# Video Object Segmentation Using Bayes-Based Temporal Tracking and Trajectory-Based Region Merging

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

***Abstract*—A novel unsupervised video object segmentation algorithm is presented, aiming to segment a video sequence to objects: spatiotemporal regions representing a meaningful part of the sequence. The proposed algorithm consists of three stages: initial segmentation of the first frame using color, motion, and position information, based on a variant of the K-Means-with-connectivity-constraint algorithm; a temporal tracking algorithm, using a Bayes classifier and rule-based processing to reassign changed pixels to existing regions and to efficiently handle the introduction of new regions; and a trajectory-based region merging procedure that employs the long-term trajectory of regions, rather than the motion at the frame level, so as to group them to objects with different motion. As shown by experimental evaluation, this scheme can efficiently segment video sequences with fast moving or newly appearing objects. A comparison with other methods shows segmentation results corresponding more accurately to the real objects appearing on the image sequence.**

***Index Terms*—Image sequence analysis, temporal tracking, trajectory-based merging, video segmentation.**

## I. Introduction

DIGITAL video is an integral part of many newly emerging multimedia applications. New image and video standards, such as MPEG-4 and MPEG-7, do not concentrate only on efficient compression methods but also on providing better ways to represent, integrate, and exchange visual information [1]–[3]. These efforts aim to provide the user with greater flexibility for "content-based" access and manipulation of multimedia data. Many multimedia applications benefit from this content-based approach, including efficient coding of regions of interest in digital video, personalized user-interactive services, and sophisticated query and retrieval from image and video databases. These issues and objectives are currently addressed within the framework of the MPEG-4 and MPEG-7 standards [4], [5].

Manuscript received April 29, 2002; revised March 27, 2003. This work was supported in part by the EU Projects SCHEMA "Network of Excellence in Content-Based Semantic Scene Analysis and Information Retrieval" (IST-2001-32795), in part by the ATTEST "Advanced Three-Dimensional Television System Technologies" (IST-2001-34396), and in part by COST211 quat. This paper was recommended by Associate Editor L. Onural.

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

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

Digital Object Identifier 10.1109/TCSVT.2004.828341

In order to obtain a content-based representation, an input video sequence must first be segmented into an appropriate set of arbitrarily shaped objects, termed the video object planes (VOPs) in the MPEG-4 Verification Model, with each object possibly representing a particular meaningful content of the video stream [6]. This process may be required to be unsupervised, depending on the targeted application (e.g., indexing large video archives). The features of each object such as shape, motion and color information can subsequently be coded into the so-called Video Object Layer for transmission or storage or they can be used for efficient indexing and retrieval. Although the standards will provide the needed functionalities in order to compose, manipulate, and transmit the "object-based" information, the production of these objects is out of the scope of the standards and is left to the content developer. Thus, the success of any object-based approach depends largely on the accurate segmentation of the scene based on its contents.

Several approaches have been proposed for video segmentation, both supervised and unsupervised. The former require human interaction for defining the number of objects present in the sequence [7] or more often for grouping homogeneous regions to semantic objects [8]–[10], while the latter require no such interaction. Some segmentation approaches rely on segmenting each frame independently, focusing either on estimating discontinuities in the decision space [11] or on classifying pixels into regions based on their homogeneity with respect to the feature space [12]–[15]. Spatiotemporal objects are subsequently formed by associating the already formed spatial regions using their low-level features. A different approach is to use motion information to perform motion projection, i.e., to estimate the position of a region at a future frame, based on its current position and its estimated motion features [16], [17], [7]. Alternatively, one could restrict the problem of video segmentation to the detection of moving objects, using primarily motion information [18], [19], or by performing in parallel segmentation using other decision spaces as well (e.g., intensity information) and employing rule-based processing to enhance the motion segmentation result [20]. The moving-object detection approaches suffer from their inability to handle objects that halt at some time and resume moving at a later time; the halted objects are treated as background and as soon as they start moving again, they are detected as newly-appearing objects.

In this paper, a homogeneity-based approach is adopted to address the problem of unsupervised spatiotemporal segmentation. The proposed spatiotemporal segmentation method is

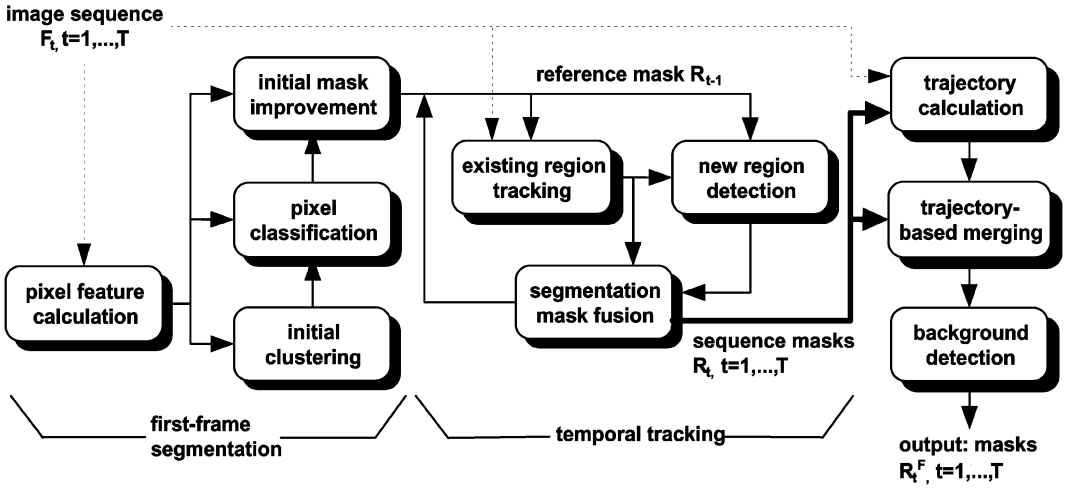



Fig. 1. Spatiotemporal segmentation algorithm overview.

based on initially applying an efficient two-dimensional (2-D) segmentation algorithm to the first frame of the image sequence, to produce an initial segmentation mask comprising regions which are homogeneous both in color and in motion. Following that, the formed regions are tracked in the subsequent frames, using a Bayes classifier and rule-based processing. Newly introduced regions are also detected and subsequently tracked. Several variations of this general architecture can be found in the literature [16], [17]. In contrast to those methods, however, the proposed algorithm does not group the formed regions to semantic objects, based on their motion, at the frame level. This allows for the temporal tracking to be performed at the homogeneous region level rather than the semantic object level, using the Bayes classifier for minimum mean-square error. In this way, generic objects can be easily tracked, by tracking their constituent homogeneous regions [17]. Motion information is used at the sequence level to group regions to semantic objects based on their long-term trajectories, as opposed to [16], [17], and [20]. Thus, the proposed system can efficiently handle objects that are moving in some frames and are halted in other frames, as opposed to systems performing motion-based merging of regions at the frame level. An overview of the proposed spatiotemporal segmentation algorithm is presented in Fig. 1.

The general idea of rule-based processing ("combining regions by a set of rules") was originally presented in [20], where segmentations of the same frame in different feature spaces were combined to create a more accurate final segmentation. In our method, rule-based processing is used for combining different segmentations of a frame, created by various probabilistic criteria being applied to the color features only. Motion information is used in our method at the sequence level, as already mentioned.

The paper is organized as follows. The segmentation algorithm that uses spatial, color, and motion features to produce an initial segmentation mask for the first frame of the sequence is briefly discussed in Section II. In Section III, the algorithm for tracking the already formed regions in the following frames and detecting the introduction of new regions is developed. The grouping in semantic objects of the spatiotemporal regions created by the tracking process is presented in Section IV. The method employed for determining which of the formed regions correspond to the background is also discussed in this section. Section V contains the results of an experimental evaluation and comparison of the developed methods. Finally, conclusions are drawn in Section VI.

## II. First Frame Segmentation

The segmentation algorithm employed for the segmentation of the first frame is based on a variant of the $K$-Means-with-connectivity-constraint algorithm (KMCC), a member of the popular $K$-Means family [21]. The KMCC algorithm is an algorithm that classifies the pixels into regions, taking into account not only the color information associated with each pixel but also the position of the pixel, thus producing connected regions rather than sets of chromatically similar pixels. In the past, KMCC has been successfully used for model-based image sequence coding [22] and content-based watermarking [23]. The variant used for first-frame segmentation introduces the use of motion features, in combination with the color and position features, using an appropriate pixel-region distance function.

The initial values required by the KMCC algorithm, including the initial number of regions, are estimated using an initial clustering procedure, based on breaking down the image to square blocks and assigning a color feature vector and a motion feature vector to each block [23]. The number of regions of the image is initially estimated by applying a variant of the *maximin* algorithm to this set of blocks. This is followed by the application of a simple $K$-Means algorithm to the blocks, $K$ being equal to the number of regions estimated by the maximin algorithm. Following a component labeling procedure, the spatial, color, and motion centers of the resulting components are calculated, to be used as input to KMCC. This automated initial clustering procedure makes unnecessary any user intervention. The resulting initial number of regions will be automatically adjusted during the execution of the KMCC algorithm.

The proposed segmentation algorithm consists of the following stages.

- Stage 1. Extraction of the color and motion feature vectors corresponding to each pixel $\mathbf{p} = [p_x p_y], p_x = 1, \ldots, p_{x,\max}, p_y = 1, \ldots, p_{y,\max}$, where $p_{x,\max}, p_{y,\max}$ are the frame dimensions. The three color components

of the CIE L*a*b* color space are used as color features, $\mathbf{I}_1(\mathbf{p}) = [I_{1,L}(\mathbf{p})I_{1,a}(\mathbf{p})I_{1,b}(\mathbf{p})]$, whereas motion feature vectors $\mathbf{V}_1(\mathbf{p})$ are calculated using a full search block matching algorithm (FSA). It has been shown that L*a*b* is more suitable for segmentation applications than the widely used RGB color space, since it provides perceptually uniform components [24].

- Stage 2. Estimation of the initial number of regions and their color, motion, and spatial centers, $\bar{\mathbf{I}}(s_k^1)$, $\bar{\mathbf{V}}(s_k^1), \bar{\mathbf{S}}(s_k^1)$, respectively, using the aforementioned initial clustering procedure. The superscript 1 denotes that $s_k^1$ is a spatial region at the first frame of the sequence (at time $t = 1$). Region centers are defined throughout this paper as the mean values of the corresponding pixel features for all pixels belonging to the region.
- Stage 3. Classification of the pixels to regions using the KMCC algorithm.
- Stage 4. Initial segmentation mask improvement by partial reclassification of pixels to the formed regions.

The distance function of a pixel $\mathbf{p}$ from a region $s_k^1$, used by the KMCC algorithm, is defined as follows:

$$D_{\mathrm{KMCC}}\left(\mathbf{p}, s_k^1\right) = \left\|\mathbf{I}_1(\mathbf{p}) - \bar{\mathbf{I}}\left(s_k^1\right)\right\| + \lambda_1 \left\|\mathbf{V}_1(\mathbf{p}) - \bar{\mathbf{V}}\left(s_k^1\right)\right\| + \lambda_2 \frac{\bar{A}}{A_k} \left\|\mathbf{p} - \bar{\mathbf{S}}\left(s_k^1\right)\right\| \quad (1)$$

where $\|\mathbf{I}_1(\mathbf{p}) - \bar{\mathbf{I}}(s_k^1)\|, \|\mathbf{V}_1(\mathbf{p}) - \bar{\mathbf{V}}(s_k^1)\|$ and $\|\mathbf{p} - \bar{\mathbf{S}}(s_k^1)\|$ are the Euclidian distances between the pixel intensity, motion, and spatial features and the region intensity, motion, and spatial centers respectively, $A_k$ is the area of region $s_k^1$ calculated in pixels, and $\bar{A}$ is the average area of all regions: $\bar{A} = E\{A_k\}$. The regularization parameters $\lambda_1, \lambda_2$ are defined as

$$\lambda_1 = 2 \cdot \frac{DI_{\max}}{\sqrt{(2u_{\max})^2 + (2v_{\max})^2}}$$

$$\lambda_2 = 0.1 \cdot \frac{DI_{\max}}{\sqrt{p_{x,\max}^2 + p_{y,\max}^2}}$$

where $DI_{\max}$ is an estimation of the image contrast and $u_{\max} = v_{\max}$ is the maximum allowed block displacements in both directions, used by the FSA.

The result of the application of the segmentation algorithm to the first frame is a segmentation mask $R_1$, i.e., a grayscale image comprising the spatial regions formed by the segmentation algorithm, $R_1 = \{s_1^1, s_2^1, \ldots, s_{K_1}^1\}$, in which different gray values $1, 2, \ldots, K_1$ correspond to different regions: $R_1(\mathbf{p} \in s_k) = k$. This mask is used for initiating the tracking process of Section III. An overview of the first-frame segmentation algorithm can be seen in a portion of Fig. 1. A detailed presentation of a variant of this algorithm can be found in [23].

## III. Region Temporal Tracking

### A. Temporal Tracking Overview

The temporal tracking module has two major functionalities:

- Tracking of existing regions, i.e., determining for each pixel of frame $F_t$ the region $s_k$ to which it belongs, given the mask $R_{t-1}$ and that region $s_k$ is present at that mask.

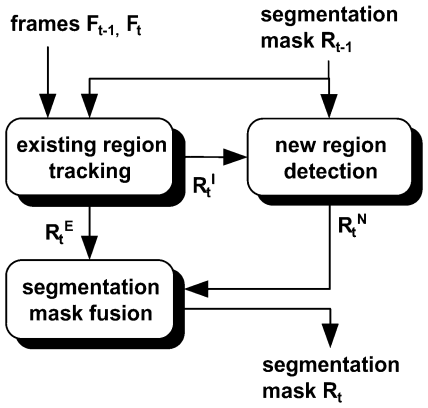



Fig. 2. Creation of mask $R_t$ by temporal tracking, using mask $R_{t-1}$.

- Detecting the introduction of new objects and forming the corresponding new regions.

These functionalities correspond to two distinct processing steps, which are coupled with a mask fusion procedure that properly combines the output of those steps to produce the segmentation mask for time $t, R_t$. Note that, as in the first frame segmentation, the CIE L*a*b* color space is used for tracking, due to its perceptual uniformity [24]; thus, before processing a frame $F_t$ in any way, its color components $I_{t,q}, q \in \{L, a, b\}$ must be calculated. Additionally, it should be noted that the image sequence $F_t, t = 1, \ldots, T$, where $T$ is the total number of frames to be processed, is assumed to belong to a single shot; this can be easily enforced by applying a scene-change detection method, as the one proposed in [25]. An overview of the proposed temporal tracking algorithm is shown in Fig. 2. A schematic example of the tracking process and the various masks estimated during this procedure can be seen in Fig. 3.

Before proceeding in discussing each step of the temporal tracking algorithm in detail, the notion of *spatiotemporal region* should be defined.

*Definition:* A spatiotemporal region, denoted $s_k$, is a set of temporally adjacent spatial regions $s_k^t, t = t_1, \ldots, t_2, t_1 < t_2$, all of which are nonempty ($s_k^t \neq \emptyset, \forall t \in [t_1, t_2]$) and for $t \in [t_1 + 1, t_2]$ have been created by temporal tracking of spatial region $s_k^{t_1}$, using the probabilistic framework described in Sections III-B–III-D

$$s_k = \left\{s_k^{t_1}, \ldots, s_k^{t_2}\right\}.$$

Any constituent spatial region $s_k^t$ of the spatiotemporal region $s_k$ can also be symbolized as $R_t \cap s_k$, where $R_t$ is the segmentation mask corresponding to frame $F_t$.

### B. Tracking Existing Regions

The segmentation mask produced for the first frame of the sequence is used for tracking the identified regions in the frame that follows. The tracking process begins by evaluating for each pixel $\mathbf{p}$ the color difference $\|\tilde{\mathbf{I}}_t(\mathbf{p}) - \tilde{\mathbf{I}}_{t-1}(\mathbf{p})\|$ between the current and the previous frame (2), where $\tilde{\mathbf{I}}_t(\mathbf{p}), \tilde{\mathbf{I}}_{t-1}(\mathbf{p})$ are the pixel color features after a simplification step, namely the application of a $3 \times 3$ moving average filter to the original pixel color features $\mathbf{I}_t(\mathbf{p}), \mathbf{I}_{t-1}(\mathbf{p})$. The filter is applied to each component of each frame independently. The resulting simplified

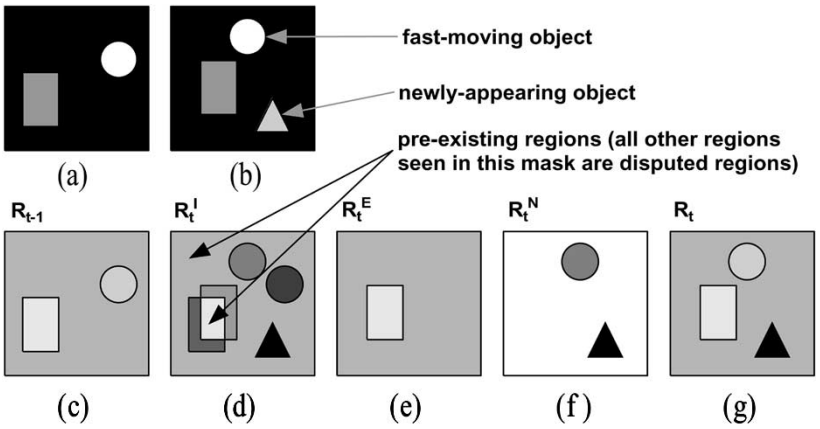



Fig. 3. Schematic example of the tracking process. (a) Frame $F_{t-1}$. (b) Frame $F_t$, containing one rapidly moving object (*circle*), for which there is no overlapping between the two frames, another moving object (*rectangle*), one newly appearing object (*triangle*), and one nonmoving object (*background*). (c) Segmentation mask $R_{t-1}$. (d) Mask $R_t^I$, containing parts of two preexisting regions (*rectangle, background*) and five disputed regions. (e) Mask $R_t^E$, containing two regions. (f) New-region mask $R_t^N$, containing two regions (*circle, triangle*). (g) Output of the mask fusion procedure: segmentation mask $R_t$. Note that the new region *circle* identified in mask $R_t^N$ has been associated with the *circle* object of mask $R_{t-1}$.

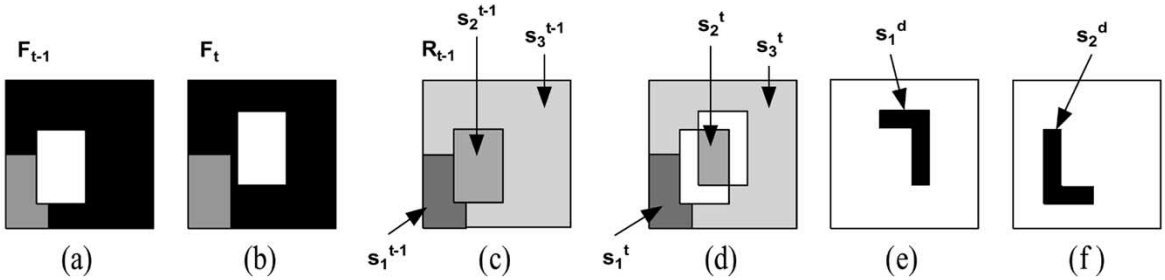



Fig. 4. Synthetic example of disputed regions and neighbor sets $g_j$: if (a) and (b) denote frames $F_{t-1}, F_t$ and (c) is the segmentation mask $R_{t-1}$, containing regions $s_1^{t-1}$ (leftmost rectangle), $s_2^{t-1}$ (other rectangle), and $s_3^{t-1}$ (background), then (d) shows the nondisputed pixels at time $t$ (disputed are painted white). (e) Disputed region $s_1^d$ in black, associated with neighbor set $g_1 = \{s_2^t, s_3^t\}$ [its neighbors can be seen in (d)]. (f) Disputed region $s_2^d$ in black, associated with neighbor set $g_2 = \{s_1^t, s_2^t, s_3^t\}$.

color components are used only in evaluating the aforementioned color difference and aim at the reduction of the number of isolated pixels marked as disputed as follows:

$$\begin{aligned}\mathrm{DIF}(\mathbf{p}) &= \|\tilde{\mathbf{I}}_t(\mathbf{p}) - \tilde{\mathbf{I}}_{t-1}(\mathbf{p})\| \\ &= \sqrt{\sum_{q=\{L,a,b\}} (\tilde{I}_{t,q}(\mathbf{p}) - \tilde{I}_{t-1,q}(\mathbf{p}))^2}. \end{aligned} \quad (2)$$

If this difference is below a reasonable threshold $\mathrm{DIF}_{\mathrm{th}} = 6$, pixel $\mathbf{p}$ is considered to belong to the same region it belonged to at time $t-1$, thus $R_t^I(\mathbf{p}) = R_{t-1}(\mathbf{p})$, where $R_t^I$ is intermediate segmentation mask. Otherwise, pixel $\mathbf{p}$ is marked as disputed: $R_t^I(\mathbf{p}) = K_{t-1} + R_{t-1}(\mathbf{p})$, where $K_{t-1}$ is the number of spatiotemporal regions identified until time $t-1$ ($K_{t-1} = \max_{\mathbf{p},\tau\in[1,t-1]}\{R_\tau(\mathbf{p})\}$; at time $t = 2, K_{t-1} \equiv K_1$ (see Section II); this may increase with time, as new regions are identified using the procedure described in Section III-C). In this way, up to $K_{t-1}$ disputed regions are formed in mask $R_t^I$; these are broken down to the minimum number $J_t$ of connected disputed regions $s_j^d, j = 1, \ldots, J_t$, using a four-connectivity component labeling algorithm [26]. Disputed regions $s_j^d$ are formed for the purpose of restricting the reclassification of disputed pixels to their neighboring nondisputed regions (Fig. 4); disputed regions need not be homogeneous in some way, as the nondisputed ones. Mask $R_t^I$ [Fig. 3(d)] at this intermediate level contains the nonchanged parts of the preexisting regions and the formed disputed regions and will be used for the detection of newly introduced regions discussed in the next subsection.

Following that, pixels belonging to disputed regions $s_j^d, j = 1, \ldots, J_t$ are classified to one of the nondisputed regions $s_k^t, k = 1, \ldots, K_{t-1}$ and $s_k^t \neq \emptyset$, to form the $R_t^E$ mask [Fig. 3(e)], as follows. For each disputed region $s_j^d, j = 1, \ldots, J_t$, the corresponding set $g_j$ of neighboring regions is identified; a synthetic example of this can be seen in Fig. 4. The reclassification of each disputed pixel to a nondisputed neighboring region $s_k^t$ is then performed using its color feature vector $\mathbf{I}_t(\mathbf{p})$ only, using a Bayes classifier.

According to the Bayes classifier for minimum classification error [27], a disputed pixel $\mathbf{p}$, $\mathbf{p} \in s_j^d$, is assigned to region $s_k^t$ if

$$p\left(s_k^t \,|\, \mathbf{I}_t(\mathbf{p})\right) > p\left(s_q^t \,|\, \mathbf{I}_t(\mathbf{p})\right), \quad \forall s_k^t, s_q^t \in g_j, k \neq q. \quad (3)$$

Using the Bayes Theorem [27], (3) can be rewritten as

$$p\left(\mathbf{I}_t(\mathbf{p}) \,|\, s_k^t\right) \cdot p\left(s_k^t\right) > p\left(\mathbf{I}_t(\mathbf{p}) \,|\, s_q^t\right) \cdot p\left(s_q^t\right) \quad \forall s_k^t, s_q^t \in g_j, k \neq q. \quad (4)$$

The probability $p(s_k^t)$ is the *a priori* probability of region $s_k^t$, whereas probability $p(\mathbf{I}_t(\mathbf{p}) \,|\, s_k^t)$ is the density function of the color features of pixels belonging to region $s_k^t$. The latter could be determined using the normalized histogram $\mathrm{hist}_k^t$ of each color component for the pixels of region $s_k^t$. Under the *constant*

*intensity assumption* [28], the histograms at time $t-1$, $\text{hist}_k^{t-1}$, can be used instead; these are chosen over the histograms at time $t$, to prevent the calculations from being influenced by histogram deterioration due to many pixels being currently marked as disputed. Thus, we have

$$p\left(\mathbf{I}_t(\mathbf{p}) \,|\, s_k^t\right) = p\left(\mathbf{I}_t(\mathbf{p}) \,|\, s_k^{t-1}\right) = \prod_{x\in\{L,a,b\}} \text{hist}_k^{t-1,x}(I_{t,x}(\mathbf{p})) \quad (5)$$

where $\text{hist}_k^{t-1,x} = (\text{Hist}_k^{t-1,x})/(M_k^{t-1}), x \in \{L,a,b\}$; $M_k^{t-1}$ is the number of pixels of region $s_k$ at time $t-1$ and $\text{Hist}_k^{t-1,x}, x \in \{L,a,b\}$ are its corresponding histograms.

Assuming that among the pixels of disputed region $s_j^d$ the *a priori* probability of region $s_k^t, s_k^t \in g_j$ is equal for all regions $s_k^t \in g_j$, the classification criterion of (4) is simplified to: disputed pixel $\mathbf{p}, \mathbf{p} \in s_j^d$, is assigned to region $s_k^t$ if

$$p\left(\mathbf{I}_t(\mathbf{p}) \,|\, s_k^t\right) > p(\mathbf{I}_t(\mathbf{p}) \,|\, s_q^t) \qquad \forall s_k^t, s_q^t \in g_j, k \neq q. \quad (6)$$

Since no connectivity constraint is enforced during the reclassification of the disputed pixels, the connectivity of the formed regions must be evaluated as soon as the reclassification is completed, using a four-connectivity component labeling algorithm [26]; through this process, any existing nonconnected parts of the formed regions and any regions smaller than $M_{\text{th,min}} = 0.1\%$ of the frame pixels (fragments) are detected and subsequently appended: a fragment $s_q$ is appended to region $s_k^t, k = 1,\ldots,K_{t-1}$, where $s_k^t \neq \emptyset$, for which the distance of their color centers

$$D_I\left(s_q, s_k^t\right) = \|\bar{\mathbf{I}}(s_q) - \bar{\mathbf{I}}\left(s_k^t\right)\| \quad (7)$$

is minimum.

The result of this intermediate procedure is a segmentation mask [$R_t^E$, Fig. 3(e)] containing all nondisappearing regions that existed in the previous frame. The detection of new objects and fast-moving ones is described in Section III-C.

### C. Detection of New Regions

New object detection is performed by creating a new mask $R_t^N$, containing possible new regions, starting from the intermediate mask $R_t^I$ (see Section III-B). Following rule-based processing, certain disputed regions $s_j^d$ contained in that mask are treated as possibly new regions and eventually these are either discarded or identified as valid new regions. The new-region mask $R_t^N$ and mask $R_t^E$ are then fused, as discussed in Section III-D.

For every nondisputed region $s_k^t, k = 1,\ldots,K_{t-1}$ and $s_k^t \neq \emptyset$, the following characteristic value is calculated:

$$P_k^t = E\left\{p\left(\mathbf{I}_t(\mathbf{p}) \,|\, s_k^t\right)\right\}, \qquad \mathbf{p} \in s_k^t.$$

The characteristic value $P_k^t$ is a measure of the homogeneity of region $s_k$ at time $t$, with respect to its color. For an ideally homogeneous region $s_k^t$, $P_k^t = 1$, whereas for a region lacking homogeneity $P_k^t$ would be close to zero. Since color homogeneity was a key criterion for the initial formation of regions, value $P_k^t$ is expected to be relatively high for all regions $s_k^t$.

A similar characteristic value is calculated for every disputed region $s_j^d, j = 1,\ldots,J_t$ as follows:

$$P_j^d = E\left\{\max_k\left\{p\left(\mathbf{I}_t(\mathbf{p}) \,|\, s_k^t\right), s_k^t \in g_j\right\}\right\}, \qquad \mathbf{p} \in s_j^d$$

where $g_j$ is the set of neighbors of disputed region $s_j^d$. Value $P_j^d$ indicates how the homogeneity of the regions $s_k^t$ neighboring $s_j^d$, i.e. $s_k^t \in g_j$, would be affected if pixels $\mathbf{p} \in s_j^d$ were reassigned to them. Such a low value indicates that the pixels of disputed region $s_j^d$ would compromise the homogeneity of any of its neighboring regions; this is a good indication that a new object has entered the scene, therefore a new region should be formed.

*Rule 1:* A disputed region $s_j^d, j = 1,\ldots,J_t$ is identified as a possible new region if $P_j^d \ll P_k^t, \forall s_k^t \in g_j$, strictly defined as

$$\frac{P_j^d}{\min_k\{P_k^t\}} < 0.05, \qquad \text{where } s_k^t \in g_j.$$

For each disputed region that is identified as a possible new region, the normalized histograms $\text{hist}_j^{t,x}, x \in \{L,a,b\}$ are calculated and the probability $p(\mathbf{I}_t(\mathbf{p}) \,|\, s_j^d)$ is defined as

$$p\left(\mathbf{I}_t(\mathbf{p}) \,|\, s_j^d\right) = \prod_{x\in\{L,a,b\}} \text{hist}_j^{t,x}(I_{t,x}(\mathbf{p})).$$

*Rule 2:* For every pixel $\mathbf{p}$ of possible new region $s_j^d$, mask $R_t^N$, initially $R_t^N(\mathbf{p}) = 0\ \forall \mathbf{p}$, is updated by

$$R_t^N(\mathbf{p}) = R_t^I(\mathbf{p}), \quad \text{if } p\left(\mathbf{I}_t(\mathbf{p}) \,|\, s_j^d\right) > p\left(\mathbf{I}_t(\mathbf{p}) \,|\, s_k^t\right) \quad \forall s_k^t \in g_j.$$

Following the processing of all disputed regions of mask $R_t^I$ according to rules 1 and 2, a component labeling algorithm is applied to mask $R_t^N$ and connected components that exceed a predefined size threshold $M_{\text{th}}$, defined as $M_{\text{th}} = 2 \cdot M_{\text{th,min}} = 0.2\%$ of the total number of pixels of a frame, are identified as valid new regions. These receive new labels $K_{t-1}+1,\ldots,K_{t-1}+L_t, L_t$ being the number of new regions identified at time $t$ and $K_{t-1}$ being the number of preexisting spatiotemporal regions.

### D. Segmentation Mask Fusion

As soon as new object mask $R_t^N$ [Fig. 3(f)] is formed, segmentation masks $R_t^N$ and $R_t^E$ [Fig. 3(e)] are fused to mask $R_t$ [Fig. 3(g)], according to the following equation:

$$R_t(\mathbf{p}) = \begin{cases} R_t^E(\mathbf{p}), & \text{if } R_t^N(\mathbf{p}) = 0 \\ R_t^N(\mathbf{p}), & \text{if } R_t^N(\mathbf{p}) \neq 0. \end{cases} \quad (8)$$

Following that, the merging of new regions with existing ones and the association of new regions with extinct ones are evaluated using rules 3 and 4.

*Rule 3:* A new region $s_q, s_q \cap R_{t-1} = \emptyset, s_q \cap R_t \neq \emptyset$, is appended to neighboring region $s_k, s_k \cap R_t \neq \emptyset$, if their color distance $D_I(s_q^t, s_k^t)$ [see (7)] is below a threshold $C_{\text{th}}$.

The above ensures that the color centers of any given new region are not particularly similar to those of an existing neighboring region; if this is the case, the new region is most likely to be part of that preexisting region, and therefore it is appended to it.

*Rule 4:* A new region $s_q, s_q \cap R_{t-1} = \emptyset, s_q \cap R_t \neq \emptyset$ is associated with extinct region $s_k, s_k \cap R_{t-1} \neq \emptyset, s_q \cap R_t = \emptyset$ if their color distance $D_I(s_q^t, s_k^{t-1})$ is below a threshold $C_{\text{th}}$ and their spatial distance $D_S(s_q^t, s_k^{t-1}) = ||\bar{\mathbf{S}}(s_q^t) - \bar{\mathbf{S}}(s_k^{t-1})||$ is below a threshold $S_{\text{th}}$. The distances are calculated using the color and spatial centers of regions $s_q, s_k$ at time $t$ and $t-1$, respectively. Following a successful association, pixels of region $s_q$ in $R_t$ receive the label of region $s_k$ and region $s_q$ ceases to exist.

Thresholds $C_{\text{th}}, S_{\text{th}}$ in the above rules require reasonable values (in our experiments, $C_{\text{th}} = 10, S_{\text{th}} = 0.2 \cdot \sqrt{p_{x,\max}^2 + p_{y,\max}^2}$); small deviations from these values should not influence the output of the algorithm, as experimentally demonstrated in Section V.

The association of newly appearing objects at time $t$ with objects that become extinct at the same time is important for the tracking of fast-moving objects that do not overlap if adjacent frames are superimposed. These initially become extinct and subsequently are associated with a newly appearing object, as described by rule 4. To avoid making the *smooth motion assumption* regarding region motion [29], which frequently fails (see the *ball* object in Fig. 11), rule 4 does not take into account the motion of the regions. The fact that not all regions have to be associated between adjacent frames (as in video segmentation schemes relying on segmenting each frame independently), but only the regions for which tracking is lost, makes this approach feasible. A demonstration of tracking a fast-moving object is included in the experimental results section of this paper.

To enforce region connectivity, the above rule-based processing is followed by the application of the four-connectivity component labeling algorithm to mask $R_t$ and the merging of any small nonconnected parts of the existing regions based on color similarity, as in Section III-B.

## IV. Trajectory-Based Region Merging and Background Detection

### A. Region Trajectory Calculation

The third and final module of the proposed segmentation algorithm aims at grouping the tracked regions to different semantic objects to produce the final segmentation masks $R_t^F, t = 1, \ldots, T$. This grouping will be performed using motion information, i.e., the trajectories of the spatiotemporal regions over the processed frames. It is therefore of importance to accurately calculate the trajectory of each tracked region.

Several motion estimation algorithms aimed at various applications have been proposed in the literature [30]–[32]; most of them belong to the family of block matching algorithms (BMAs) [33], [34]. In the proposed segmentation algorithm, an FSA is used. Although computationally intensive, the full search algorithm has the advantage of not relying on the assumption that the mean absolute difference (MAD) distortion function increases monotonically as the search location moves away from the global minimum, thus producing more accurate motion vectors than most fast block matching techniques. In addition to that, hardware implementation of the full search algorithm is much easier than that of other block matching algorithms due to its simplicity and regularity [35].

Following the calculation of motion vector $\mathbf{V}_t^b(\mathbf{b}) = [u_t(\mathbf{b})v_t(\mathbf{b})]$ for every block $\mathbf{b} = [b_x b_y]$, where $b_x = 1, \ldots, p_{x,\max}/w, b_y = 1, \ldots, p_{y,\max}/w$ and $w = 8$ is the dimension of the square blocks, the motion of region $s_k, k = 1, \ldots, K_T$, at time $t$ is estimated by the motion vectors of the blocks belonging to it, using a least-squares approximation. The bilinear motion model [36] is used for approximating the motion of each region, being less susceptible to noise than the commonly used affine model [37]. To accommodate for possibly erroneous block motion vectors, due to blocks overlapping moving object contours or due to extreme color uniformity in parts of the frame, an iterative rejection scheme is employed in estimating the parameters of the bilinear model [36]; specifically, for each frame, the region motion parameters are estimated and those blocks whose estimation error is higher than the average are rejected. Upon convergence, the parameters $a_0^t, \ldots, a_7^t$ that minimize the motion compensation error $E_{\text{all}}$ are estimated using the least-squares estimation method. Let $\mathbf{b}^i$, $i = 1, \ldots, N_k^t$, be the blocks belonging to region $s_k^t$. Then,

$$E_{\text{all}}^t(s_k) = \sum_{i=1}^{N_k^t} (u_t(\mathbf{b}^i) - \hat{u}_t(\mathbf{b}^i))^2 + (v_t(\mathbf{b}^i) - \hat{v}_t(\mathbf{b}^i))^2 \quad (9)$$

where $\mathbf{V}_t^b(\mathbf{b}^i) = [u_t(\mathbf{b}^i)v_t(\mathbf{b}^i)]$ is the block motion vector of block $\mathbf{b}^i$ belonging to region $s_k^t$, calculated by block matching, and

$$\begin{aligned} \hat{u}_t(\mathbf{b}^i) &= a_0^t + a_1^t b_x^i + a_2^t b_y^i + a_3^t b_x^i b_y^i \\ \hat{v}_t(\mathbf{b}^i) &= a_4^t + a_5^t b_x^i + a_6^t b_y^i + a_7^t b_x^i b_y^i. \end{aligned} \quad (10)$$

Variables $b_x^i, b_y^i$ are the spatial coordinates of block $\mathbf{b}^i$. This process is similar to that suggested in [36] for global motion estimation, with the difference that in our case it is applied to arbitrarily-shaped spatial regions rather than entire frames.

This process produces a region motion parameter vector $\mathbf{U}^t(s_k) = [a_0^t \ldots a_7^t]$; estimating $\mathbf{U}^t(s_k)$ for every $t$ provides a region trajectory matrix $\mathbf{U}(s_k)$, given as

$$\mathbf{U}(s_k) = [\mathbf{U}^1(s_k)\mathbf{U}^2(s_k) \ldots \mathbf{U}^{T-1}(s_k)]^T. \quad (11)$$

Since region $s_k$ is not necessarily present in all segmentation masks $R_t, t = 1, \ldots, T$, a function $\Upsilon_t(s_k)$ is used to monitor its presence as follows:

$$\Upsilon_t(s_k) = \begin{cases} 1, & \text{if } R_t \cap s_k \neq \emptyset \\ 0, & \text{if } R_t \cap s_k = \emptyset. \end{cases} \quad (12)$$

### B. Unsupervised Region Merging

The goal of the trajectory-based region merging module is to group the tracked homogeneous regions to semantic objects under the assumption that regions belonging to the same object have similar trajectories, as opposed to regions belonging to different objects. It is further assumed that any object at any given time should be a spatially connected component. To help enforce the latter, the notion of *spatiotemporal neighbors* is defined.

*Definition:* Two regions $s_m, s_n$, are *spatiotemporal neighbors* if they co-exist in at least one segmentation mask $R_t, \sum_{t=1}^T \Upsilon_t(s_m)\Upsilon_t(s_n) \neq 0$, and they are spatial neighbors

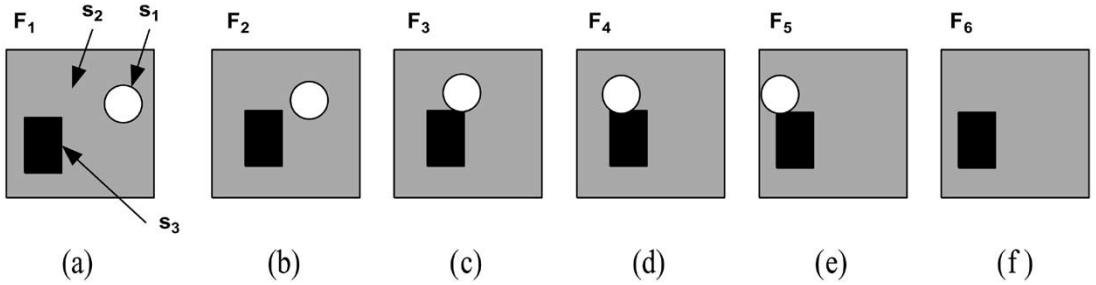



Fig. 5. Example of spatiotemporal neighbors: if the white, gray, and black regions in the above six frames denote spatiotemporal regions $s_1$, $s_2$, and $s_3$, respectively, then $s_2$ and $s_3$ are spatiotemporal neighbors; $s_1$ and $s_2$ are spatiotemporal neighbors as well, despite the fact that $s_1$ is not present in (f). Regions $s_1$ and $s_3$ are not spatiotemporal neighbors, because $s_1^t$ and $s_3^t$ are not spatial neighbors for $t = 1, 2, 6$ [(a), (b), and (f)], despite the fact that they are spatial neighbors for $t = 3, 4, 5$ [(c), (d), and (e), respectively].

in all segmentation masks that they co-exist in. A synthetic example of spatiotemporal neighbors is given in Fig. 5.

The motion similarity $D_U(s_m, s_n)$ of regions $s_m$ and $s_n$ that are spatiotemporal neighbors over the $T$ examined frames is defined by means of a per-frame motion distance

$$D_U(s_m, s_n) = \frac{\sum_{t=1}^{T-1} \Upsilon_t(s_m)\Upsilon_t(s_n) D_U^t(s_m, s_n)}{\max\left\{1, \sum_{t=1}^{T-1} \Upsilon_t(s_m)\Upsilon_t(s_n)\right\}} \quad (13)$$

where $D_U^t(s_m, s_n)$ is the motion similarity of the two regions at time $t$ (equivalently, the motion distance of the two spatial regions $s_m^t, s_n^t$), to be defined in the sequel.

The motion similarity of spatial regions is usually measured by the increment of mean-square motion compensation error [17]. Let $E_m^t(s_m), E_n^t(s_n)$ be the sum of square compensation errors for regions $s_m, s_n$ at time $t$, comprising $N_m^t, N_n^t$ blocks, respectively, and let $E_{m,n}^t(s_m), E_{m,n}^t(s_n)$ be the corresponding sum of square compensation errors using the motion model calculated for region $s_m \cup s_n$. Then, the increment of mean-square motion compensation error $\Delta^t(s_m, s_n)$ is defined as

$$\Delta^t(s_m, s_n) = \frac{E_{m,n}^t(s_m) + E_{m,n}^t(s_n) - E_m^t(s_m) - E_n^t(s_n)}{N_m^t + N_n^t}.$$

However, this similarity measure is not reliable when the sizes $N_m^t, N_n^t$ of the two examined regions are significantly different. In such cases, if, for example, $N_m^t \gg N_n^t$, the motion model parameters for region $s_m \cup s_n$ are approximately equal to those for region $s_m$, thus $E_{m,n}^t(s_m) \simeq E_m^t(s_m)$ and $\Delta^t(s_m, s_n) \simeq (E_{m,n}^t(s_n) - E_n^t(s_n))/(N_m^t + N_n^t)$. Even if the average increment of the mean-square motion compensation error for the pixels of region $s_n$, $(E_{m,n}^t(s_n) - E_n^t(s_n))/(N_n^t)$, is sufficiently high, indicating that the new motion model cannot adequately express its motion, $\Delta^t(s_m, s_n)$ can be sufficiently low for a merging to take place, since $N_m^t \gg N_n^t$. For the *table tennis* sequence [first frame in Fig. 6(a)], this is demonstrated in Fig. 7(a) and (c), where small parts of the hand are merged with the significantly larger background area, instead of merging with other parts of the hand. To alleviate this problem, a new similarity measure $D_U^t(s_m, s_n)$ is proposed, the *sum of mean-square error increments*, that demands both regions to be sufficiently well represented by their common motion model for a merging to take place:

$$D_U^t(s_m, s_n) = \frac{E_{m,n}^t(s_m) - E_m^t(s_m)}{N_m^t} + \frac{E_{m,n}^t(s_n) - E_n^t(s_n)}{N_n^t}.$$

The improvement achieved by adopting this region similarity measure is illustrated in Fig. 7.

The trajectory-based region merging stage begins by enforcing the following two rules to the spatiotemporal regions formed, in order to eliminate regions that are too thin or have a short temporal duration, and thus cannot qualify for representing a semantic object on their own.

*Rule 5:* If region $s_m$ has a temporal duration, calculated in the number of frames, shorter than $\tau_{\min} = 4, \sum_{t=1}^{T} \Upsilon_t(s_m) < \tau_{\min}$, then it is forced to merge with its spatiotemporal neighbor $s_n$ for which the distance $D_U(s_m, s_n)$ is minimized.

*Rule 6:* If region $s_m$ is particularly thin, $\sum_{t=1}^{T} (M_m^t)/(\sqrt{(bb_{m,x}^t)^2 + (bb_{m,y}^t)^2}) < l_{\min} \sum_{t=1}^{T} \Upsilon_t(s_m)$, where $l_{\min} = 6, M_m^t$ is the region size in pixels and $bb_{m,x}^t, bb_{m,y}^t$ are the dimensions of its bounding box at time $t$, then it is forced to merge with its spatiotemporal neighbor $s_n$ that satisfies the following conditions:

$$\sum_{t=1}^{T} \frac{M_n^t}{\sqrt{\left(bb_{n,x}^t\right)^2 + \left(bb_{n,y}^t\right)^2}} \geq l_{\min} \sum_{t=1}^{T} \Upsilon_t(s_n) \quad (14)$$

$$\frac{\sqrt{\left(bb_{n,x}^t\right)^2 + \left(bb_{n,y}^t\right)^2}}{\sum_{t=1}^{T} \Upsilon_t(s_n)} \geq \frac{\sqrt{\left(bb_{m,x}^t\right)^2 + \left(bb_{m,y}^t\right)^2}}{\sum_{t=1}^{T} \Upsilon_t(s_m)} \quad (15)$$

and for which the distance $D_U(s_m, s_n)$ is minimized. If no spatiotemporal neighbor $s_n$ that satisfies the above conditions exists, condition (14) is dropped, followed by condition (15) if necessary.

After the above two rules have been enforced, the remaining regions are merged in an agglomerative manner [15], starting from the region pairs for which the distance $D_U(s_m, s_n)$ is minimized. After each merging, the motion parameter matrix of the resulting region is updated. This process is similar in nature with the Recursive Shortest Spanning Tree segmentation method [14], where neighboring nodes are merged while considering the minimum of a cost function, and the cost value of the new node is recalculated after the merging.

The merging process is based on the assumption that, for the given region similarity measure, the desired mergings (e.g., mergings between spatiotemporal regions belonging to the same object) are characterized by a lower value of the region similarity measure, thus precede undesired mergings (those between differently moving objects). Therefore, to allow for unsupervised termination of this procedure, the point must be detected

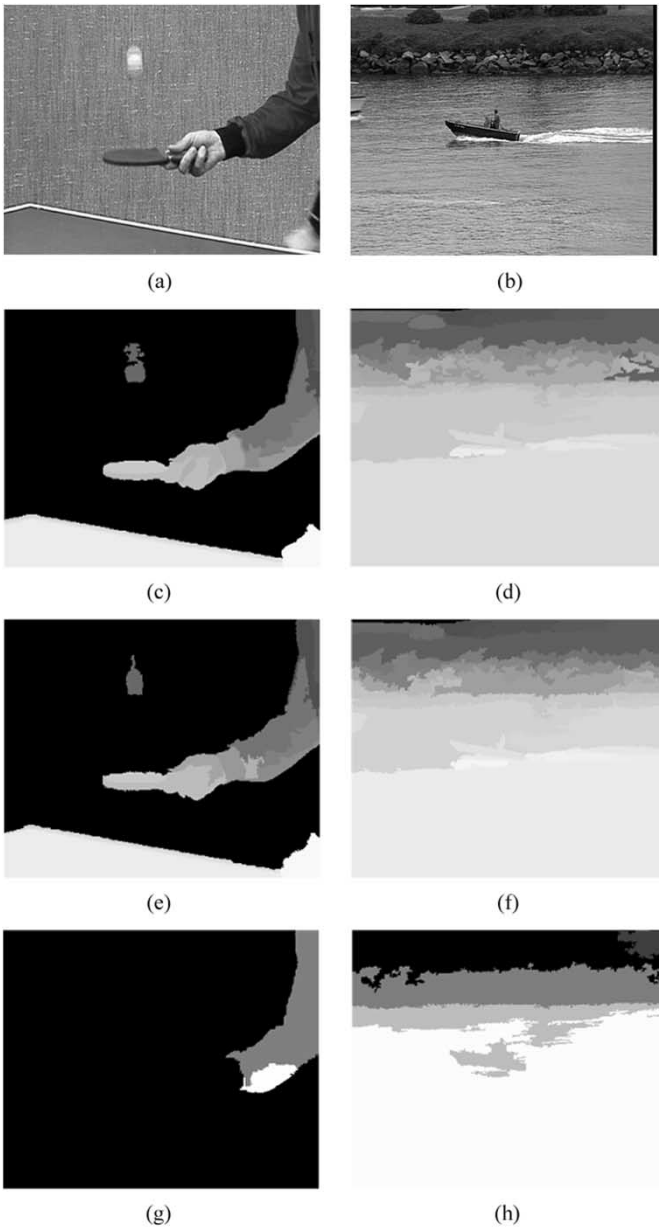



Fig. 6. First frame segmentation results for the CIF sequences (a) *table tennis* and (b) *coastguard*: sequence first frame. (c) and (d) Corresponding segmentation masks before the application of the Bayes-based enhancement stage. (e) and (f) Corresponding segmentation masks after the application of the enhancement stage. It is evident, particularly in the case of the tennis ball, that the Bayes-based enhancement stage is effective in improving the initial segmentation. (g) and (h) Results using the still-image segmentation method of [20]. Not using motion features and attempting to avoid over-segmentation resulted in small but important moving objects (racket, ball, and coastguard ship) being missed.

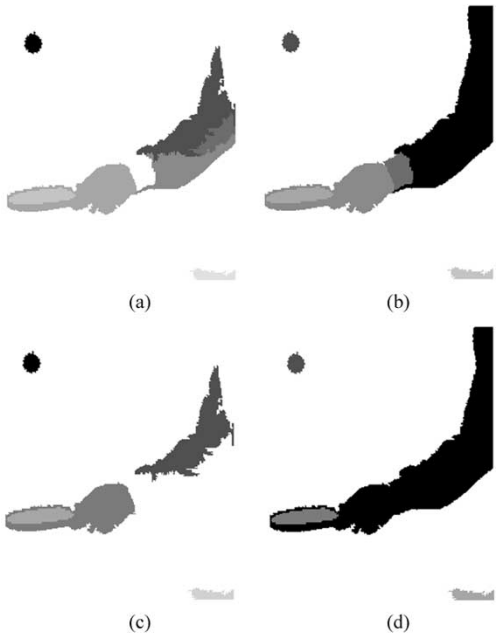



Fig. 7. Supervised segmentation results for the CIF sequence *table tennis*: (a) measuring region similarity by the increment of mean-square motion compensation error, region number set to 8. (b) Measuring region similarity by the sum of the mean-square error increments, for the same number of regions. (c) and (d) Region number set to 6, region similarity measured as in (a) and (b), respectively. It can be seen that the proposed region similarity measure [(b) and (d)] performs better than the traditional approach used in (a) and (c).

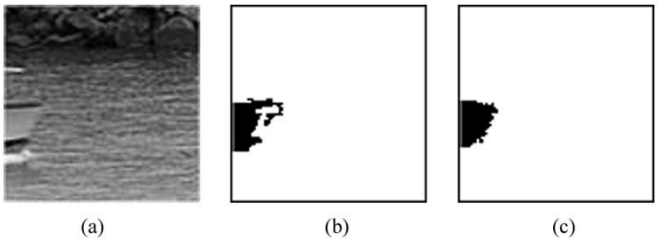



Fig. 8. Comparison between Bayes-based reclassification and Euclidean-distance-based reclassification for the improvement stage of the first frame segmentation. (a) Detail of the first frame of the *coastguard* sequence. (b) Coastguard ship after Euclidean-distance-based reclassification. (c) After Bayes-based reclassification. It is clear that region boundaries are better formed in (c).

where the last desired merging is followed by the first undesired one. Since this transition is expected to result in a large step-up of the value of the region similarity measure, as opposed to less significant changes induced by consecutive desired or consecutive undesired mergings, the process should stop when the ratio of the error increase function between successive mergings is maximized. The values of this ratio and the merging termination point are shown in Fig. 13 for four sequences.

Let $D_{U,k}, k = K-1, \ldots, 1$ be the value of the region similarity measure for the merging that reduces the $k+1$ spatiotemporal regions to $k$: $D_{U,k} = \min_{m,n}\{D_U(s_m, s_n)\}$. These values are computed prior to the actual merging procedure by a simulation of it. Then, the actual agglomerative procedure merges regions until the following ratio:

$$\frac{D_{U,k}}{D_{U,k+1}}, \quad \forall k \in [K-2, 1], D_{U,k+1} > 0, D_{U,k} > D_{U,\min} \tag{16}$$

is maximized (for example, for the coastguard sequence the ratio is maximized for $k = 2$ (Fig. 13), thus three final objects are created). According to (16), mergings characterized by a region similarity value lower or equal to $D_{U,\min}$ are positively desired; for the remaining ones, the error increase ratio is evaluated. Although the simplest case $D_{U,\min} = 0$ could be assumed (meaning that only mergings not resulting in an increase

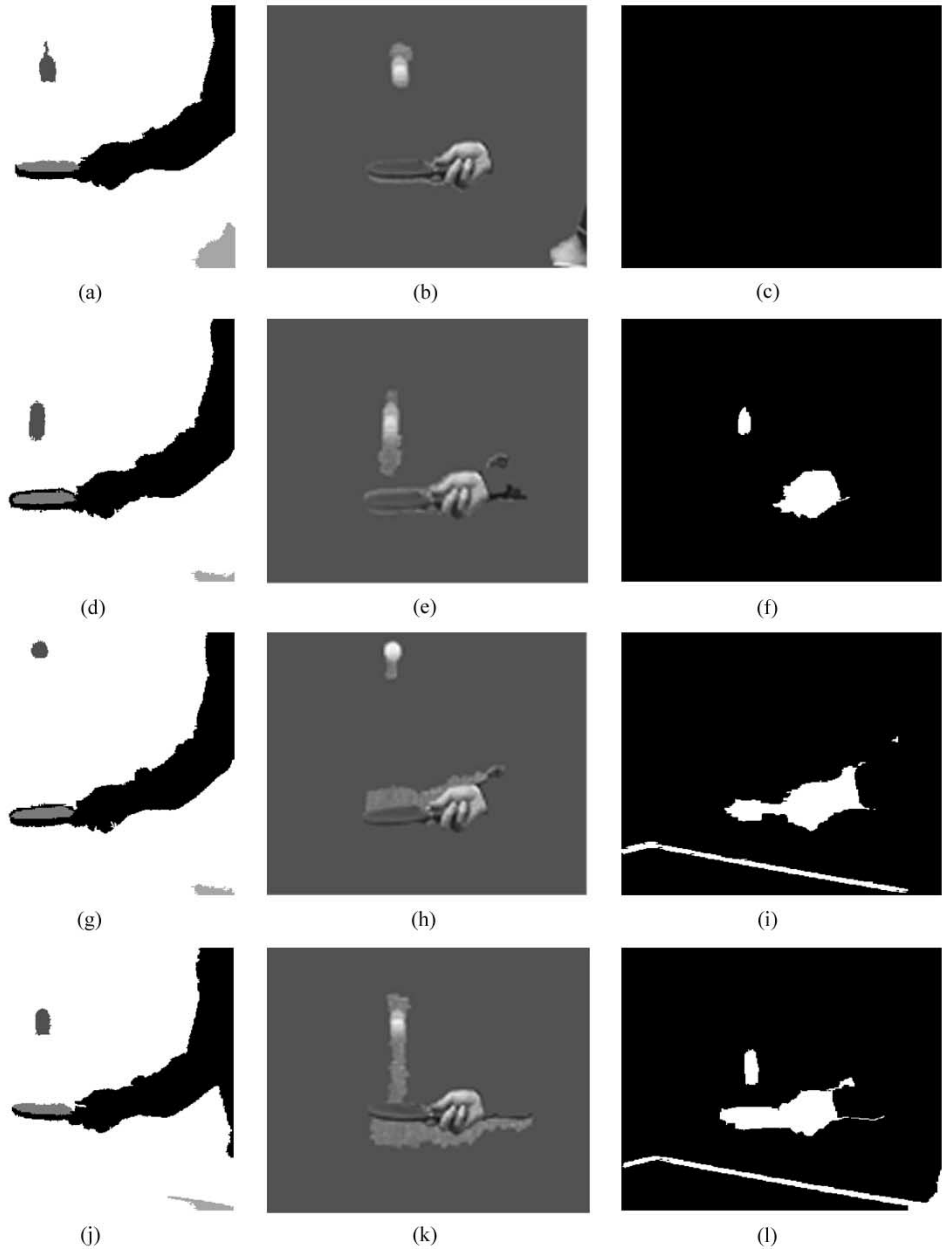



Fig. 9. Segmentation results for the first 30 frames of the CIF *table tennis* sequence. (a), (d), (g), and (j) Segmentation masks for frames $\sharp 1, \sharp 10, \sharp 20$, and $\sharp 30$, respectively, produced by the proposed segmentation algorithm. The region identified as background is painted white. (b), (b), (h), and (k) Corresponding segmentation results of Sifakis. (c), (f), (i), and (l) Corresponding segmentation results of the Cost211 Analysis Model.

of the sum of motion compensation errors are positively desired), a slightly higher value was found to improve the robustness of the merging termination. In our experiments, the value $D_{U,\min} = 0.1$ was used; however, as documented in Section V, the quality of the results is in fact highly insensitive to the value of this threshold.

### C. Supervised Region Merging

While this study addresses the problem of unsupervised spatiotemporal segmentation, supervised operation of the proposed algorithm is also possible by allowing the user to manually define the number of objects to be finally formed. Conformance to the user's desire is made possible by controlling the termination of the agglomerative process of the previous section by constantly evaluating the number of formed regions; region merging continues until the number of objects has reached the desired one. It is assumed that the desired number of objects is not greater than the number of remaining regions, after the enforcement of rules 5 and 6; this assumption is valid in practice, since experiments demonstrate that the frames before the application of the trajectory-based merging are oversegmented.

### D. Background Region Detection

Detecting which of the regions formed denotes the background can be useful both in coding and in indexing applications. Background detection, as soon as the image sequence has been segmented to spatiotemporal objects, relies on estimating the motion of the camera [38] and marking as background the object or objects whose perceived trajectory is coherent with the estimated camera motion.

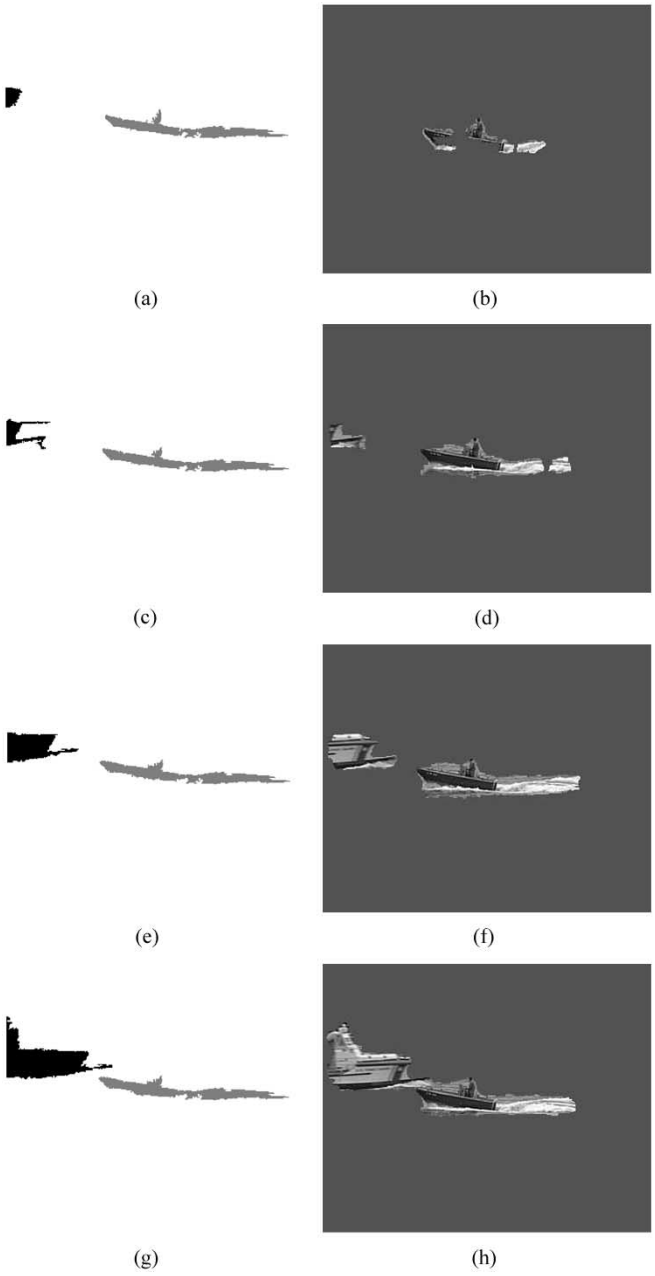



Fig. 10. Segmentation results for the first 30 frames of the CIF *coastguard* sequence. (a), (c), (e), and (g) Segmentation masks for frames $\sharp 1, \sharp 10, \sharp 20$, and $\sharp 30$, respectively, produced by the proposed segmentation algorithm. The region identified as background is painted white. (b), (d), (f), and (h) Corresponding segmentation results of Sifakis. The Cost211 Analysis Model could not identify any moving objects in the $T = 30$ first frames of this sequence; a moving object was identified in following frames.

Camera motion is estimated by the method suggested in [36] and employed, in combination with segmentation information, for region motion estimation in Section IV-A. This process produces a camera trajectory matrix $\mathbf{B}$ containing the parameters of the bilinear motion model for all frames, by applying the iterative rejection scheme to all blocks of each frame. Then, for every object the following error increment function is calculated:

$$D_B(s_k) = \frac{\sum_{t=1}^{T-1} \Upsilon_t(s_k) D_B^t(s_k)}{\sum_{t=1}^{T-1} \Upsilon_t(s_k)} \tag{17}$$

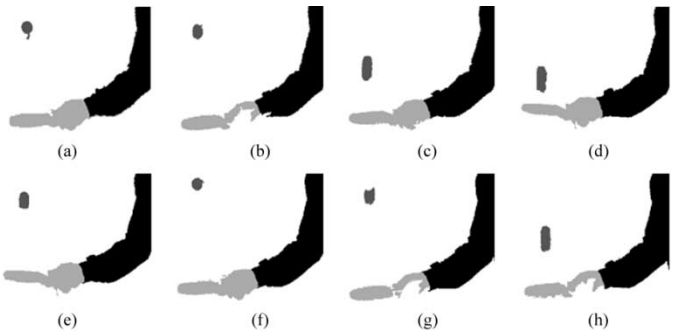



Fig. 11. Tracking results for a temporally subsampled by three *table tennis* sequence, $T = 10$. (a)–(h) Final segmentation masks for frames 2–9, namely, frames $\sharp 4, \sharp 7, \sharp 10, \sharp 13, \sharp 16, \sharp 19, \sharp 22$, and $\sharp 25$ of the original sequence. The tennis ball is being tracked, although there is no overlapping of the ball object between frames corresponding to mask pairs (b)–(c), (d)–(e), and (g)–(h).

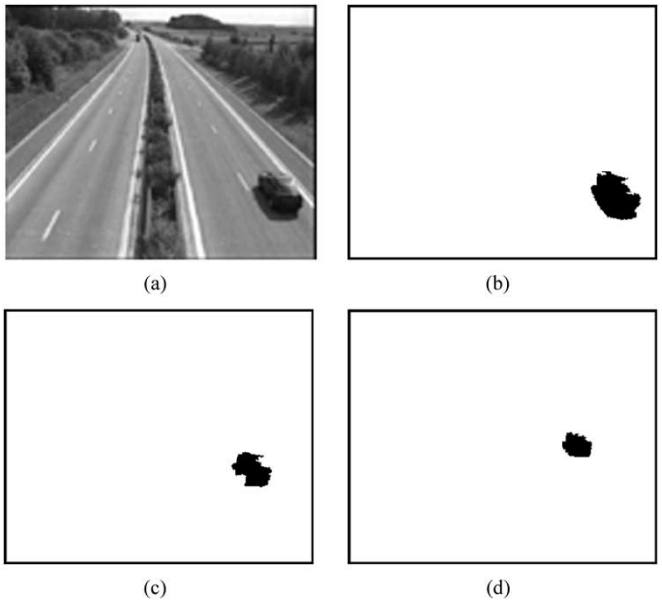



Fig. 12. Segmentation results for a modified *road surveillance* sequence, where $T = 20$ but frames $\sharp 5$ to $\sharp 15$ are identical; thus, the modified sequence shows an object (*car*) that moves in frames $\sharp 1$ to $\sharp 5$, then halts for the next ten frames, and subsequently resumes moving. (a) First frame. (b)–(d) Segmentation masks for frames $\sharp 1, \sharp 10$, and $\sharp 20$, respectively.

where $D_B^t(s_k) = (E_B^t(s_k) - E_k^t(s_k))/(N_k^t)$; $E_k^t(s_k), E_B^t(s_k)$ are the mean-square motion compensation errors for region $s_k$ at time $t$, using the model parameters calculated for this region [region trajectory matrix $\mathbf{U}(s_k)$, see (11)] and for the camera (camera trajectory matrix $\mathbf{B}$), respectively.

If the background is treated as a spatiotemporal object itself, then it can only be made of a single connected component in order to comply with the connectivity constraint; this would be the spatiotemporal region for which the value $D_B(s_k)$ is minimum. Alternately, if the constraint for background connectivity is relaxed, as was the case in our experiments, an arbitrary number of spatiotemporal regions may be assigned to the background, defined as the union of all regions for which the value $D_B(s_k)$ falls bellow an appropriate threshold.

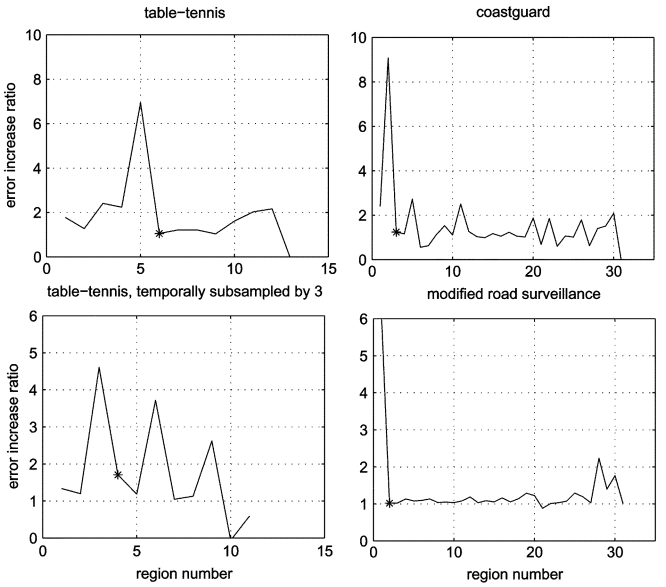



Fig. 13. Error increase ratio as a function of the number of regions. Starting from the spatiotemporal regions produced by the temporal tracking module, trajectory-based region merging takes place in an agglomerative fashion. This process is terminated when the next merging to take place would take the error increase ratio function to its global maximum. The termination point, indicating the final number of objects created by the algorithm, is marked with a star in the above diagrams. The values of the error increase ratio function are computed prior to the actual merging procedure by a simulation of it.

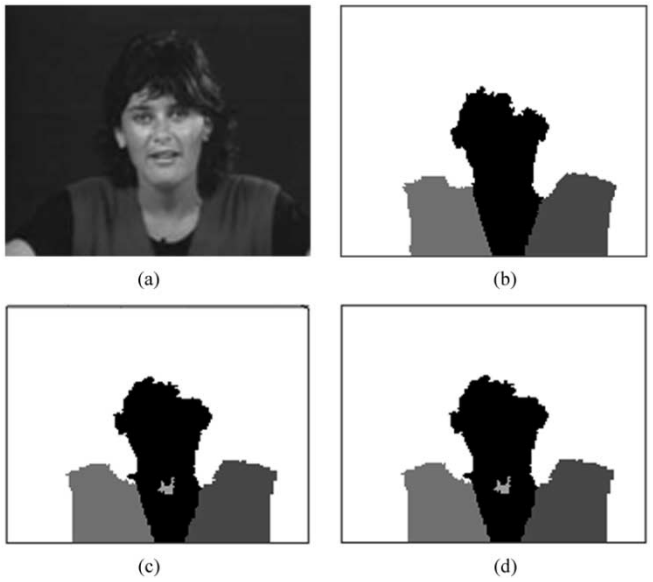



Fig. 14. Segmentation results for the QCIF *Miss America* sequence, where $T = 30$. (a) First frame. (b)–(d) Segmentation masks for frames $\sharp 1$, $\sharp 10$, and $\sharp 20$, respectively.

## V. Experimental Results

The proposed algorithms were tested on a number of test sequences. Here, results are presented for the *table tennis* and *coastguard* CIF sequences, as well as two temporally modified CIF sequences and the *Miss America* QCIF sequence.

Initially, the algorithm for first-frame segmentation, described in Section II, was applied to the first frame of each sequence. In Fig. 6(a) and (b), the first frames of two sequences are shown, followed by the corresponding segmentation masks [Fig. 6(c) and (d)] after the convergence of the KMCC algorithm.

At this point a variant of the Bayes classification procedure described in Section III-B can be used for the initial mask improvement mentioned in Section II. This improvement stage reclassifies the pixels on edges between objects, using a Bayesian approach similar to the one described in Section III. Compared to the approach of Section III, pixels are marked as disputed by evaluating their proximity to region boundaries rather than color difference between adjacent frames. The normalized histograms for each region are calculated using only the pixels not marked as disputed. The output of this procedure is presented in Fig. 6(e) and (f). The usefulness of the improvement stage can be seen in Fig. 6(e), where the tennis ball has a better shape than in Fig. 6(c). Note that in both cases the first frame has been somewhat oversegmented and, after the application of the aforementioned improvement stage, no region contains parts of the image belonging to two or more semantic objects. Comparison with the still-image segmentation method of [23] shows the importance of using motion features in detecting small but important moving objects in the first frame [Fig. 6(g) and (h)]. Fig. 8 illustrates the superior performance of the Bayes-based reclassification when compared to the Euclidean-distance-based reclassification scheme, where pixels are assigned based on their color difference with region centers.

Following the extraction of mask $R_1$ (segmentation mask corresponding to the first frame), tracking, as described in Section III, is performed for the remaining frames of the sequence; in our experiments, frames $\sharp 2$ to $T$, where $T = 30$ for the *table tennis* and *coastguard* sequences. The tracked regions are then grouped to semantic objects using the unsupervised framework of Section IV-B (Figs. 9 and 10). Before the creation of the final segmentation masks shown in the aforementioned figures, the background region detection method of Section IV-D was also performed to identify which of the formed regions should be assigned to the background. The resulting background object is shown in white in those figures.

In addition to these experiments, the temporal tracking and unsupervised trajectory-based merging algorithms were also applied to the first 10 frames of a temporally subsampled by three *table tennis* sequence (Fig. 11), using the first frame segmentation mask of Fig. 6(e). This aims to demonstrate the tracking algorithm's capability to handle fast moving objects that do not overlap if adjacent frames are superimposed, in this case the *ball* object.

In order to experimentally assess the advantage of employing the long-term trajectory of regions, a modified *road surveillance* sequence was used, with $T = 20$ and frames $\sharp 5$ to $\sharp 15$ being identical; thus, the modified sequence contains an object (*car*) that moves in frames $\sharp 1$ to $\sharp 5$, halts for the next ten frames, and subsequently resumes moving. As shown in Fig. 12, the proposed unsupervised algorithm identified correctly the moving object in all frames, whereas any algorithm performing region merging based on motion at the frame level would have lost track of it.

As can be seen from all of the aforementioned figures, the image regions have been tracked successfully and the proposed trajectory-based merging has grouped them in meaningful objects. Furthermore, the background detection module has succeeded in identifying the appropriate regions as background. In Fig. 13, the different values of the error increase ratio of (16) during the agglomerative merging procedure, and the termination point of it, are shown for four sequences. Table I presents, for

TABLE I
VALUES OF REGION SIMILARITY MEASURE FOR DESIRED/UNDESIRED MERGINGS

| Sequence | Last desired merging | First undesired one |
|---|---|---|
| *table tennis* | 0.64 | 4.45 |
| *coastguard* | 0.18 | 1.62 |
| *table tennis, temporally sub-sampled* | 2.93 | 13.51 |
| *road surveillance* | 4.90 | 29.70 |

TABLE II
THRESHOLD DEPENDENCY EXPERIMENTS

| Threshold | New values | Outcome |
|---|---|---|
| $DIF_{th} = 6$ | 4, 5, 7 & 8 | in one case, a new region substituted an old one (the racket in figure 15(a)&(b)), in two cases, the automatic termination of the trajectory-based merging resulted in more or fewer regions than expected (figure 15(c)) |
| $\{C_{th}, S_{th}\} = \{10,\ 0.2A\}$ where $A = \sqrt{p_{x,max}^2 + p_{y,max}^2}$ | $\{8,\ 0.15A\}$, $\{15,\ 0.3A\}$ | no changes |
| $\{\tau_{min}, l_{min}\} = \{4,\ 6\}$ (rules 5 and 6) | $\{2,\ 4\}$, $\{6,\ 8\}$ | $\tau_{min} = 2$ resulted in the second hand, entering the scene in frames ♯28-♯30, be identified as a valid object (figure 15(d)) |


Fig. 15. (a)–(d) Consequences of using threshold values deviating from those suggested, as discussed in Table II. In all cases the results are of good quality. In those cases where the automatic termination of the trajectory-based merging procedure fails, as in (c), supervised segmentation can be used to produce the desired number of objects.

each sequence, the values of region similarity measure $D_{U,k}$ for the last merging performed (last desired one) and the one that could have followed it (first undesired one). These figures support the claim that the choice $D_{U,\min} = 0.1$ is not restrictive or critical (Section IV-B); the minimum experimental value of $D_{U,k}$ for a merging judged as undesired is over 15 times higher. Additionally, Table I clearly shows that using a fixed threshold for terminating the merging procedure would have been insufficient.

The proposed unsupervised algorithm was also applied to the *Miss America* QCIF sequence; results for $T = 30$ are presented in Fig. 14.

To test the threshold dependency of the algorithm, some additional tests were conducted using threshold values deviating from those described in the previous sections. The values used for these tests and the corresponding results for the *table tennis* sequence are summarized in Table II and Fig. 15; in all cases the results are satisfactory. Comparable to these are the results for the temporally subsampled *table tennis*, the modified *road surveillance*, and the *Miss America* sequences. The *coastguard* sequence was found to be immune to these threshold changes: in all cases, three correct objects were created.

The results of the proposed unsupervised video segmentation algorithm compare favorably to the results presented by Sifakis [18], [39], where moving objects are detected by change detection using an appropriate reference frame and by subsequent object localization using local color features. The proposed video segmentation algorithm also compares favorably to the Cost211 Analysis Model [20]; this could not identify any moving objects in the $T = 30$ first frames of the *coastguard* sequence (Fig. 10); a moving object was identified in following frames. Results of the Cost211 Analysis Model [20] for the *table tennis* sequence are shown in Fig. 9; results of Sifakis [18] can be seen in Figs. 9 and 10. Additional comparisons using the *Miss America* sequence reveal that the proposed method does not produce oversegmentation of the facial area in contrast to, e.g., [17], while in the *table tennis* sequence the ball is correctly tracked in all frames, unlike, e.g., [40], where in some frames the ball is merged with the background.

TABLE III
LEGEND OF SYMBOLS

| Symbol | Description |
|---|---|
| $\mathbf{p} = [p_x\ p_y]$ | pixel |
| $\mathbf{I}_t(\mathbf{p})$, $\mathbf{V}_t(\mathbf{p})$ | pixel color/motion feature vector |
| $s_k^t$, $s_k$ | spatial/spatiotemporal region |
| $M_k^t$, $N_k^t$ | size of spatial region $s_k^t$ in pixels/blocks |
| $\overline{\mathbf{I}}(s_k^t)$, $\overline{\mathbf{V}}(s_k^t)$, $\overline{\mathbf{S}}(s_k^t)$ | Color/motion/spatial center of region $s_k^t$ |
| $D_I(s_m^t, s_n^t)$, $D_S(s_m^t, s_n^t)$ | Euclidean distance of color/spacial centers of regions $s_m^t$, $s_n^t$ |
| $K_t$ | number of spatiotemporal regions |
| $s_j^d$, $g_j$ | disputed region/set of regions neighboring to it |
| $P_k^t$, $P_j^d$ | characteristic values for region $s_k^t$/disputed region $s_j^d$, used for the detection of newly-introduced regions |
| $bb_{k,x}^t$, $bb_{k,y}^t$ | bounding box dimensions for region $s_k^t$ |
| $R_t$, $R_t^F$ | segmentation mask before/after trajectory-based merging |
| $R_t^I$, $R_t^E$, $R_t^N$ | segmentation masks used for temporal tracking |
| $\Upsilon_t(s_k)$ | region presence function |
| $\mathbf{b} = [b_x\ b_y]$ | square block of dimension $w$ |
| $\mathbf{V}_t^b(\mathbf{b})$ | motion vector of block $\mathbf{b}$ |
| $\mathbf{U}^t(s_k)$, $\mathbf{U}(s_k)$ | region motion parameter vector/matrix |
| $\mathbf{B}$ | camera motion parameter matrix |
| $E_m^t(s_m)$, $E_{m,n}^t(s_m)$, $E_B^t(s_m)$ | mean-square motion compensation error for region $s_m$, using region motion parameters estimated for $s_m/s_m \cup s_n$/the camera |
| $D_U^t(s_m, s_n)$, $D_U(s_m, s_n)$ | motion similarity of regions $s_m$, $s_n$ at time $t$/for the sequence |
| $D_{U,k}$ | the value of the region motion similarity measure for the merging that reduces the $k+1$ spatiotemporal regions to $k$ |
| $D_B^t(s_m)$, $D_B(s_m)$ | motion similarity between region $s_m$ and the camera at time $t$/for the sequence |

## VI. CONCLUSION

A methodology was presented for the segmentation of image sequences to semantic spatiotemporal objects. The proposed methodology is based on the popular scheme of segmenting the first frame and tracking the identified regions through the remaining frames. However, it employs novel algorithms both for first-frame segmentation and for tracking. In addition to that, motion is handled in a different way, as it is utilized at the sequence level rather than the frame level for merging regions to semantic objects. The unsupervised video segmentation algorithm resulting from combining the above algorithms handles fast-moving, newly appearing, and disappearing regions efficiently, as discussed in Section III and demonstrated experimentally.

The proposed video segmentation algorithm is appropriate for use as part of a content-based video coding scheme, in the context of the MPEG-4 standard, or a content-based multimedia application, such as video object querying, in the context of the MPEG-7 standard.

## REFERENCES

[1] F. Pereira, "MPEG-4: Why, what, how and when?," *Signal Processing: Image Communication, Tutorial Issue on MPEG-4*, vol. 15, no. 4–5, Jan. 2000.
[2] R. Koenen, "MPEG-4 multimedia for our time," *IEEE Spectr.*, vol. 36, pp. 26–33, Feb. 1999.
[3] R. Koenen and F. Pereira, "MPEG-7: A standardised description of audiovisual content," *Signal Process.: Image Commun.*, vol. 16, no. 1-2, pp. 5–13, 2000.
[4] L. Chiariglione, "MPEG and multimedia communications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 5–18, Feb. 1997.
[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] T. Sikora, "The MPEG-4 video standard verification model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 19–31, Feb. 1997.
[7] L. Patras, E. A. Hendriks, and R. L. Lagendijk, "Video segmentation by MAP labeling of watershed segments," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 23, pp. 326–332, Mar. 2001.
[8] R. Castagno, T. Ebrahimi, and M. Kunt, "Video segmentation based on multiple features for interactive multimedia applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 562–571, Sept. 1998.
[9] S. Herrmann, H. Mooshofer, H. Dietrich, and W. Stechele, "A video segmentation algorithm for hierarchical object representations and its implementation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 1204–1215, Dec. 1999.

[10] D. Gatica-Perez, C. Gu, and M.-T. Sun, "Semantic video object extraction using four-band watershed and partition lattice operators," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 603–618, May 2001.
[11] T. Meier and K. N. Ngan, "Automatic segmentation of moving objects for video object plane generation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 525–538, Sept. 1998.
[12] J. G. Choi, S.-W. Lee, and S.-D. Kim, "Spatio-temporal video segmentation using a joint similarity measure," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 279–286, Apr. 1997.
[13] 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.
[14] E. Tuncel and L. Onural, "Utilization of the recursive shortest spanning tree algorithm for video-object segmentation by 2-D affine motion modeling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, pp. 776–781, Aug. 2000.
[15] Y. Deng and B. S. Manjunath, "Unsupervised segmentation of color-texture regions in images and video," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 23, pp. 800–810, Aug. 2001.
[16] Y. Yokoyama, Y. Miyamoto, and M. Ohta, "Very low bit rate video coding using arbitrarily shaped region-based motion compensation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, pp. 500–507, Dec. 1995.
[17] D. Wang, "Unsupervised video segmentation based on watersheds and temporal tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 539–546, Sept. 1998.
[18] E. Sifakis, I. Grinias, and G. Tziritas, "Video segmentation using fast marching and region growing algorithms," presented at the Workshop Image Analysis for Multimedia Interactive Services, Tampere, Finland, 2001.
[19] E. Izquierdo and M. Ghanbari, "Key components for an advanced segmentation system," *IEEE Trans. Multimedia*, vol. 4, pp. 97–113, Mar. 2002.
[20] 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.
[21] J. McQueen, "Some methods for classification and analyis of multivariate observations," in *Proc. 5th Berkeley Symp. Mathematics Statistics and Probability*, vol. 1, 1967, pp. 281–296.
[22] 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.
[23] 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 Process.*, vol. 2002, no. 4, pp. 418–231, Apr. 2002.
[24] S. Liapis, E. Sifakis, and G. Tziritas, "Color and/or texture segmentation using deterministic relaxation and fast marching algorithms," in *Proc. Int. Conf. Pattern Recognition*, vol. 3, Sept. 2000, pp. 621–624.
[25] C.-L. Huang and B.-Y. Liao, "A robust scene-change detection method for video segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 1281–1288, Dec. 2001.
[26] R. Jain, R. Kasturi, and B. G. Schunck, *Machine Vision*. New York: McGraw-Hill, 1995.
[27] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. New York: Academic, 1990.
[28] Y. Wang, J. Ostermann, and Y.-Q. Zhang, *Video Processing and Communications*. Upper Saddle River, NJ: Prentice-Hall, 2002.
[29] S. Intille and A. Bobick, "Visual Tracking Using Closed-Worlds," MIT Media Laboratory, Cambridge, MA, Tech. Rep. TR-294, 1994.
[30] D. Tzovaras, M. G. Strintzis, and H. Sahinoglou, "Evaluation of multiresolution block matching techniques for motion and disparity estimation," *Signal Process.: Image Commun.*, vol. 6, no. 1, pp. 59–67, Mar. 1994.
[31] D. Tzovaras and M. G. Strintzis, "Motion and disparity field estimation using rate-distortion optimization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 171–180, Apr. 1998.
[32] S. Malassiotis and M. G. Strintzis, "Motion estimation based on spatiotemporal warping for very low bit-rate coding," *IEEE Trans. Commun.*, vol. 45, pp. 1172–1176, Oct. 1997.
[33] J. H. Lee, K. W. Lim, B. C. Song, and J. B. Ra, "A fast multi-resolution block matching algorithm and its LSI architecture for low bit-rate video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 1289–1301, Dec. 2001.
[34] S. Zhu and K.-K. Ma, "A new diamond search algorithm for fast block-matching motion estimation," *IEEE Trans. Image Processing*, vol. 9, pp. 287–290, Feb. 2000.
[35] J.-C. Tuan, T.-S. Chang, and C.-W. Jen, "On the data reuse and memory bandwidth analysis for full-search block-matching VLSI architecture," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 61–72, Jan. 2002.
[36] T. Yu and Y. Zhang, "Retrieval of video clips using global motion information," *Electron. Lett.*, vol. 37, no. 14, pp. 893–895, July 2001.
[37] 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.
[38] D. Tzovaras, N. Grammalidis, and M. G. Strintzis, "3-D camera motion estimation and foreground/background separation for stereoscopic image sequences," *Opt. Eng.*, vol. 36, no. 2, pp. 574–580, Feb. 1997.
[39] E. Sifakis and G. Tziritas, "Moving object localization using a multi-label fast marching algorithm," *Signal Process.: Image Commun.*, vol. 16, pp. 963–976, 2001.
[40] F. Moscheni, "Spatio-temporal segmentation and object tracking: An application to second generation video coding," Ph.D. dissertation, Ecole Polytechnique Fédérale de Lausanne, Lausanne, France, 1997.

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

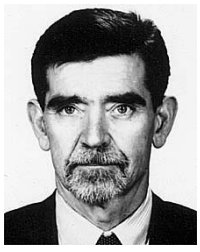
He is also a Graduate Research Assistant with the Informatics and Telematics Institute, Thessaloniki, Greece. His research interests include still image segmentation, video segmentation and object tracking, and content-based indexing and retrieval.

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 the 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 two-dimensional (2-D) and three-dimensional (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 retreival and knowledge discovery, and MPEG-4 and MPEG-7 standards. His involvement with those research areas has led to the coauthoring of two book chapters, 13 papers in refereed journals, and more than 40 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 13 projects in Greece, funded by the EC, and the Greek Ministry of Research and Technology.

Dr. Kompatsiaris is a member of the IEE Visual Information Engineering Technical Advisory Panel and a member of the Technical Chamber of Greece.



**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 VIDEO TECHNOLOGY since 1999. In 1984, he was the recipient of a Centennial Medal of the IEEE.