Knowledge – Assisted Video Analysis and Object Detection

Gabriel Tsechpenakis, Giorgos Akrivas, Giorgos Andreou, Giorgos Stamou and Stefanos Kollias
National Technical University of Athens
Department of Electrical and Computer Engineering
Image Video and Multimedia Laboratory
Iroon Politechniou 9 Athens Greece
Phone: +30-10-772-4352, Fax: +30-10-772-2492
email:{gtsech, gakrivas, geand}@image.ntua.gr, {gstam, stefanos}@softlab.ntua.gr

ABSTRACT: Intelligent video analysis is a problem of great importance for applications such as surveillance and automatic annotation. We present, in this paper, a hybrid, knowledge – based approach for object recognition in video sequences. Objects are modelled, in the signal level, through the visual descriptors defined by MPEG-7, the ISO standard for description of audiovisual content and in the semantic level, through the semantic relations defined by MPEG-7. The method of video analysis is synopsised as follows: first, moving regions are extracted using an active contour technique. Second, visual descriptions of the moving regions are extracted and are matched with the ones defined for recognizable objects.

KEYWORDS: Video Analysis, Object Localization, MPEG-7 Visual Descriptors, Semantic Relations, Semantic Entities

INTRODUCTION

Information Retrieval Systems (IRSs) consist of a database, containing a number of documents, an *index*, that associates each document to its related terms, and a *matching mechanism*, that maps the user's query (which consists of *terms*), to a set of associated documents [1]. In the case of multimedia documents, the *content* of the document is difficult to be directly used by the user of the IRS in the query, since matching of multimedia content is not as simple as string matching and *features* of the content must be used instead. The needs for description of multimedia documents' content have been addressed by MPEG-7, the ISO standard for description of multimedia content [2] [pp 688-695]. A large number of MPEG-7 – compliant multimedia descriptions are currently being produced. The standard defines three kinds of features that comprise the description, which are Creation and Usage Information, Structural Information and Semantic Information. The former regards mostly textual information, commonly known as metadata, or metainfo. Structural features express a low-level and machine-oriented kind of description, since they describe content in the form of signal segments and their properties. On the other hand, conceptual features express a high-level and human – oriented kind of description, since they deal with semantic entities, such as objects and events.

Of the former three kinds of multimedia content features, the first one, metainfo, is best retrieved using traditional term – based approaches. The other two kinds correspond to two common approaches found in the literature. The first one, termed *Query by Example (QBE)* [7], requires the user to supply a multimedia document, as an example of the desired document. A *description* of the example, composed of low – level features, is extracted in query time; the index of the system is constructed based on the same features, and the documents whose indexed features best match the example's description are retrieved. The second approach, the *Semantic Indexing (SI)* [8], uses the semantics of the content, i.e. the objects and events found in the documents, as index terms. The user's query is transformed into a semantic query, through the terms' verbal description.

A comparison between the two aforementioned methodologies yields that QBE uses features that can be automatically extracted and compared but, on the other hand, have insufficient expressive power to capture the meaning of a multimedia document, as it is perceived by the human user. Moreover, having to use an exemplar multimedia document makes it difficult for the user to express his or her needs. On the other hand, the semantics of the document is currently difficult be extracted automatically, except in very narrow domains, i.e. domains where the environment is highly controlled.

This paper presents an attempt to bridge this gap between low – level and high – level features. Our approach, which in principle lies on the semantic indexing methods, attempts to extract semantics automatically, by detecting and tracking moving objects in video sequences and then using low-level features of each semantic entity, in order to associate mov-

ing objects with them. Another feature of the proposed method is that it collects signal – level and semantic information about classes of semantic entities into a knowledge base. Semantic entities are described, in the signal level, through the visual descriptors of MPEG-7, particularly those of color, shape and texture. Once the moving objects have been extracted, their visual descriptors are computed and matched with those of the semantic entities, defined in the knowledge base. The matching value is interpreted as the degree of recognition of the respective semantic entity.

For the purpose of moving object detection, this paper proposes a method for tracking objects with low computational cost. Since the issues of shape modeling and object tracking have emerged in the fields of image and video processing and numerous applications related to them, many approaches have been adopted to achieve the most appropriate accuracy or computational cost, depending on the application criteria. Among these approaches, in the last decade a category of deformable templates, entitled active contours, has drawn special attention [9],[10].

So far, active contours have been successfully applied to problems like image segmentation, object detection, localization and tracking in video sequences. A wide range of applications is therefore involved, such as object–based video coding, remote surveillance, content-based retrieval and object recognition. Active contours successfully deal with object distortion due to temporal clutter or changes in viewing geometry. Thus, these deformable models can sufficiently handle natural sequences, obtained by either a static or moving camera, with the presence of noise [10]. On the other hand, the main drawback of these methods is their high computational complexity, which is actually inhibitory for their utilization in real-time applications.

Regarding the object localization and tracking problem, for a sequence acquired by a moving camera, the proposed approaches can be divided into two categories: one deriving constraints for object's 2D motion parameters from the 3D motion of the camera, and another constructing a 2D parametric motion model for the background dominant motion [11],[12]. For the special case of static cameras, the proposed methods can be further divided in two main classes: the feature–based, depending on the extraction of general sequence characteristics and the pixel–based, examining the differences between successive frames using pixels as input features. Based on the motion estimation results, mobile objects are then extracted using a variety of motion segmentation algorithms. In both cases (static and moving camera), using the extracted motion features, various active contour models are then employed to estimate the contours of the mobile objects.

In the proposed method, we focus on the issue of mobile object localization and tracking with the use of an active contour model, trying to improve the procedure's performance in terms of both accuracy and computational cost. The proposed algorithm consists of two main steps: the detection and localization of "*regions-of-interest*" in a sequence [11], and the estimation of the main mobile object contours [13]. We use the term "*regions-of-interest*" to describe the regions in a frame where moving objects are located (in the MPEG-7 sense). These regions are presented by polygons, and each one of them entirely includes only one mobile object. Thus, for each moving object of a scene, we extract its bounding polygon, which is assumed to be the initial estimate for its position and shape. For all the main moving objects of the sequence, after we have derived the respective bounding polygons, we use an active contour model to estimate their accurate contours. Moreover, since we focus on both accuracy and low computational cost, we propose a fast curve evolution method, which can lead to satisfactory object contours. Figure 1: shows the proposed stages of mobile object detection and its contour estimation.

The paper is organized as follows. In section 2, the features of the knowledge base of the proposed system are presented. Section 3 presents the semantic indexing process. Section 4 contains some experimental results. Finally, section 5 discusses possible extensions of our work.

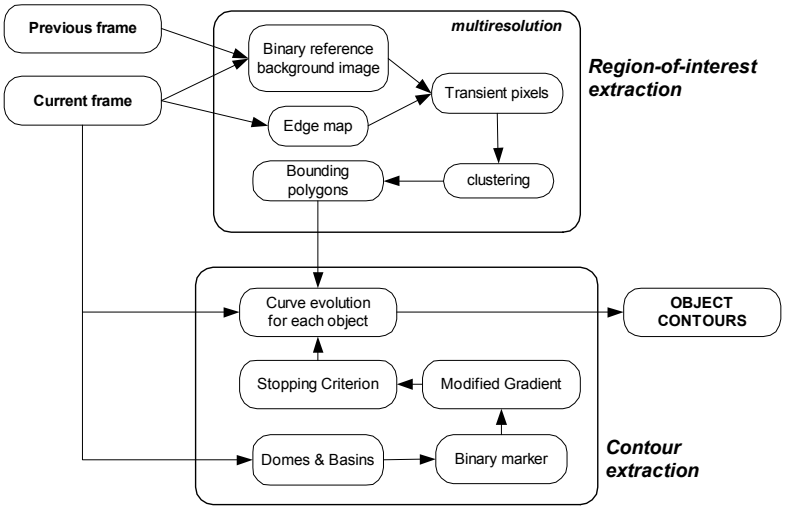


Figure 1: The proposed scheme for object localization and tracking

VISUAL DESCRIPTORS IN THE FRAMEWORK OF KNOWLEDGE REPRESENTATION

Knowledge technologies support information retrieval, because of their abstraction capabilities. So far, however, knowledge has rarely included visual information. In the following, we incorporate the MPEG-7 description schemes in order to model the visual information of concepts into the representation of semantic entities.

The MPEG-7 standard refers to the reality, in which a description makes sense as a *narrative world*. The *description tools* (data structures) that compose a narrative world are the set of its *semantic entities* and the set of its *semantic relations*. The latter is further partitioned into the sets of objects, events, concepts, places, times and states. Each semantic entity can contain textual (annotation), low-level (audiovisual descriptors) and structural (sub-entities) information that is useful for identifying it.

Objects and events can be viewed as corresponding to nouns and verbs of natural language, respectively. It is important that there exist in the same set both specific semantic entities (*instances)*, e.g. "George", "Spanish soccer team" and classes of objects, (*formal abstractions*), e.g. "human" , "soccer team". Finally, a concept is defined as "a semantic entity that cannot be described as a generalization or abstraction of a specific object, event, time, place, or state" [2]. Concepts correspond to words such as "democracy" and "commerce". The MPEG-7 standard defines a rich set of relations among semantic entities. Each relation consists of pairs of semantic entities, and an optional degree of correlation (fuzziness). Of the relations defined by the standard, only two are relevant in the video analysis process. The first one, Specialization, is a taxonomy relation, and, as such it enables the system to find the classes of entities that the extracted entities belong to. The second one, Similar, contains the degree to which two semantic entities are expected to have similar low-level features.

The MPEG-7 standard has been based on XML Schema in order to define its semantic data structures that describe a multimedia document. In order to define domain – specific knowledge, recent developments tend to use RDF – based languages, such as RDF Schema and DAML+OIL, in the framework of ontological representation and the Semantic Web. The basic structures of an RDF Schema language are the classes, which describe groups of similar objects, and the properties, which describe characteristics of the classes. The description languages of ontologies are, for the purpose of knowledge representation, more powerful and expressive than XML Schema, which is concerned with structure rather than semantics.

In the following, visual descriptors, which are used to model visual content associated with semantic entities are categorized according to the MPEG-7 framework and are briefly described.

The *dominant color* descriptor specifies a set of dominant colors in any arbitrary shaped region. The extraction algorithm takes as an input a set of RGB color value and quantizes the image color vectors based on the Generalized Lloyd Algorithm (GLA). The dominant colors are extracted as a result of successive divisions of the color clusters with the GLA algorithm in between each division and then merging of the color clusters.

The *scalable color* descriptor is a color histogram in HSV color space, which is encoded by a Haar transform. Its binary representation is scalable in terms of bin numbers and bit representation accuracy over a broad range of data rates. The feature extraction consists of histogram extraction, uniformly quantized into 256 bins; histogram values are then nonlinearly quantized and finally the 4-bit values undergo a Haar transform.

The *color layout* descriptor specifies the spatial distribution of colors for high-speed retrieval and browsing. The descriptor is extracted from the 8x8 array using average colors as local dominant colors.

Color structure expresses local color structure in an image by the use of a structuring element that is comprised of several image samples. The color-structure descriptor is computed by block-based image retrieving colors of all pixels contained in the block element and extracting the descriptor-bins corresponding to each color.

Texture represents the regularity of an image such as directionality, coarseness, regularity of patterns etc. Three texture descriptors are defined.

The *homogeneous texture* descriptor characterizes texture through the distribution of its energy in a set of frequency channels. The first and the second components of the descriptor are extracted by computing the mean and standard deviation of the image pixel intensities. The remaining energy and energy deviation features are computed by applying a set of 30 Gabor filters (6 orientations and 5 scales) in the frequency domain. A fast and efficient extraction method is to work with the projections of the image followed by interpolation in the frequency domain.

On the other hand, the *texture browsing* descriptor relates to a perceptual characterization of texture, similar to a human characterization, in terms of regularity, coarseness and directionality. We refer to this descriptor as the TBC, which stands for Texture Browsing Component. This descriptor is extracted with multiresolution decomposition by using Gabor filtering with a similar frequency layout as in the homogeneous texture descriptor. The only difference in the layout is that here we use 4 scales instead of 5. From the multiresolution decomposition, a given image is decomposed into a set of filtered images. Each of these images represents the image information at a certain scale and at a certain orientation. The TBC captures the regularity (or the lack of it) in the texture pattern.

The *edge histogram* descriptor represents the spatial distribution of five types of edges, namely two orthogonal edges, two diagonal edges and one non-directional edge for 16 non-overlapping local regions in the image, which is called *sub-image*. This edge distribution for all the *sub-images* represents the *local edge histogram* using 16*5=80 histogram bins,

additional edge distribution information can be used from the whole image space and some horizontal and vertical semi-global edge distributions as well as local ones. The *global edge histogram* represents the edge distribution for the whole image space using 5 histogram bins. Similarly the *semi-global edge histogram* represents the edge distribution for some sub-sets of the sub-images space. Thus, total edge histogram contains local, global and semi-global histograms.
Contour shape is a descriptor that supports contour shape similarity matching and filtering based on contour curvature. This descriptor is based on the Curvature Scale Space (CCS) representation of the contour. The descriptor is extracted from the list of the contour points, which are resampled to equidistance points and then repeatedly low-pass filtered by performing a convolution with the normative (0.25,0.5,0.25) kernel. As the curvature of the contour evolves during curve low-pass filtering, the minima and maxima of the curvature are determined by finding the curvature zero crossings. Zero-crossings and corresponding number of passes of the filter are depicted in the form of a CSS image. The CSS descriptor is extracted by rescaling the numbers of passes of the filter to lie in the range [0.0, 1.0], thresholding zero-crossings, and then linearly quantizing both values.

RECOGNITION OF SEMANTIC ENTITIES FROM VISUAL CONTENT

In this paper, semantic indexing focuses on moving objects. Extraction of moving objects is investigated in this section, in terms of main mobile regions and object contours (in the MPEG-7 sense), focusing on bounding polygon extraction and object contour extraction. The basic scheme is shown below.
Our method extracts the main mobile object bounding polygons in video sequences, in the general case of sequences acquired by a moving camera [11],[14]. In our work, these bounding polygons define the regions in each frame, described as "regions-of-interest". Each one of these polygons entirely includes only one mobile object and thus can be used as initial approximation of its shape and position for its contour estimation and tracking.
In order to improve the extraction of the moving objects with the usage of the bounding polygons, as well as to eliminate the appearing noise, a multiresolution approach is also proposed. The main advantage of this approach is that it retains a low computational cost, given that all the operations are performed on the edge maps and not the images themselves.
The algorithm is based on background updating, where the slow insertion of updated areas of the observed scene into a reference background image is interpreted as a newly detected static feature of the scene; otherwise any observed alterations are considered either as noise or as a moving object. The main advantage of this algorithm is the ability to reject changes that occur due to the presence of excessive temporal clutter. In the following the specific steps of the algorithm are briefly described.
When a mobile camera is utilized, we first obtain the frames' edge-maps and we extract a reference background image. In order to achieve that, two criteria are used: (a) a frame counter that has high values at static pixels and lower values at transient ones and (b) a gradient change counter that indicates at which pixels the image gradient's direction significantly changes and thus these pixels are considered as transient. By merging these two counters into one decision module, and thresholding it appropriately (adaptation time), we obtain a binary reference background image. The inverse of this image is combined with the current frame edge map, through the logical operator *AND*, and results in a binary one containing the transient pixels (or edges).
The next step is to cluster the extracted transient pixels, if more than one mobile object exist in the sequence. In this module a block-matching motion estimation scheme is utilized, only for the pixels that have been considered as transient. To improve the performance in terms of complexity, we choose to count the edge pixels included in each block to decide for the best matching block, instead of computing e.g. the mean absolute difference. The motion vectors are then clustered, using a fuzzy c-means scheme.
In the final step, the bounding polygon of each main mobile object is extracted. The basic idea is to estimate the minimum polygon for every point set, since more than one polygon may contain a main mobile object. The Graham Scan method is utilized for that purpose, due to its low computational cost.
In order to obtain more accurate bounding polygons closer to the main moving objects' contour and without any significant noise, the above approach is implemented in different image resolutions. For each one of them, one bounding polygon for each main moving object is estimated. In the cases of low resolution, the background's undesirable details are eliminated, but also worse bounding polygons are obtained. On the other hand, in high-resolution cases, the bounding polygons are estimated more accurately, but also other undesirable pixels appear. By combining the results of all resolutions (in our experiments two resolution levels proved to be sufficient), we can obtain accurate polygons, close to main mobile objects' contours without any significant noise.
It is noted that the proposed method for mobile object localization may fail when: (a) the examined sequence is acquired by a camera with strong rotational motion, (b) the main mobile objects are relatively large (occupy a large area of each frame) and thus the estimation of the background dominant motion cannot be accurate, (c) the desirable objects move close to the camera, so that the assumption of the scene's orthographic projection on the image plane is not valid (slight rotation of the camera cannot be approximated by a simple translation) and (d) more than one objects move relatively

close to each other with similar velocities and thus the clustering of the transient pixels motion vectors fails to distinguish the different mobile objects. Figure 2: shows three successive frames (first row) of an outdoor cluttered sequence, acquired by a slightly rotating hand-held camera. The second row shows the respective edge maps and the third row illustrates the transient pixels/edges, along with the extracted bounding polygons.

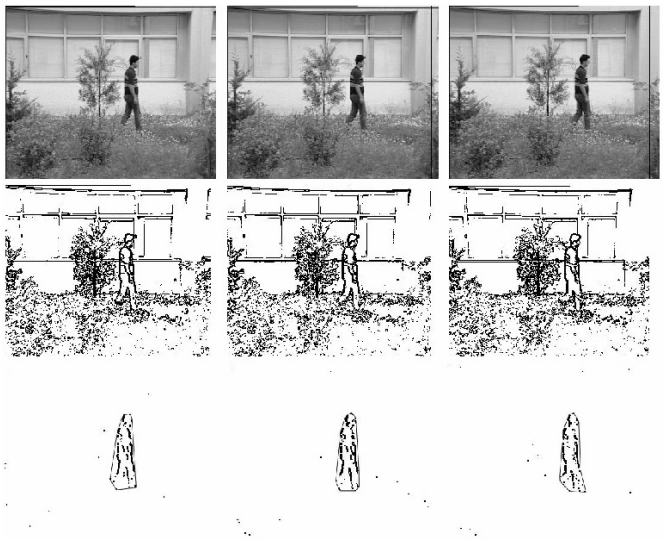


Figure 2: Moving object's bounding polygon extraction

OBJECT CONTOUR EXTRACTION

In the present work we utilize the method described above to obtain approximations for the shape and the position of the mobile objects of a sequence [18]. These initial results are used as input in a contour estimation method, in order to extract the objects' accurate contours; contour estimation can be achieved with an active contour model [19,22]. But since the object's bounding polygons extraction is nearly real time (depending on the implementation, the programming language used and/or the system's technical characteristics), the need for a faster but equally or more accurate method emerged.

The proposed method is a contour estimation approach [19] that requires an initial approximation of the position and the size of the objects. This condition would be a drawback, if this initialization were crucial for the results' accuracy, like the case of most snake models [22]; but as is explained below, the initialization required has to do only with the method' s implementation efficiency, since we focus on the problem of real time object localization. This method could actually be described as a snake model since, apart from the initialization issue, it evolves a curve to 'catch' only one object in an image. On the other hand, it does not include any energy minimization, but it seems more like a force–driven model, like the Level–Set ones. This model was developed for the problem of object tracking; therefore it is adapted to the needs of the specific problem and that is the reason why it cannot be applied in other problems, like image segmentation, straightforwardly.

Regarding the curve to be evolved, a force applied to each point of it is defined. Despite the fact that this force is quite easy to implement, it cannot avoid any possible overlapping of parts of the curve resulting in self-intersecting curves. Therefore we propose another module that automatically eliminates possible overlapping, without any significant loss of information.

Another idea of the proposed model is to define an adequate stopping criterion [21] for the curve evolution, which performs efficiently even in the worst cases, such as noisy images or weak edges of interest (boundaries of objects that do not fairly differ from the background due to local image smoothness) etc. Also, this criterion must not depend on any parameter tuning, so that it can be applied to any image straightforwardly. Compared with the respective criteria proposed in literature, this should overcome the implementation's problems that exist in most natural sequences due to temporal clutter.

RECOGNITION OF EXTRACTED MOVING REGIONS

Once the moving regions are extracted, their visual descriptors are computed, according to the relative section above. Afterwards, the descriptors are matched to the ones corresponding to the semantic entities. Each descriptor matching is

performed through a matching function and yields a matching value. A variety of matching criteria, including statistical matching, neural and neurofuzzy, graph matching criteria can be used for this purpose, and form an active field of research [4]. Moreover, the MPEG-7 standard provides standardised matching functions for the preselected visual descriptors [3]. These matching functions are briefly presented below.

The similarity between two *dominant color* descriptors, F_1 and F_2, can be measured by the following L2 distance function $D(F_1, F_2)$:

$$D^2(F_1, F_2) = \sum_{i=1}^{N_1} p_{1i}^2 + \sum_{j=1}^{N_2} p_{2j}^2 - \sum_{i=1}^{N_1}\sum_{j=1}^{N_2} 2a_{1i,2j} p_{1i} p_{2j}, \quad (1)$$

where F is the dominant color, , c and p are the corresponding color and percentage value, respectively. N is the total number of dominant colors, and $a_{k,l}$ is the similarity coefficient between two colors c_k and c_l, as shown in the following formula.

$$a_{k,l} = \begin{cases} 1 - d_{k,l} / d_{\max} & d_{k,l} \le T_d \\ 0 & d_{k,l} > T_d \end{cases}, \quad (2)$$

where $d_{k,l}$ is the Euclidean distance $d_{k,l} = \|c_k - c_l\|$, between two colors c_k and c_l, T_d is the maximum distance for two colors to be considered similar, and $d_{max} = \alpha T_d$. An appropriate value for T_d is between 10-20 in the CIE LUV color space and for α between 1.0-1.5.

The reconstruction of color histogram from Haar coefficients allows matching with highest retrieval efficiency. It is recommended to use the L1 norm for scalable color matching in the coefficient domain. In the case where only the coefficient signs are retained, however, the recommended matching measure is the Hamming distance, because of its very low complexity. The Hamming distance is computed by comparing two 63-bit descriptors and by finding the number of bit positions at which the binary bits are different. One way of implementing the above is to compute the XOR of the two descriptors to be compared and computing the number of '1' bits in the resulting bitstream.

The distance between two color layout descriptors values [Y,Cb,Cr] and [Y',Cb',Cr'] can be calculated as follows.

$$\begin{aligned} D = & \sqrt{\sum_{i=0}^{Max\{NumberOfYCoeff\}-1} \lambda_{Yi}(Y[i] - Y'[i])^2} \\ & + \sqrt{\sum_{i=0}^{Max\{NumberOfCCoeff\}-1} \lambda_{Cbi}(Cb[i] - Cb'[i])^2} \\ & + \sqrt{\sum_{i=0}^{Max\{NumberOfCCoeff\}-1} \lambda_{Cri}(Crf[i] - Cr'[i])^2} \end{aligned}, \quad (3)$$

where lamdas denote weighting values for each coefficient and they should be decreased according to the zigzag-scanline order.

The color structure matching procedure determines the similarity of two visual items by computing of the L_1 distance between their descriptors, as shown in the following formula.

$$\mathrm{dist}(A, B) = \sum_i \left|\mathbf{h}_A(i) - \mathbf{h}_B(i)\right|, \quad (4)$$

where $\mathbf{h}_A$ and $\mathbf{h}_B$ are the descriptor vectors of visual items A and B.

The similarity distance between two texture images is measured by summing the weighted absolute difference between two sets of homogeneous texture vectors, with an object (TD_{obj}), and the semantic entity $(TD_{SemEntity})$.

$$d(TD_{obj}, TD_{SemEntity}) = \sum_k \left| \frac{TD_{obj}(k) - TD_{SemEntity}(k)}{\alpha(k)} \right|, \quad (5)$$

The recommended normalization value $\alpha(k)$ is the standard deviation of $TD_{SemEntity}(k)$ for a given database.

The TBC vector , as texture browsing descriptor, captures the regularity v_1, direction v_2 and v_4 , and scale v_3 and v_5 in the texture pattern. Browsing using the TBC descriptor overcomes a typical retrieval performance, as we can select any of the components and browse along that dimension by matching the L1 distances between two sets of corresponding coefficients of TBC vector, which is shown in following formula.

$$TBC = \begin{bmatrix} v_1 & v_2 & v_3 & v_4 & v_5 \end{bmatrix} \tag{6}$$

Edge histogram similarity *E* is measured by the L1 distances between two sets of inverse quantized edge histograms A and B. A weighting factor 5 is applied to Global Edge (*GE*) histogram, since its number of bins is relatively smaller than that of Local Edge (*LE*) and Semi-Global Edge (*SGE*) histogram.

$$E = \sum_{i=0}^{79} \left| LE_A[i] - LE_B[i] \right| + 5 \times \sum_{i=0}^{4} \left| GE_A[i] - GE_B[i] \right| + \sum_{i=0}^{k} \left| SGE_A[i] - SGE_B[i] \right|, \tag{7}$$

Contour shape similarity measure *M* is computed as a weighted sum of the similarity measure between the global curve parameters and the similarity measure between the CSS peaks associated with the object and the semantic entity.

$$M = 0.4 \times E + 0.3 \times C + Mcss, \tag{8}$$

where *E* and *C* are the absolute values of *Eccentricity* and *Circularity* for the object and the semantic entity, and they are thresholded by the values 0.6 and 1.0 respectively. *Mcss* is the L2 measure similarity value between the CSS matching peaks with an additional penalty for each unmatched peak equivalent to the missing peak height. Two peaks are considered as matched if the L2 distance between their x-coordinates is below the threshold taken to be 0.1.

$$Mcss = \sum_1 \left((xpeak[i] - xpeak[j])^2 + (ypeak[i] - ypeak[j])^2 \right) + \sum_2 (ypeak[i])^2, \tag{9}$$

where Σ_1 is summation over all matched peaks (i and j are indices of object and visual entity peaks that match), and Σ_2 is summation over all unmatched object and visual entity peaks.
The multiple matching values, corresponding to the multiple descriptors are combined to compute a single matching value, which is considered to be the degree of confidence for recognition of the semantic entity. The overall matching value is computed via a weighted linear combination of the individual values. When no weights (degrees of importance) are used for the descriptors, then the single matching value is simply the mean matching value.

SIMULATION EXAMPLE

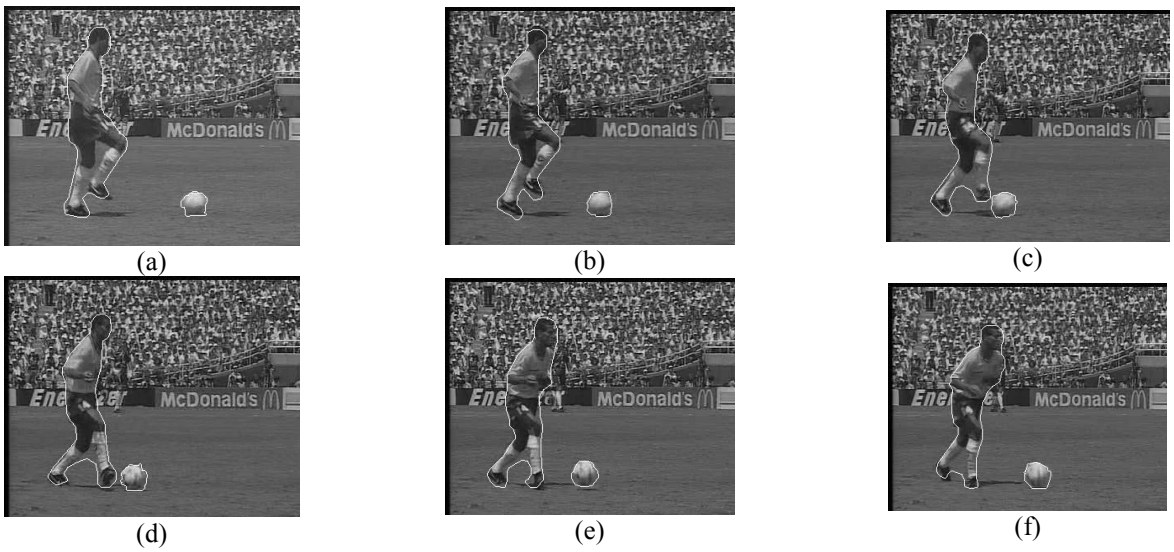
In our simulation we use a soccer sequence of images. Figure 3: shows the two moving regions extracted, i.e. a soccer player and a ball. The visual descriptors of the ball were extracted and matched to the ones of our model ball. The matching values and the overall matching value are shown in Table I:.

CONCLUSIONS AND FUTURE WORK

We showed, in this paper, how a hybrid knowledge base, combined with an efficient moving object extraction algorithm can be used to assist video analysis. Possible extensions of this work would be the exploration of the relation between modelling of semantic entities and description of video segments, usage of advanced, non – MPEG descriptors and detection of composite object through analysing spatiotemporal relations among moving regions.

Descriptor	Matching value	Weight	
Dominant color	0,91	0,5	
Scalable color	0,88	0,6	$d = 0.93$
Homogenous texture	0,85	0,1	
CSS	0,96	0,2	

Table I:. Matching of a soccer ball



(a) (b) (c)

(d) (e) (f)

Figure 3: Application of the proposed approach in a soccer sequence

REFERENCES

[1] Akrivas Giorgos, Wallace Manolis, Andreou Giorgos, Stamou Giorgos and Kollias Stefanos, "Context – Sensitive Semantic Query Expansion", IEEE International Conference Artificial Intelligence Systems AIS-02 (to appear).

[2] ISO/IEC JTC 1/SC 29 M4242, Text of 15938-5 FDIS Information Technology -- Multimedia Content Description Interface -- Part 5 Multimedia Description Schemes , October 2001.

[3] ISO/IEC JTC1/SC29/WG11, Text of ISO/IEC 15938-3/FCD Information Technology – Multimedia Content Description Interface – Part 3 Visual, October 2001.

[4] FAETHON, Unified Intelligent Access to Heterogeneous Audiovisual Content, EU, IST program, http://manolito.image.ece.ntua.gr/faethon/,

[5] X G. Votsis, A. Drosopoulos, G. Akrivas, V. Tzouvaras and Y.Xirouhakis, An MPEG-7 Compliant Integrated System for Video Archiving, Characterization and Retrieval, IASTED International Conference on Signal and Image Processing (SIP2000), Las Vegas, November 2000

[6] M. Wallace and G. Stamou, Towards a Context Aware Mining of User Interests for Consumption of Multimedia Documents}, International Conference on Multimedia and Expo, 2002 (to appear)

[7] A. Yoshitaka, T. Ichikawa, A Survey on Content-Based Retrieval for Multimedia Databases, IEEE Transactions on Knowledge and Data Engineering, 11(1), 1999, pp. 81-93

[8] W. A-Khatib, Y. F. Day, A. Ghafoor, P. B. Berra, Semantic Modeling and Knowledge Representation in Multimedia Databases, IEEE Transactions on Knowledge and Data Engineering, 11(1), 1999, pp. 64-80

[9] Horace H.S., Dinggang S., "An Affine – Invariant Active Contour Model (AI – Snake) for Model – Based Segmentation," *Image and Vision Computing*, vol. 16, pp. 135 – 146, 1998.

[10] Paragios N., Deriche R., "Geodesic Active Contours and Level Sets for the Detection and Tracking of Moving Objects," *IEEE Trans. on PAMI*, vol.22, no. 3, pp. 266 – 280, 2000.

[11] Y. Xirouhakis, G. Tsechpenakis and A. Delopoulos, "Fast Mobile Object Detection and Localization in Video Sequences," *Computer Vision and Image Understanding*, submitted in March 2002. http://www.image.ece.ntua.gr/~gtsech

[12] G. Tsechpenakis, Y. Xirouhakis and A. Delopoulos, "A Multiresolution Approach for Main Mobile Object Localization in Video Sequences," *In Proc. International Workshop on Very Low Bitrate Video Coding (VLBV01)*, Athens Greece, 2001. http://www.image.ece.ntua.gr/vlbv01/proceedings

[13] G. Tsechpenakis, Y. Avrithis and S. Kollias, "Efficient Moving Object Detection and Tracking in Video Sequences," *Image Vision and Computing*, submitted. http://www.image.ece.ntua.gr/~gtsech

[14] A. Makarov, Comparison of Background Extraction Based Intrusion Detection Algorithms, *In Proc. International Conference on Image Processing, Lausanne, Switzerland, 1996*, pp. 521-524.

[15] G. Tsechpenakis, N. Tsapatsoulis and S. Kollias, "Probabilistic Boundary-Based Contour Tracking with Snakes in Natural Cluttered Video Sequences," *International Journal of Image and Graphics (IJIG): Special Issue*, submitted. http://www.image.ece.ntua.gr/~gtsech