04) received the Diploma degree in electrical engineering from the National Technical University of Athens, Athens, Greece, in 1967, and the M.A. and Ph.D. degrees in electrical engineering from Princeton University, Princeton, NJ, in 1969 and 1970, respectively.

He then joined the Electrical Engineering Department, University of Pittsburgh, Pittsburgh, PA, where he served as an Assistant Professor (1970–1976) and an Associate Professor (1976–1980). Since 1980, he has been a Professor of electrical and computer engineering with the University of Thessaloniki, Thessaloniki, Greece, and, since 1999, Director of the Informatics and Telematics Research Institute, Thessaloniki. His current research interests include 2-D and 3-D image coding, image processing, biomedical signal and image processing, and DVD and Internet data authentication and copy protection.

Dr. Strintzis has been serving as an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR V*ideo* TECHNOLOGY since 1999. In 1984, he was the recipient of a Centennial Medal of the IEEE.