Multiple Moving Object Detection for Fast Video Content Description in Compressed Domain

F. Manerba, Jenny Benois-Pineau, R. Leonardi, Boris Mansencal

Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2008, Article ID 231930, 15 pages
doi:10.1155/2008/231930

Research Article

Multiple Moving Object Detection for Fast Video Content Description in Compressed Domain

Francesca Manerba,[1] Jenny Benois-Pineau,[2] Riccardo Leonardi,[1] and Boris Mansencal[2]

[1] *Department of Electronics for Automations (DEA), University of Brescia, 25123 Brescia, Italy*
[2] *Laboratoire Bordelais de Recherche en Informatique (LaBRI), Université Bordeaux 1/Bordeaux 2/CNRS/ENSEIRB, 33405 Talence Cedex, France*

Correspondence should be addressed to Jenny Benois-Pineau, jenny.benois@labri.fr

Received 20 November 2006; Revised 13 June 2007; Accepted 20 August 2007

Recommended by Sharon Gannot

Indexing deals with the automatic extraction of information with the objective of automatically describing and organizing the content. Thinking of a video stream, different types of information can be considered semantically important. Since we can assume that the most relevant one is linked to the presence of moving foreground objects, their number, their shape, and their appearance can constitute a good mean for content description. For this reason, we propose to combine both motion information and region-based color segmentation to extract moving objects from an MPEG2 compressed video stream starting only considering low-resolution data. This approach, which we refer to as "rough indexing," consists in processing P-frame motion information first, and then in performing I-frame color segmentation. Next, since many details can be lost due to the low-resolution data, to improve the object detection results, a novel spatiotemporal filtering has been developed which is constituted by a quadric surface modeling the object trace along time. This method enables to effectively correct possible former detection errors without heavily increasing the computational effort.

1. INTRODUCTION

The creation of large databases of audiovisual content in professional world and the extensively increasing use of consumer devices able to store hundreds of hours of multimedia content strongly require the development of automatic methods for processing and indexing multimedia documents.

One of the key components consists in extracting meaningful information allowing to organize the multimedia content for easy manipulation and/or retrieval tasks. A variety of methods [1] have recently been developed to fulfill this objective, mainly using global features of the multimedia content such as the dominant color in still images or video keyframes. For the same purpose, the MPEG4 [2], MPEG7 [3], or MPEG21 [4] family of standards does not concentrate only on efficient compression methods but it also aims at providing better ways to represent, integrate, and exchange visual information. MPEG4, for example, as the predecessor MPEG1,2, is a coding standard, but in some of its profiles, a new content-based visual data concept is adopted: a scene is viewed as a composition of video objects (VO), with intrinsic spatial attributes (shape and texture) and motion behavior (trajectory), which are coded separately. This information can then be used in a retrieval system as in [5] where object information coded in the MPEG4 stream, such as shape, is used to build an efficient retrieval system. On the other hand, MPEG7 does not deal with coding but it is a content description standard; and MPEG21 deals with metadata exchange and adaptation. They supply metadata on video content for instance, where object description may play an important role for subsequent content interpretation. However, in both cases, object-based coding or description, the creation of such object-based information for indexing multimedia content is out of the scope of the standard and is left to the content provider. In [5] as well, the authors suppose that the objects have been already encoded and they do not address the problem of their extraction from raw-compressed frames, which supposes the segmentation of the video. So, because of the difficulties to develop automatic reliable video

object extraction tools, object-based MPEG4 has not really become a reality and MPEG4 simple profile remains thus the most frequently used frame coding. Moreover, it is clear that the precision requirements and complexity constraints of object extraction methods are strongly application-dependent, so an effective object extraction from raw or compressed video still remains an open challenge.

Several approaches have been proposed in the past, and most of them can be roughly classified either as intraframe segmentation-based methods or as motion segmentation-based methods. In the former approach, each frame of the video sequence is independently segmented into regions of homogeneous intensity or texture, using traditional image segmentation techniques [6], while in the latter approach, a dense motion field is used for segmentation; and pixels with homogeneous motion field are grouped together [7]. Since both approaches have their own drawbacks, most object extraction tools combine spatial and temporal segmentation techniques [8, 9].

Most of these joint approaches concentrate on segmentation in the pixel domain, requiring high-computational complexity; moreover video sequences are usually archived and distributed in a compressed form, so video sequences have to be fully decoded before processing. To circumvent these drawbacks of pixel-domain approaches, a few compressed-domain methods have been attempted for spatiotemporal segmentation.

In [10], a region merging algorithm based on spatiotemporal similarities is used to extract the blocks of segmented objects in compressed domain; such blocks are then decompressed in the pixel domain to better detect object details and edges. In this last work, the use of motion information appears inefficient. In [11, 12] instead, to enhance motion information, motion vectors are accumulated over a few frames and they are further interpolated to get a dense motion vector field. The final object segmentation is obtained by applying the expectation maximization (EM) algorithm and finally by extracting precise object boundaries with an edge refinement strategy. Even if this method starts working with motion vectors extracted from a compressed stream, partial decoding is required for a subsequent refinement phase. These approaches, which can be considered as partial compressed-domain methods, although significantly faster than pixel-domain algorithms, cannot however be executed in real time.

Here, we propose a fast method for foreground object extraction from MPEG2 compressed video streams. The work is organized in two parts: in the first part, each group of picture (GOP) is analyzed and, based on color and motion information, foreground objects are extracted; the second part is a postprocessing filtering, realized using a new approach based on a quadric surface able to refine the result and to correct the errors due to the low-resolution approach.

The paper is organized as follows: in Section 2, the "rough indexing" paradigm is introduced; in Section 3, a general framework of the method is presented and it is developed in details in Sections 4, 5, and 6. In Section 4, we will explain how rough object masks can be obtained from P-frame extracted motion information and extrapolated to I-frames. Next, Section 5 describes how these results are combined with rough low-resolution color segmentation applied to I-frames to refine the object shape and to capture meaningful objects at I-frame temporal resolution. In Section 6, a spatiotemporal algorithm to derive approximate object shapes and trajectories is presented; and the way to use it to cancel errors resulting from previous stages of the approach is shown. Our comments on experimental results are introduced in Section 7 and finally some conclusions are drown in Section 8.

2. ROUGH INDEXING PARADIGM

The rough indexing paradigm is the concept we introduced in [29] to describe this new trend in analyzing methods for quick indexing multimedia content. In many cases [14, 15], for a rapid and approximate analysis of multimedia content, it is sufficient to start from a low (or intentionally degraded) resolution of the original material. Encoded multimedia streams provide a rich base for the development of these methods, as limited resolution data can be easily decoded from the compressed streams. Thus, many authors have proposed to extract moving foreground objects from compressed MPEG1,2 video with still background [16], by starting to estimate, as in [17], a global camera model without decompressing the stream in pixel domain. These rough data alone, for example, the noisy motion vectors and the DC images, can be used to achieve an acceptable level of indexing. Due to the noisiness of input data and due to missing information, "rough indexing" does not aim at a full recovering of objects in video but it is intended for a fast browsing of the content, that is to say when the attention is focused only on the most salient features of video scenes.

An example of "rough indexing" has been presented in [13] where an algorithm for real-time, unsupervised, spatiotemporal segmentation of video sequences in the compressed domain is proposed. This method works with low-resolution data using P-frame motion vectors and I-frame color information to extract rough foreground objects. When object boundaries are uncertain due to low-resolution data used, they are refined with a pixel-domain processing. The drawback is that if the object motion does not differ enough from the camera motion model or if the object is still, the algorithm can miss the detection. With the spatiotemporal filtering proposed in our work, we are able in most cases to locate the object and find its approximate dimensions using only the information associated to immediately previous and following frames.

In the next section, we describe our methodology developed in this "rough indexing" context.

3. METHODOLOGY FOR FOREGROUND OBJECT EXTRACTION

In this section, a brief overview of the proposed system is given, leaving the details for the subsequent sections. The global block-diagram for object extraction is shown in Figure 1; the blocks in italic are the ones that present something novel with respect to the literature.

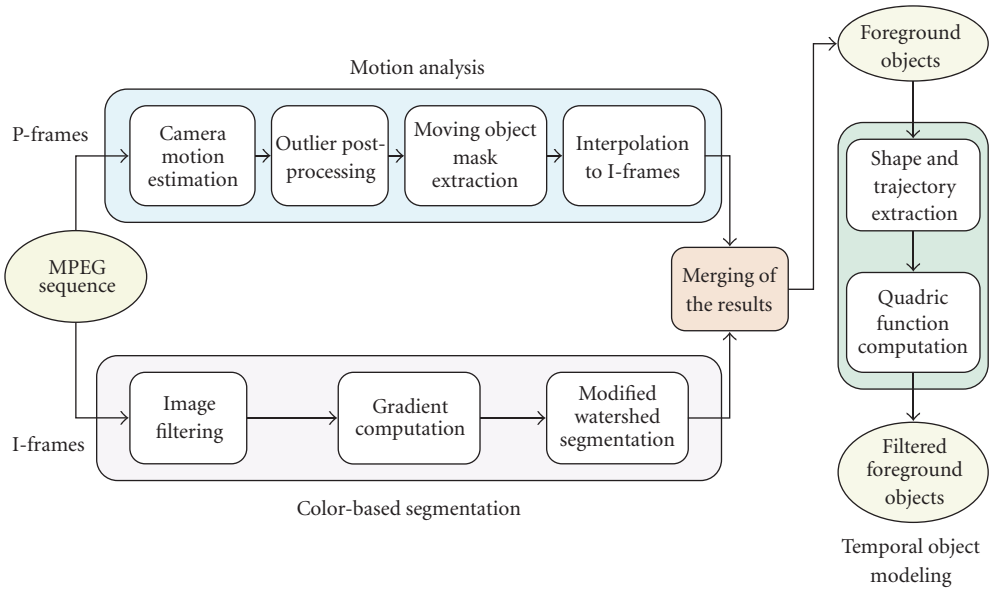


Figure 1: Flow chart of the proposed moving object detection algorithm.

The first part, illustrated on the left side of Figure 1, is based on a combined motion analysis and color-based segmentation which turns out to be an effective solution when working in the framework of a "rough indexing" paradigm: regions having a motion model inconsistent with the camera are first extracted from P-frames; these regions define the "object masks" since it is reasonable to expect that moving foreground objects are contained with a high probability within such regions. This kind of approach has demonstrated to be effective when looking for moving objects, so that many works in literature ([13], e.g.) use MPEG2 motion vectors to detect the outlier regions. A similar approach is also presented in [18] where MPEG motion vectors are used for pedestrian detection in video scenes with still background. In this case, motion vectors can be efficiently clustered. As in our work, in [18] too, a filtering is applied to refine the object mask result.

The upper left part of Figure 1 indicates the motion analysis process; we first perform the camera motion estimation starting from the P-frame macroblock motion vectors [19] to be able to separate the "foreground blocks" which do not follow the camera motion. Then, since it is necessary to discriminate the macroblock candidates which are part of a foreground object from the ones that are noisy, an outlier removal is performed by means of a postprocessing step. The next stage consists in evaluating the P-frame "object masks" using the results of two consecutive GOPs, to interpolate the "object masks" of the I-frame for which no motion vectors are available. In Section 4, the first part from camera motion estimation to I-frame "object mask" extraction is explained.

In parallel, a color-based segmentation of I-frames at DC resolution is realized by a morphological approach. The color-based segmentation is indicated in the lower left part of Figure 1 and can be subdivided into three steps: first, a preprocessing morphological filtering is applied to the I-frames of the sequence to reduce the image granularity due to the DC resolution, then a morphological gradient computation follows to detect the borders of homogeneous color regions. The final step is a modified watershed segmentation process, performed in the regions detected with the gradient computation, to isolate and label the different color regions. This morphological segmentation is presented in details in Section 5.

Once color and motion information for I-frames has been extracted, it is possible to merge them to obtain a first estimate of the foreground objects which appears, in most cases, quite accurate.

The second part of our approach is a novel temporal object modeling. It is based on the computation of the object shape and trajectory followed by the computation of a quadric function in $2D + t$ so as to model the object behavior for along time. At any given moment of time, the section of this function represents the rough foreground object shape so that the object can be recovered in those few cases where the first processing stage has led to an inaccurate detection.

In the next sections, all steps of our methodology will be described in details.

4. MOTION ANALYSIS

We assume that objects in a video scene are animated by their proper motion which is different from the global camera motion. The basic idea here to roughly define objects masks is to extract for each P-frame those foreground blocks which do not follow the global camera motion and to separate them out from noise and camera motion artifacts. The initial obtained resolution is at macroblock level and since for MPEG1 or MPEG2 compressed video, the macroblock size is limited to 16×16 pixels, the resulting motion masks have a very low resolution. The next step tries to increase this low

resolution. For this purpose, I-frames foreground moving objects are first projected starting from previously obtained P-frame rough object masks without any additional motion information usage, this way keeping the algorithm computationally efficient. This initial I-frame object extraction is then combined with a color-based segmentation (see Section 5) to increase the precision of the moving object detection. Before describing the segmentation process, the motion analysis that is first performed is discussed in more details.

4.1. Global camera motion estimation

In order to detect "foreground blocks" which do not follow the global camera motion, we have to estimate this motion first. Here, we consider a parametric affine motion model with 6 parameters, as the "parametric motion" descriptor proposed in MPEG7 which is defined as follows for each macroblock (x_i, y_i) with motion vector (dx_i, dy_i):

$$\begin{aligned} dx_i &= a_1 + a_2 x_i + a_3 y_i, \\ dy_i &= a_4 + a_5 x_i + a_6 y_i. \end{aligned} \tag{1}$$

The obtained estimation vector can be written as $\theta = (a_1, a_2, a_3, a_4, a_5, a_6)^T$ and it allows us to represent the different camera movements (pan, tilt, zoom, rotation).

To estimate vector θ that models the camera motion parameters from an MPEG2 macroblock motion field, we use a robust weighted least-square estimator (see [19] for more details) taking the MPEG2 macroblock motion vectors as measures. The robustness of the method is based on the use of Tukey biweight estimator [20]. This estimation process [19] not only gives the optimal values of the model parameters, but also assigns two additional parameters (w_{dx}, w_{dy}), called weights, to the initial measures which express their relevance to the estimated model in the (x, y) directions. Hence, it is possible to use this information to define "outliers" with respect to global motion model and then to use them in the so called "object masks" building process. This is illustrated in the next.

4.2. Outlier postprocessing in P-frames

Once the estimation of camera motion model is performed, the problem of object extraction can be formulated as separation of the macroblocks with irrelevant motion with respect to the estimated model so that objects in the frame with independent motion can be detected.

Let us consider a normalized grey-level image $I_{x,y}$, called camera motion incoherence image, defined using the weights w_{dx}, w_{dy} in the directions x and y and normalized to fit an interval $[0, I_{\max}]$ as follows:

$$I_{x,y} = \left[\left(1 - \max\left(w_{dx}, w_{dy}\right)\right) \cdot I_{\max}\right]. \tag{2}$$

Accordingly, the brighter pixels correspond to macroblocks with low weights and thus they belong to macroblocks that do not follow the global camera motion. Consequently, relevant pixels that well represent those areas with an independent motion are simply identified with a binary image $I^b_{x,y}$ obtained by threshold $I_{x,y}$.

The whole process is graphically exemplified in Figure 2. In Figure 2(a), a P-frame is shown with two objects of interest representing two walking women tracked by the camera. In Figure 2(b), we see the motion vectors associated to this frame. As it can be seen, in the middle of the frame, there are two regions with completely different motion vectors from their surroundings due to the presence of associated objects. Figure 2(c) shows the associated binary image $I^b_{x,y}$. The two white regions in the middle match the zones where the foreground objects are located. In Figure 2(c), it is possible to notice that on the right side of the frame some additional "outliers" exist because of camera motion. The problem is that in each frame there are some new macroblocks entering the frame in the direction opposite to the camera movement. The pixels of the original video frame for these macroblocks do not have any reference in the previous frame. Therefore, motion vectors are erroneous and do not follow camera motion in most cases so there are high irrelevance weights along these zones even if no foreground moving object is present.

Often, the outlier problem is solved in the literature by simply removing the border macroblocks from the whole image; instead, we prefer to filter the image using camera motion information (as we are going to explain in the next subsection) to ensure the preservation of possible useful information near the image boundaries.

With forward prediction motion coding, the displacement vector $\overline{d} = (dx, dy)^T$ of a macroblock in the current frame relates the coordinates of a pixel $(x_c, y_c)^T$ in the current frame to its reference pixel $(x_r, y_r)^T$ in the reference frame by

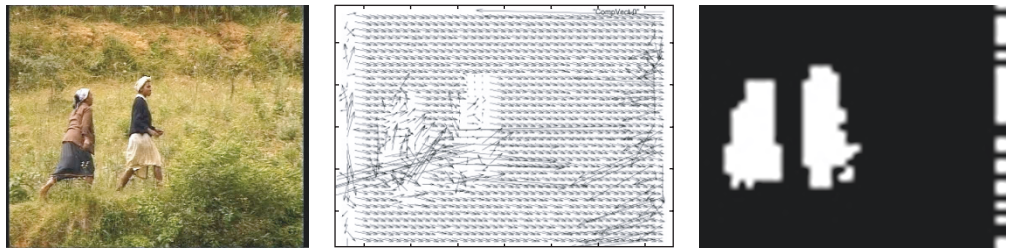
$$\begin{aligned} dx &= x_r - x_c, \\ dy &= y_r - y_c. \end{aligned} \tag{3}$$

Now using the camera model equations (1), we solve (3) for each of the reference frame camera corner macroblocks taking as reference pixels the corners of the reference frame. Consequently, the reference frame is warped into the current frame, leading to the geometry of the previous frame domain entering the current frame. If some "outliers" are present in that zone, we can assume that they have been caused by the camera motion so they are discarded from being candidate object masks (see Figure 3).

Repeating the method described above for all P-frames within a single video shot, we obtain the motion masks for all the foreground moving objects in the shot which represent a first guess for the foreground moving object at the reduced temporal resolution according to the previously introduced rough indexing paradigm.

4.3. Moving object mask extraction in I-frames

The approximated motion masks estimated so far represent a good guess for locating the foreground moving object shape in P-frames. Nevertheless, using motion information alone is not sufficient for a robust and to some extent for accurate object extraction. Thus, we propose to merge the motion masks with the result of a color-based intraframe segmentation process performed on the I-frames. Since motion masks have been obtained for P-frames only, we have to build



(a) (b) (c)

Figure 2: Extraction of motion masks from P-frames: (a) the original P-frame; (b) the associated motion vectors; (c) the corresponding binary image $I^b_{x,y}$, SFRS-CERIMES.

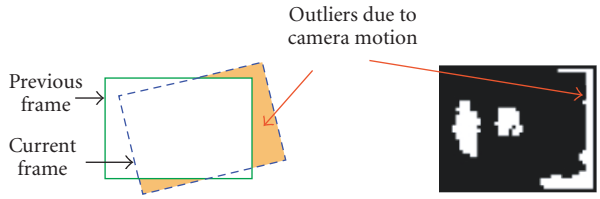


Figure 3: An example of outlier detection as a result of camera motion.

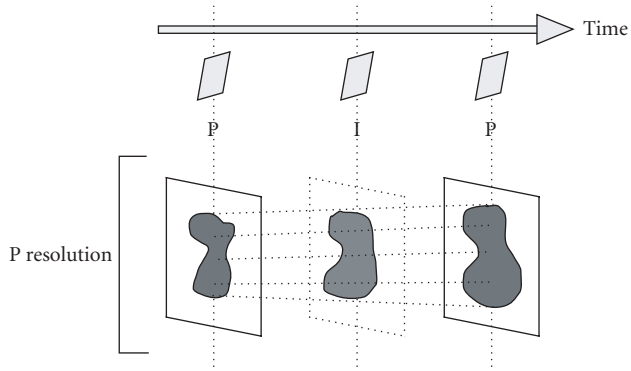


Figure 4: Motion mask construction for I-frames: creation of the mask for the I-frame by interpolation of two P-frames.

the corresponding masks for the I-frames in order to overlap them to color-based segmentation result. As the MPEG decoder does not give motion vectors for I-frames, we cannot extract the mask using the information available in an MPEG stream as we have done for P-frames, but we can have a good estimate interpolating the masks available in adjacent P-frames so as to predict a projection of such motion masks on the I-frame.

The interpolation can be fulfilled by two approaches: (i) a motion-based one [22], where the region masks are projected into the frame to be interpolated; (ii) a simpler spatiotemporal interpolation without using the motion information. For the sake of low computational cost, we decided to use a spatiotemporal interpolation (as shown in Figure 4) using a morphological filter. As a result, the binary mask in I-frame $\widetilde{I}^b_{x,y}(t)$ is computed as

$$\widetilde{I}^b_{x,y}(t) = \min\left(\delta\widetilde{I}^b_{x,y}(t-\Delta t), \delta\widetilde{I}^b_{x,y}(t+\Delta t)\right). \quad (4)$$

Here, δ denotes the morphological dilation with a 4-connected structural element of radius 1, $\widetilde{I}^b_{x,y}(t-\Delta t)$ and $\widetilde{I}^b_{x,y}(t+\Delta t)$ are the binary masks of previous and next P-frames, respectively. In this way, we obtain the mask for the I-frame that exhibits the approximate position of the objects.

This process leads to a rough estimate of the mask for the I-frame which will approximately locate the objects in the I-frame. Figure 5 depicts some I-frames extracted for an MPEG2 video and the resulting I-frame masks.

5. OBJECT MASK REFINEMENT BY COLOR SEGMENTATION

Interpolated motion masks for I-frames indicate the likely locus of objects with independent motion but with limited resolution, so using I-frame color information inside such masks, we refine the object shapes and furthermore estimate their appearance (color, texture information, and so on), thus indexing the video content by spatial features at I-frame temporal resolution.

For this reason, a color segmentation is performed on the I-frame to subdivide it into homogeneous regions (recall Figure 1). Regions overlapping with the foreground moving object masks are retained and they represent the set of objects of interest. In order to follow the rough indexing paradigm, only DC coefficients of the I-frames are taken into account [23] since they are easily extracted from the compressed stream with only partial decoding.

In this work, we applied a morphological approach for color-based segmentation that we first proposed for full, mid, and low-resolution video for MPEG4, MPEG7 content description [24]. The approach follows the usual morphological scheme: simplification-filtering, computation of morphological gradient, watershed-based region growing. Here, we will briefly describe these principal steps and justify their necessity for low-resolution DC frames.

The first step, simplification, is useful for DC frames to smooth the typical granularity of DC images. This simplification is realized by open-close filter with partial reconstruction. The morphological gradient is then calculated on the simplified signal (see [29] for more details).

The particularity of the third step is a simplified version of a classical watershed [30]. The main difference is twofold. First of all, in a classical watershed, at the initialization, only

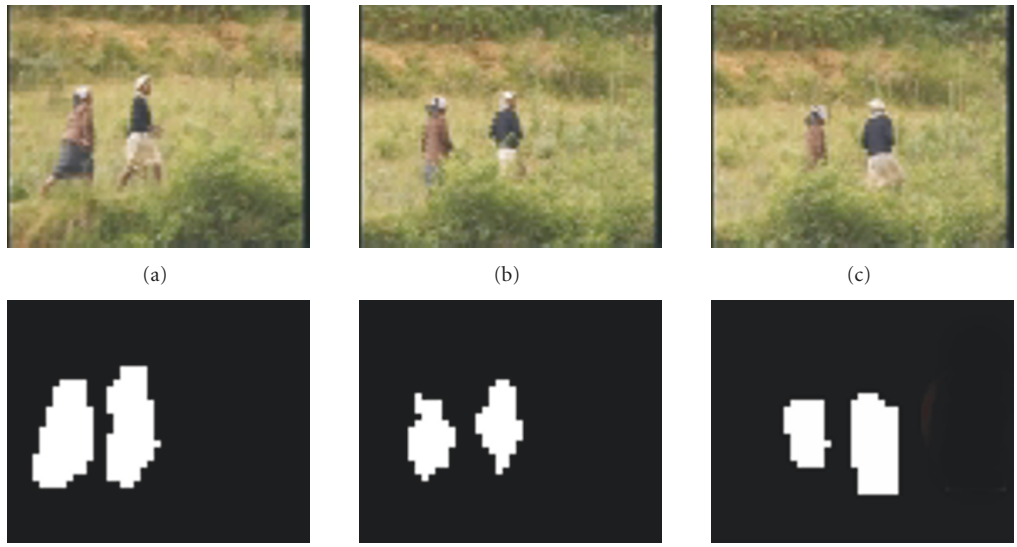


Figure 5: Extracted motion masks from I-frames: (a), (b), (c) original I-frames at DC resolution; (d), (e), (f) the corresponding masks, SFRS-CERIMES.

zero gradient values are taken as seeds for "water" propagation. In our scheme, all pixels with gradient values lower than a threshold are labelled in connected components. These connected components are considered as a marker image, that is, seeds for regions. Secondly, in a classical watershed, a creation of new regions is possible at each grey level. In our method, the creation of new regions is prohibited. Instead, we keep on making grow initial connected components. This region growing algorithm is realized in a color space with progressively relaxed threshold. Thus, a pixel from a strong gradient area (uncertainty area) is assigned to its neighboring region if

$$|I^Y(x,y) - m^Y| + |I^U(x,y) - m^U| + |I^V(x,y) - m^V| < 3F(\overline{m})g(\Delta). \quad (5)$$

Here, $(m^Y, m^U, m^V)^T$ is the color mean value of the region, $F(\overline{m}) = |\overline{m} - 127| + 128$. The function $F(\overline{m})$ in (5) depends on the mean color level $\overline{m} = (m^Y + m^U + m^V)/3$ of the considered region and is adjusted according to the principles of the Weber-Fechner law, which implies that the gray-level difference which the human eye is able to perceive is not constant but depends on the region intensity.

The function $g(\Delta)$ is an incremental term that progressively relax the thresholds to merge boundary pixels of increasing grey-level difference. The threshold is continuously relaxed until all uncertain pixels are assigned to a surrounding region (see [24] for more details). Figure 6 shows the result of the segmentation process. Here, the original low-resolution DC frame is presented in Figure 6(a), the marker (black) and uncertainty (white) pixels are presented in Figure 6(b), the resulting region map with mean color per region is shown in Figure 6(c).

As our previous studies show [24], this modified watershed algorithm reduces the number of regions, as the creation of new regions is prohibited. Furthermore, the initialization step already gives regions of larger area than the initialization by gradient.

The modified watershed algorithm is of the same complexity as a classical watershed, but the number of operations is reduced. Let us consider $\overline{n}$ the new number of pixels from uncertain areas to be assigned to one region at each iteration, J the number of iterations, and K the number of initial regions. Then in our modified watershed, the mean complexity is $K\overline{n}J$. In a classical watershed, if the number of new regions to be added at each iteration is K_j then the mean complexity would be $\overline{n}(KJ + \sum_{j=1}^{J} K_j)$.

Once the above I-frame segmentation has been performed, foreground objects are finally extracted from I-frames by superimposing and merging motion masks and color regions at DC frame resolution. In Figure 7, we show examples of intraframe segmentation within the projected foreground object masks for the sequence "de l'arbre l'ouvrage" (see [29] for more details on this first part). It can be seen that, in general, the segmentation process makes clear the aliased structure of object borders (due to DC image formation), but still gives a good overview of an object.

6. SPATIOTEMPORAL FILTERING USING QUADRIC SURFACES

Once color and motion information have been merged, moving foreground objects at I-frame temporal resolution are obtained. However, as I-frames are processed independently, one from the others, no information about objects variations in time is given. Furthermore, it may happen that if the object movement does not differ a lot from the camera motion or the object is still, this object cannot be detected or some of its components could be lost. In fact, nothing can

(a) (b) (c)

Figure 6: Morphological color-based segmentation: (a) original DC frame, (b) morphological gradient after threshold, (c) region map.

Figure 7: Examples of "intraframe" segmentation for the sequence "De l'arbre l'ouvrage," SFRS-CERIMES.

prevent some annoying effects of segmentation such as flickering [26], especially when dealing with low-spatial and temporal resolution. Nevertheless, it can be assumed that an object cannot appear and disappear rapidly along a short sequence of frames, so we can preserve existing moving objects at I-frame temporal resolution and try to recover any "lost" information. The objective here is to build the object trajectory along the sequence of I-frames starting from its initial estimates and then to approximate its shape with a quadric surface for all other frames where it might have been poorly or not detected at all. The purpose is to try to extract a sort of "tube" where each section (namely, the intersection of the tube with the I-frame image plan) along time represents the object position at every frame.

Therefore, the tube sections at each moment of time approximate the object shape and can be used to recover any mistakes occurred in the first stage of the object extraction process. Furthermore, the visualization of the tube along time will provide information about the temporal evolution of objects.

As a natural video sequence can contain several objects, the preliminary step for tube construction consists in the identification of the same object from the detected masks in consecutive I-frames. Consequently, the scheme for spatiotemporal filtering comprises of two stages (see Figure 1).

(i) Object identification and trajectory computation.
(ii) Object fitting by quadric surfaces.

Object identification and trajectory computation

The objective is to separately track the extracted objects along each I-frame. To do this, we estimate the motion for each detected object and project the object mask from I-frame at time t to I-frame at time $t + \Delta t$. If such projection overlaps with the result of the moving object detection in the forward I-frame (at time $t + \Delta t$), the object is confirmed for the considered pair of frames. To perform the projection, the object motion has to be estimated.

As we are interested in the global object motion considered as a rigid mask, we can suppose that the motion of each object O_k can be sufficiently well described using the affine model (introduced in (1)) for a pair of I-frames at times t and $t + \Delta t$. Since we have no motion vectors in the MPEG stream, to determine the I-frame motion model, we can interpolate the motion vector fields of the object from closer P-frames. Such motion vectors are then used by a least square estimator [25] to estimate the global object motion model θ_k.

The estimated motion vector given by $\theta_k = (a_0, a_1, a_2, a_3, a_4, a_5)^T$ describes the object O_k movement. Once θ_k has been obtained, the object motion model is reversed for all involved I- and P-frames so as to define the object projected location, this way linking the object along the sequence between any two consecutive I-frames.

Next step then is to calculate the object trajectory, which will become the principal axis of the quadric surface to be computed. As it may happen that the objects are not correctly detected or occluded, the real object centers can be different from the estimated one. As we suppose that the object motion does not change along the sequence, we can suppose that the object centers also follow a straight line trajectory or a piecewise linear approximation. It can seem a weak assumption but if we consider that in most cases the length of a GOP varies between 15 to 30 frames in NTSC or 12 to 25 in PAL SECAM; taking into account only I-frames means that we observe the object position every half a second, and in most cases, it can be observed that the object trajectory with respect to the camera is constant in such time interval. We have observed that in short sequences the objects follow a straight-line trajectory, while in longer ones, it has been necessary to use a piecewise linear approximation to model the object behavior. To obtain the approximation of the object center of mass line, we use again the least-square fitting.

Object fitting by quadric surfaces

To recover objects in the frames where a miss detection has occurred, we will construct a spatiotemporal tube and we will center it on the trajectory computed in the previous step. In order to use a suitable model, we assume that the object trajectory is linear in the simpler cases and that a piecewise linear approximation can be employed in the more complex ones. Accordingly, all objects have to be aligned prior to computing the tube approximation, as will be explained later. Based on this assumption, we propose as tube model a quadric surface in a $(2D + t)$ dimensional space.

Generally speaking, a quadric equation in an n-dimensional space is written as follows:

$$\sum_{1\le i\le j\le n} a_{ij}x_i x_j + \sum_{1\le i\le n} b_i x_i + c = 0, \tag{6}$$

where a_{ij}, b_i, c are coefficients in the n-dimensional space, and at least one of the a_{ij} is different from zero; in the particular case of $n = 3$, the function is called *quadric surface* [27] and becomes

$$\begin{aligned} f(x, y, t; a_{ij}) = {} & a_{11}x^2 + a_{22}y^2 + a_{33}t^2 + a_{12}xy + a_{13}xt \\ & + a_{23}yt + a_{14}x + a_{24}y + a_{34}t + a_{44} = 0. \end{aligned} \tag{7}$$

The purpose now is to find the coefficient a_{ij} in (7) that can best approximate the contours of the moving objects in the sequence.

Usually, finding the best approximation of the objects with this surface means to compute the parameters a_{ij} that minimize the distance between $C^k_{x,y,t}$, intended as the contour of the object O^k at the time instant t , and the quadric $f(x, y, t; a_{ij}) = 0$ defined in (7), that is,

$$\min_{a_{ij}} \sum d(C^k_{x,y,t} - f(x, y, t; a_{ij})). \tag{8}$$

This minimization problem is not so easy to be solved as it could seem. In fact, the function to be minimized is the sum of the distances in the different instants of time of two curves which is not even easy to be defined. Moreover, the presented problem is not linear, that is, a variable cannot be written as function of the others maintaining a linear relation between the explicit variable and the parameters a_{ij}; in this

last condition, in fact, some fast methods could be used to easily solve the problem. Because of these difficulties, we can propose a different solution which can nonetheless give us a good approximation, even if it is not the optimal one.

Instead of considering the object contour which is quite difficult, we consider a new image obtained computing the function $z(x, y, t)$ which is a 2D Gaussian function centered on the object centroid (μ_x, μ_y) and with variance values (σ_x, σ_y) obtained in this way: the estimated coordinates of the optimal straight-line $(x_c(t), y_c(t))$ are used to set $\mu_x(t) = x_c(t), \mu_y(t) = y_c(t)$ for each value of t. The standard deviations $(\sigma_x(t), \sigma_y(t))$ are represented by the maximum distance between the optimal center of mass (x_c, y_c) and the object bounding box in both x and y directions (see Figure 8(b)). So $z(x, y, t)$ becomes

$$z(x, y, t) = \exp\left(-\frac{1}{2}\left(\frac{(x(t) - \mu_x(t))^2}{\sigma_x(t)^2} + \frac{(y(t) - \mu_y(t))^2}{\sigma_y(t)^2}\right)\right). \tag{9}$$

In Figure 8, an example of z function computation is given. In Figure 8(a), a DC image of the sequence is presented and in Figure 8(b), the corresponding object masks are shown; in particular, for the object on the left, σ_x and σ_y are depicted. It is possible to notice that in this case the centroid does not correspond to the center of mass of the object mask, in fact in this case, the object is half-detected, so when computing the object trajectory using the least-square approximation as illustrated in the previous paragraph, using the masks of the adjacent frames, it is possible to partially correct the detection and to obtain a more realistic center of mass.

We may have chosen instead of 9 any other function with the same characteristics, that is, having maximum value on the object centroid and decreasing values as a function of object size.

Then, we force the quadric equation to verify

$$\begin{aligned} z(x, y, t) = {} & a_{11}x^2 + a_{22}y^2 + a_{33}t^2 + a_{12}xy + a_{13}xt \\ & + a_{23}yt + a_{14}x + a_{24}y + a_{34}t + a_{44}. \end{aligned} \tag{10}$$

This is translated into forcing a sort of regular behavior in time for the $z(x, y, t)$ functions, which are obtained independently one from the other, at each time t.

The result will not be exactly a quadric, but a function in four dimensions x, y, t, z representing a set of quadrics with the same axis and different extent so that fixing a value of z, it will be possible to obtain different quadrics which will depend on the quality of the values $(\mu_x(t), \mu_y(t))$ and $(\sigma_x(t), \sigma_y(t))$ used, the latter being related to the object characteristics.

Equation (7) represents a generic quadric function, but for the purpose it is being used, it can be simplified and only some specific cases can be considered. As one is not interested in recovering the 3D volume but only in the volume slices along the time, the computation is reduced by forcing all object center of mass to lie parallel to the time axis. This eliminates all xy, xt, or yt in (7).

Furthermore, we can add to (10) some further constraints to the parameters to avoid degenerate cases (such as a couple of planes). Under this constraints, (10) in this way becomes

$$z(x, y, t) = a_{11}x^2 + a_{22}y^2 + a_{33}t^2 + a_{14}x + a_{24}y + a_{34}t + a_{44}. \tag{11}$$

Adopting further a canonic form of the quadric solution centered in (x_0, y_0), assuming positive values of t, we have

$$\begin{aligned} & z(x, y, t) \\ & \quad = a_{11}x^2 + a_{22}y^2 + a_{33}t^2 - 2a_{11}x_0x - 2a_{22}y_0y + a_{34}t + a_{44} \end{aligned} \tag{12}$$

with the following constraints adopted to avoid degenerate cases:

$$\begin{aligned} a_{11} &> 0, \\ a_{22} &> 0. \end{aligned} \tag{13}$$

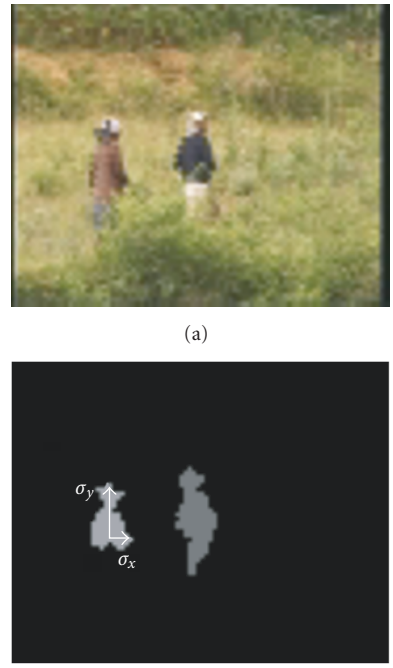
The problem has been reduced to estimate the five parameters in (12) to obtain the function which best approximates the evolution of object shape and dimensions along the sequence.

Given the set of coordinates $(x_1, y_1, t_1), \ldots, (x_W, y_H, t_N)$ for the sequence of N I-frames of dimensions $W \times H$, given the vector of measures $z = [z_1, \ldots, z_{W \times H \times N}]^T$ computed on this set of coordinates, we can write (12) in a matrix form, as

$$z = \mathcal{H}\beta \tag{14}$$

under the constraint

$$\mathcal{A}^T\beta > 0, \tag{15}$$



(a)

(b)

Figure 8: Computation of standard deviation on the extracted objects: (a) original DC frame; (b) σ_x and σ_y on the object mask, SFRS-CERIMES.

where β is the parameter vector. Here,

$$\mathcal{H} = \begin{bmatrix} x_1^2 - 2x_0x_1 & y_1^2 - 2y_0y_1 & t_1^2 & t_1 & 1 \\ \vdots & & & & \vdots \\ x_N^2 - 2x_0x_N & y_N^2 - 2y_0y_N & t_N^2 & t_N & 1 \end{bmatrix},$$
$$\mathcal{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}. \tag{16}$$

Let us denote by $\mathbf{e}(\beta) = \mathbf{z} - \mathcal{H}\beta$ the error with respect to the exact model (14). We will solve the following optimization problem:

$$\begin{aligned} &\min && \frac{1}{2}\mathbf{e}^T\mathbf{e}, \\ &\text{under constraint} && \mathcal{A}^T\beta \geq 0. \end{aligned} \tag{17}$$

This is a *quadratic programming*. Generally speaking, if a problem of quadratic programming can be written in the form

$$\begin{aligned} &\min && \frac{1}{2}\mathbf{x}^T\mathcal{G}\mathbf{x} + \mathbf{g}^T\mathbf{x}, \\ &\text{under constraint} && \mathcal{A}^T\mathbf{x} - \mathbf{b} \geq 0, \end{aligned} \tag{18}$$

then it is possible to define the dual problem [28]

$$\begin{aligned} &\max && \frac{1}{2}\mathbf{x}^T\mathcal{G}\mathbf{x} + \mathbf{g}^T\mathbf{x} - \lambda^T(\mathcal{A}^T\mathbf{x} - \mathbf{b}), \\ &\text{under constraint} && \mathcal{G}\mathbf{x} + \mathbf{g} - \mathcal{A}\lambda = 0 \quad \text{with } \lambda \geq 0, \end{aligned} \tag{19}$$

where λ is a vector of *Lagrange multipliers*. Equation (19) can be rewritten as

$$\begin{aligned} \max \quad & -\frac{1}{2}\lambda^T(\mathcal{A}^T\mathcal{G}^{-1}\mathcal{A})\lambda \\ & + \lambda^T(\mathbf{b} + \mathcal{A}^T\mathcal{G}^{-1}\mathbf{g}) - \frac{1}{2}\mathbf{g}^T\mathcal{G}^{-1}\mathbf{g}, \\ \text{under constraint} \quad & \lambda \geq 0. \end{aligned} \tag{20}$$

This is still a quadratic programming problem in λ, but it is easier to solve. Once the value of λ has been found, the value of $\mathbf{x}$ is obtained by solving (19). In our case, developing (17) for $\mathbf{e} = \mathbf{z} - \mathcal{H}\beta$, we obtain

$$\begin{aligned} &\min && \frac{1}{2}\beta^T(\mathcal{H}^T\mathcal{H})\beta - \mathbf{z}^T\mathcal{H}\beta + \frac{1}{2}\mathbf{z}^T\mathbf{z}, \\ &\text{under constraint} && \mathcal{A}^T\beta \geq 0. \end{aligned} \tag{21}$$

This problem is in the same form of (19). It can be rewritten in the form of (20), where $\mathcal{G} = \mathcal{H}^T\mathcal{H}$ and $\mathbf{g} = -\mathcal{H}^T\mathbf{z}$. The value of λ is obtained by solving the derivative in (20):

$$\lambda = -\left[\mathcal{A}^T(\mathcal{H}^T\mathcal{H})^{-1}\mathcal{A}\right]^{-1}\left[\mathcal{A}^T(\mathcal{H}^T\mathcal{H})^{-1}\mathcal{H}^T\mathbf{z}\right]. \tag{22}$$

Consequently, the vector β can be obtained from (19) setting β to $\mathbf{x}$, $\mathcal{G}$ to $\mathcal{H}^T\mathcal{H}$, and $\mathbf{g}$ to $-\mathcal{H}^T\mathbf{z}$:

$$\beta = (\mathcal{H}^T\mathcal{H})^{-1}(\mathcal{A}\lambda + \mathcal{H}^T\mathbf{z}). \tag{23}$$

Now with these optimal parameters, a set of quadric surfaces with different extent but with the same central axis can be obtained. To compute the function that best fits all the object masks, we have to fix the value of z.

To choose the value of z and find a unique quadric surface that gives a good approximation of all object masks, we minimize the following global criterion:

$$\min \sum_{(x,y,t)} \delta(x, y, t), \tag{24}$$

where, for a fixed t,

$$\delta(x, y) = \begin{cases} \alpha_1 & \text{if } (x, y) \in (\text{quadric section} - \text{mask}), \\ 0 & \text{if } (x, y) \in (\text{quadric section} \cap \text{mask}), \\ \alpha_2, & (x, y) \in (\text{mask} - \text{quadric section}), \end{cases} \tag{25}$$

with $\alpha_2 \gg \alpha_1$. This function will privilege "larger" quadrics enclosing object masks.

The result of quadric computation for an extract of "aquaculture" sequence at I-frame resolution is shown in Figure 9.

It can be seen that when objects are not detected due to the very weak relative motion with respect to the camera, the quadric section still allows for object location in the frame.

In this work, the overlapping of objects is handled only partially. If objects that were separated in a given frame superimpose, that is, will partially occlude in the next frame, we will be able to identify which object is closer to the viewpoint by collecting motion vectors in the projected bounding box and identifying the object label in the past frame with the estimated motion model. If the objects overlap strongly, then the tube will be maintained only for the object closer to the viewpoint. In case of objects crossing their trajectories, when an object will reappear in the sequence, we start a new tube.

An example of overlapping objects is given in Figure 10. On the first frames, we have three objects, of which two overlap. These two objects are detected as only one object, then when they split, the object the most in the background is identified as a new object, and thus a new tube is created.

We are conscious that such a method is limited. We cannot apply such fine technique for occlusion handling as we did in [31]. Rough indexing paradigm is not a framework for this. Nevertheless, the objects can be identified by the method of object matching we propose in [32], in the context of rough indexing paradigm constraints such as low resolution and noisy segmentation results.

7. RESULTS AND PERSPECTIVES

The motion and color-based approach with spatiotemporal postprocessing which has been presented in this paper has been tested on different sequences from a set of natural video content. Two types of content have been used: feature documentaries and cartoons; the duration of each sample document was of about 15 minutes.

The temporal segmentation of the video into shots is available in advance. A random set of shots amongst those containing foreground objects is selected.

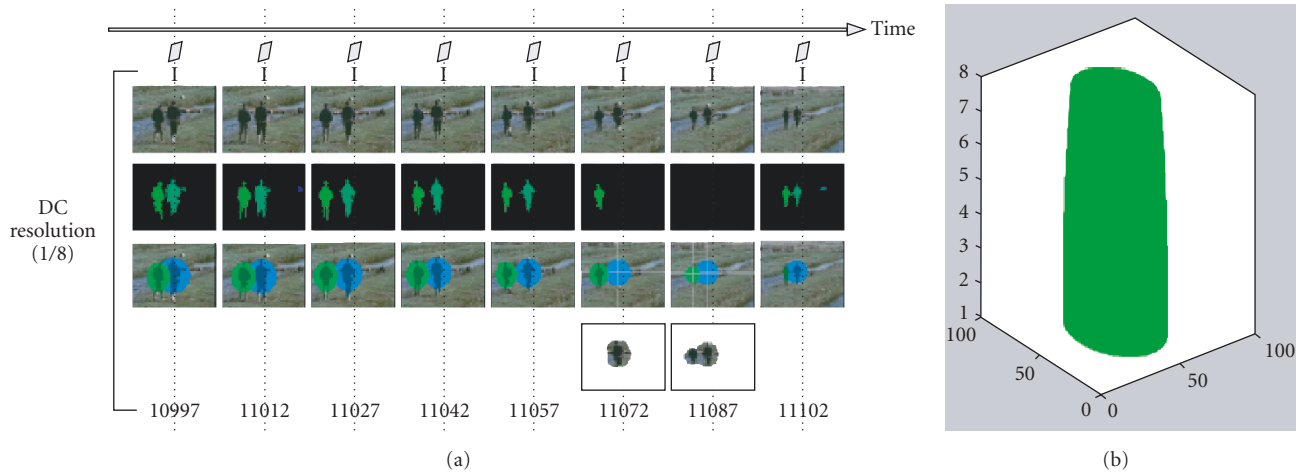


Figure 9: Object approximation by quadric functions: (a) video sequence at I-frame temporal resolution and DC spatial resolution; (b) shape of the quadric functions, SFRS-CERIMES.

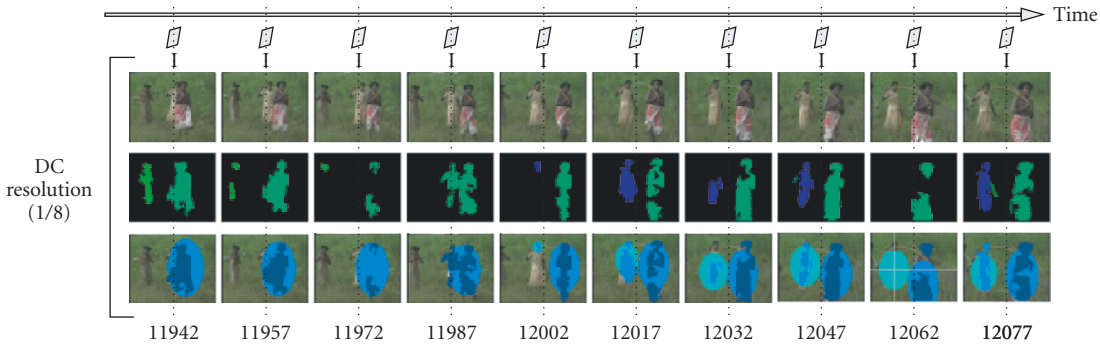


Figure 10: Results on sequence with overlapping objects, SFRS-CERIMES.

As far as camera motion is concerned for the set of processed content, pan, tilt, zoom, and hand-carried camera motion artifacts have been observed. In order to assess the performance of the method, it is obviously very difficult to use precise metrics such as the number of pixel mismatch with respect to a ground truth segmentation. Indeed, the rough indexing paradigm is not designed to achieve a very accurate segmentation of the objects. In this paper, the following user-centered approach has been instead adopted: the object is said to be detected if the user recognizes it as a meaningful moving object in the scene. On the contrary, an overdetection is declared when there are background areas which do not contain any moving object of interest.

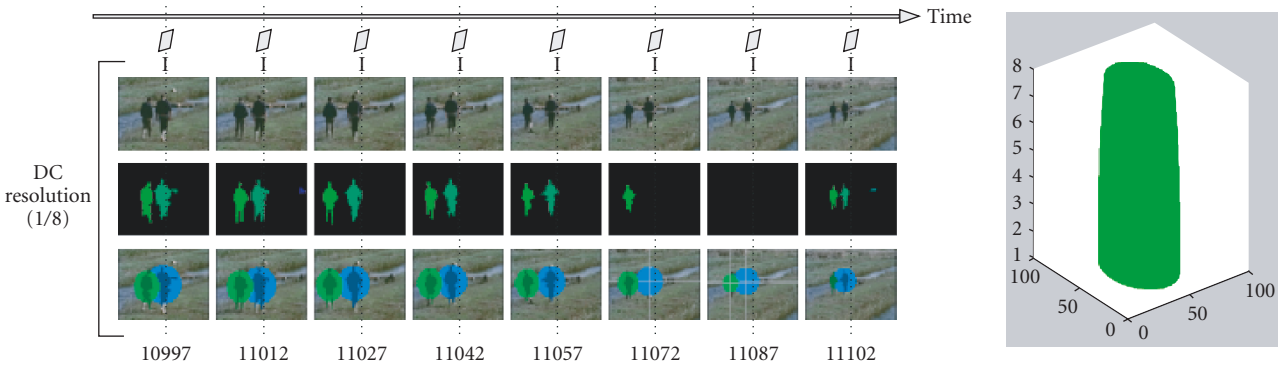
Tables 1 and 2 display the results of our segmentation method for the documentaries and for the cartoons, respectively.

As it can be seen from Tables 1 and 2, the method performs worse for the cartoons with respect to more "natural" video content. The reason for this is that in cartoons, which contain a lot of uniform color areas, the MPEG motion vector fields contain a lot of erroneous values, so that a good motion model cannot be estimated. Outlier motion areas are not appropriate to determine easily moving object areas.

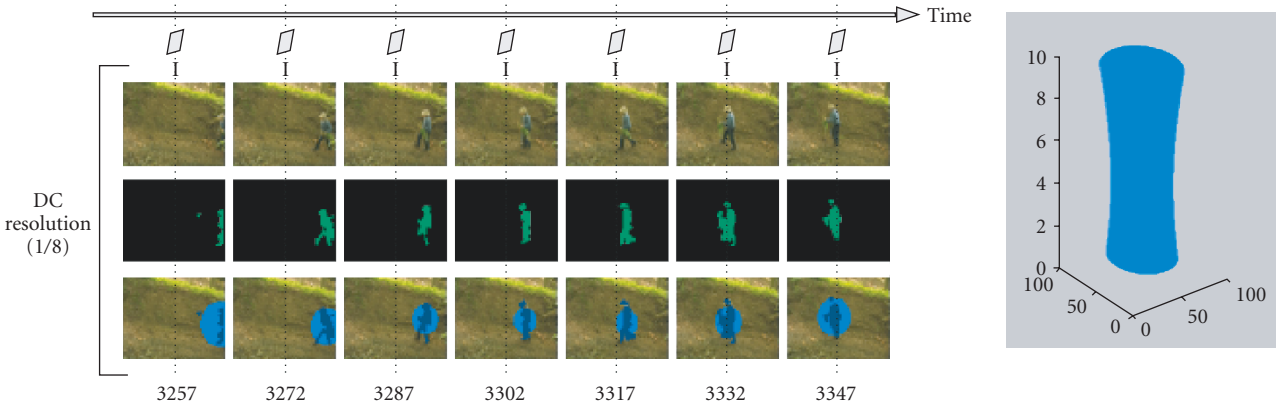
The algorithm has also been tested under limit conditions, that is to say with objects so near to the camera to cover a large part of the background (30%) and in the case of no foreground objects. In the first case, the robustness of the motion detection algorithm has led to a correct extraction of the camera motion parameters and consequently to a correct detection of the object as foreground moving ones. In the second case, even if the noise present on the MPEG motion vectors has caused the presence of macroblocks with high-weight value during the motion estimation, the used postprocessing has allowed to classify the "outliers" as pure noise.

Various results of spatiotemporal filtering of the segmentation masks by quadrics are shown in Figure 11. It can be seen that objects can be tracked along time. A quadric is a locked object along time and it allows to "highlight" it. In case the object disappears, the cross-section of the quadric still allows for a smooth object observation along time. Thus a computation of object descriptors is still possible in the area delimited by the quadric section, allowing for a more accurate indexing process to take place in a second stage.

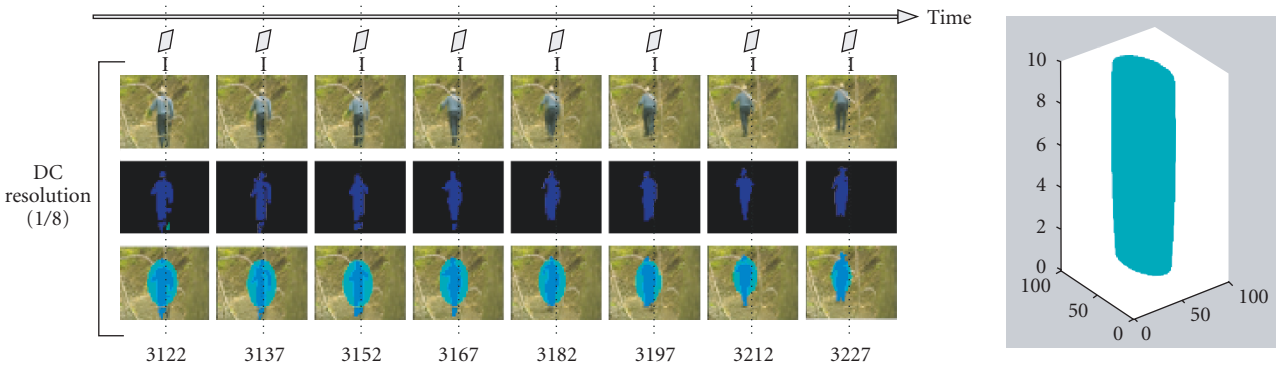
Detailed computational times for the examples of Figure 11 are listed in Table 3. Note that we do not perform the complete decoding process, but only need decoded



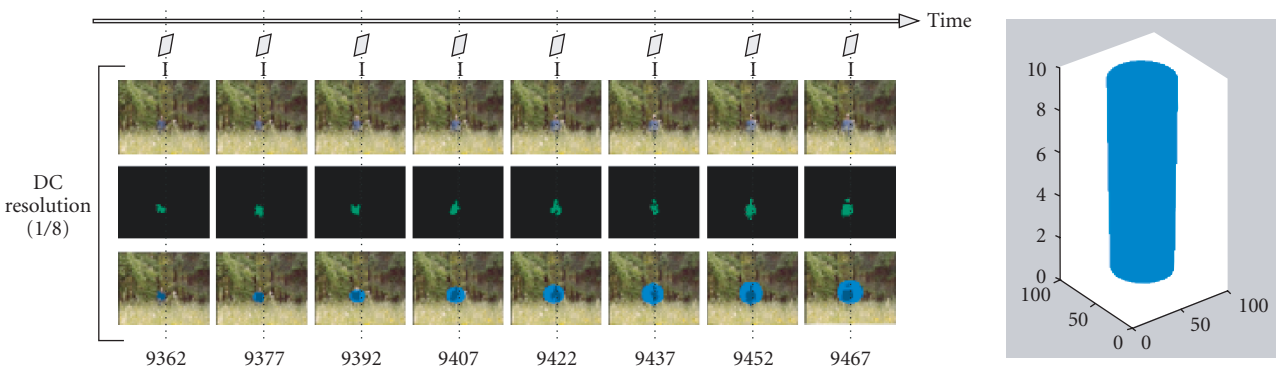
(a) S1



(b) S2



(c) S3



(d) S4

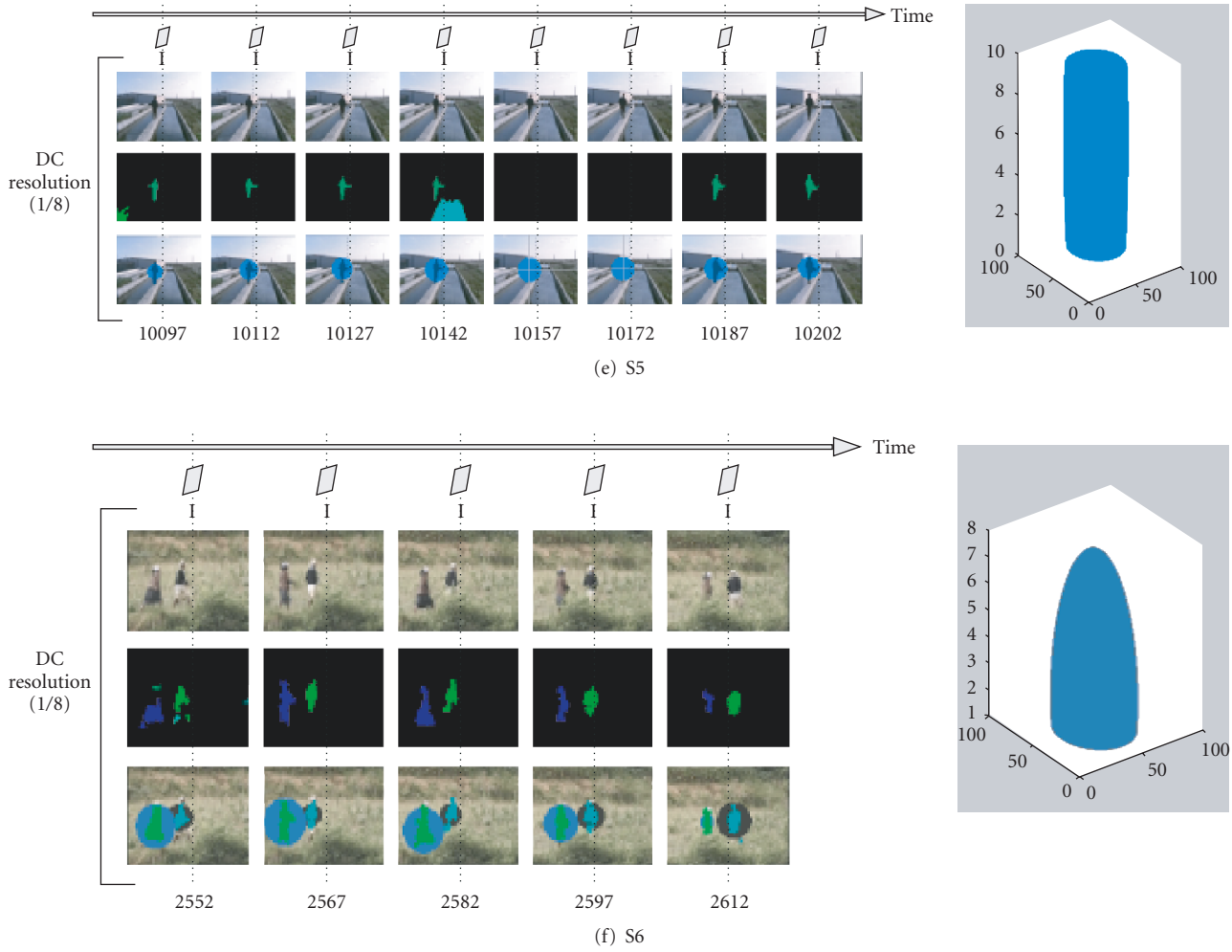


Figure 11: Change the title to Results on the excerpts of documentaries "Comportement alimentaire de homes pre-historiques" (sequences (b)S2, (c)S3, (d)S4), "Aquaculture" (sequences (a)S1, (e)S5), "Hiragasy" (sequence (f)S6), SFRS-CERIMES.

Table 1: Results of object extraction on feature documentaries.

Sequence	% detected objects	Miss detection	Overdetection	% correct frames	Miss detection	Overdetection
Arbre ♯1	48/59 (81%)	10/59	1/59	32/40 (80%)	8/40	0
Arbre ♯2	21/26 (81%)	4/26	0	16/19 (84%)	3/19	0
Arbre ♯3	47/71 (66%)	5/71	19/71	42/63 (67%)	1/63	20/63
Arbre ♯4	6/10 (60%)	2/10	2/10	6/11 (54%)	0	5/11
Arbre ♯5	10/16 (63%)	6/16	0	23/29 (79%)	6/29	0
Arbre ♯6	15/22 (68%)	5/22	2/22	15/22 (68%)	5/22	2/22
Arbre ♯7	22/64 (34%)	42/64	0	15/32 (47%)	17/32	0
Arbre ♯8	53/64 (83%)	11/64	0	22/24 (92%)	2/24	0
Hiragasy ♯1	3/30 (10%)	0	27/30	3/30 (10%)	0	27/30
Hiragasy ♯2	26/26 (100%)	0	0	13/13 (100%)	0	0
Hiragasy ♯3	7/11 (64%)	0	4/11	7/11 (64%)	0	4/11
Hiragasy ♯4	6/8 (75%)	2/8	0	6/8 (75%)	2/8	0
Hiragasy ♯5	8/10 (80%)	0	2/10	4/5 (80%)	0	2/10
Chancre	18/18 (100%)	0	0	9/9 (100%)	0	0
Aqua	30/60 (60%)	24/60	0	14/29 (48%)	15/29	0

Table 2: Result of object extraction on cartoon content.

Sequence	% detected objects	Miss detection	Overdetection	% correct frames	Miss detection	Overdetection
Ferrailles	9/11 (82%)	2/11	0	9/11 (82%)	2/11	0
Escapade	8/10 (80%)	1/10	1/10	8/10 (80%)	1/10	1/10
Boutdumonde	0/12 (0%)	10/12	2/12	0/12 (0%)	10/12	2/12
Casa	11/50 (22%)	35/50	4/50	0/24 (0%)	24/24	0
François	7/26 (27%)	17/26	2/26	7/26 (27%)	17/26	2/26
Bouche	13/20 (65%)	7/20	0	13/20 (65%)	7/20	0
Chat	2/12 (17%)	6/12	4/12	2/12 (17%)	6/12	4/12
Moine	7/26 (27%)	17/26	2/26	2/14 (14%)	10/14	2/14
Roman ♯1	15/50 (30%)	6/50	29/50	15/50 (30%)	6/50	29/50
Roman ♯2	13/24 (54%)	6/24	5/24	13/24 (54%)	6/24	5/24

Table 3: Computation times for the sequences in Figure 11.

Sequence	Number of frames	Movie duration(s)	Motion estimation(s)	Object extraction(s)	Tube construction(s)
S1	27	13.5	1.54	0.43	2.36
S2	7	3.5	0.43	0.12	0.73
S3	13	6.5	0.74	0.21	1.49
S4	11	5.5	0.65	0.18	0.98
S5	23	11.5	0.95	0.35	2.01
S6	5	2.5	0.32	0.08	0.88

motion information for P-frames and DC coefficients for I-frames. This partial stream decoding time has been discarded since it is highly dependent on the efficiency of the used decoder, and it can be in general considered negligible.

8. CONCLUSIONS

In this paper, we have presented a method for foreground object extraction following a "rough indexing" paradigm which allows extraction of foreground objects in MPEG1,2 compressed video at I-frame temporal resolution.

The method performs in near real time and gives promising results. It is clear that, because of the low-resolution data used, a good detection will be obtained only in low-crowding and low-occlusion situation, as in the state-of-the-art in this field anyway.

It can be used with a user-oriented interface for fast visual browsing of video content or as a starting point for object-based indexing of compressed video. The advantage of the method is that it is not limited to a fixed camera. The extension of the method to handled compressed bit-streams of forthcoming Scalable Video Coding (SVC) standards is straight forward, as only the low-frequency information is used and motion information remains available. These are the perspectives of this work from the application point of view. Due to its performance, the method can be used for indexing broadcast or multicast streams at the client side. It can also be used for semantic video adaptation, allowing for intelligent downsampling of video resolution.

REFERENCES

[1] Y. Wang, Z. Liu, and J.-C. Huang, "Multimedia content analysis using both audio and visual clues," *IEEE Signal Processing Magazine*, vol. 17, no. 6, pp. 12–36, 2000.

[2] ISO/IEC JTC1/SC29/WG11 N4030, "Overview of the MPEG-4 standard," V.18, Singapore, March 2001.

[3] ISO/IEC JTC1/SC29/WG11/M6156, "MPEG-7 Multimedia Description Schemes WD (Version 3.1)," Beijing, July 2000.

[4] I. Burnett, F. Pereira, R. Koenen, and R. Van De Walle, Eds., *The MPEG21 Book*, John Wiley & Sons, New York, NY, USA, 2006.

[5] B. Erol and F. Kossentini, "Retrieval of video objects by compressed domain shape features," in *Proceedings of the 7th IEEE International Conference on Electronics, Circuits and Systems (ICECS '00)*, vol. 2, pp. 667–670, Jounieh, Lebanon, December 2000.

[6] P. Salembier and F. Marqués, "Region-based representations of image and video: segmentation tools for multimedia services," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 8, pp. 1147–1169, 1999.

[7] T. Meier and K. N. Ngan, "Video segmentation for content-based coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 8, pp. 1190–1203, 1999.

[8] D. Zhong and S.-F. Chang, "An integrated approach for content-based video object segmentation and retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 8, pp. 1259–1268, 1999.

[9] M. Kim, J. G. Choi, D. Kim, et al., "A VOP generation tool: automatic segmentation of moving objects in image sequences based on spatio-temporal information," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 8, pp. 1216–1226, 1999.

[10] O. Sukmarg and K. R. Rao, "Fast object detection and segmentation in MPEG compressed domain," in *Proceedings of the 10th IEEE Region Annual International Conference (TENCON '00)*, vol. 3, pp. 364–368, Kuala Lumpur, Malaysia, September 2000.

[11] R. V. Babu, K. R. Ramakrishnan, and S. H. Srinivasan, "Video object segmentation: a compressed domain approach," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 4, pp. 462–474, 2004.

[12] R. V. Babu and K. R. Ramakrishnan, "Content-based video retrieval using motion descriptors extracted from compressed domain," in *Proceedings of IEEE International Symposium on Circuits and Systems*, vol. 4, pp. 141–144, Phoenix, Ariz, May 2002.

[13] V. Mezaris, I. Kompatsiaris, N. V. Boulgouris, and M. G. Strintzis, "Real-time compressed-domain spatiotemporal segmentation and ontologies for video indexing and retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 5, pp. 606–621, 2004.

[14] J. Fauqueur and N. Boujemaa, "Region-based retrieval: coarse segmentation with fine color signature," in *Proceedings of IEEE International Conference on Image Processing (ICIP '02)*, vol. 2, pp. 609–612, Rochester, NY, USA, September 2002.

[15] F. Porikli, "Real-time video object segmentation for MPEG encoded video sequences," Tech. Rep., Mitsubishi Electric Research Laboratories, Cambridge, Mass, USA, March 2004.

[16] N. H. AbouGhazaleh and Y. El Gamal, "Compressed video indexing based on object motion," in *Visual Communications and Image Processing*, vol. 4067 of *Proceedings of SPIE*, pp. 986–993, Perth, Australia, June 2000.

[17] Y.-P. Tan, D. D. Saur, S. R. Kulkarni, and P. J. Ramadge, "Rapid estimation of camera motion from compressed video with application to video annotation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 1, pp. 133–146, 2000.

[18] M. Coimbra and M. Davies, "Pedestrian detection using MPEG-2 motion vectors," in *Proceedings of the 4th European Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS '03)*, pp. 164–169, London, UK, April 2003.

[19] M. Durik and J. Benois-Pineau, "Robust motion characterisation for video indexing based on MPEG-2 optical flow," in *Proceedings of the International Workshop on Content Based Multimedia Indexing (CBMI '01)*, pp. 57–64, Brescia, Italy, September 2001.

[20] P. Bouthemy, M. Gelgon, and F. Ganansia, "A unified approach to shot change detection and camera motion characterization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 7, pp. 1030–1044, 1999.

[21] S. Benini, E. Boniotti, R. Leonardi, and A. Signoroni, "Interactive segmentation of biomedical images and volumes using connected operators," in *Proceedings of IEEE International Conference on Image Processing (ICIP '00)*, vol. 3, pp. 452–455, Vancouver, Canada, September 2000.

[22] J. Benois-Pineau and H. Nicolas, "A new method for region-based depth ordering in a video sequence: application to frame interpolation," *Journal of Visual Communication and Image Representation*, vol. 13, no. 3, pp. 363–385, 2002.

[23] B.-L. Yeo and B. Liu, "On the extraction of DC sequence from MPEG compressed video," in *Proceedings of IEEE International Conference on Image Processing (ICIP '95)*, vol. 2, pp. 260–263, Washington, DC, USA, October 1995.

[24] A. Mahboubi, J. Benois-Pineau, and D. Barba, "Suivi et indéxation des objéts dans des séquences vidéo avec la mise-à-jour par confirmation retrograde," in *COmpression et REprésentation des Signaux Audiovisuels (CORESA '01)*, Dijon, France, November 2001.

[25] M. Najim, *Modélisation et identification en traitement du signal*, edited by Masson, chapter 2, 1988.

[26] Ç. E. Erdem, F. Ernst, A. Redert, and E. Hendriks, "Temporal stabilization of video object segmentation for 3D-TV applications," in *Proceedings of IEEE International Conference on Image Processing (ICIP '04)*, vol. 1, pp. 357–360, Singapore, October 2004.

[27] D. Hilbert and S. Chon-Vossen, "The second-order surfaces," in *Geometry and the Imagination*, chapter 3, pp. 12–19, Chelsea, Bronx, NY, USA, 1999.

[28] A. Agnetis, "Introduzione all'ottimizzazione vincolata," University of Siena, Siena, Italy.

[29] F. Manerba, J. Benois-Pineau, and R. Leonardi, "Extraction of foreground objects from a MPEG2 video stream in "rough - indexing" framework," in *The International Society for Optical Engineering, Storage and Retrieval Methods and Applications for Multimedia*, vol. 5307 of *Proceedings of SPIE*, pp. 50–60, San Jose, Calif, USA, January 2004.

[30] L. Vincent and P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 6, pp. 583–598, 1991.

[31] L. Wu, J. Benois-Pineau, Ph. Delagnes, and D. Barba, "Spatio-temporal segmentation of image sequences for object-oriented low bit-rate image coding," *Signal Processing: Image Communication*, vol. 8, no. 6, pp. 513–543, 1996.

[32] F. Chevalier, J.-P. Domenger, J. Benois-Pineau, and M. Delest, "Retrieval of objects in video by similarity based on graph matching," *Pattern Recognition Letters*, vol. 28, no. 8, pp. 939–949, 2007.