MPEG VIDEO OBJECT SEGMENTATION UNDER CAMERA MOTION AND MULTIMODAL BACKGROUNDS

Marcos Escudero, Fabrizio Tiburzi, Jesús Bescós
Grupo de Tratamiento de Imágenes, Escuela Politécnica Superior
Universidad Autónoma de Madrid, E-28049 Madrid, Spain
e-mail: {marcos.escudero, fabricio.tiburzi, j.bescos}@uam.es

ABSTRACT
This paper starts from a state-of-the-art efficient approach to real-time video object segmentation in the MPEG domain. It then describes several techniques to extend the algorithm’s compelling behavior to a more generic set of situations. These are focused on the management of intra-coded macroblocks, the exploitation of objects motion coherence, the integrated use of color information and an approach to discriminate objects from multimodal background (e.g., water, flames), always under camera motion conditions. Results evaluation is presented over a ground truth generated with the help of chroma studio in order to reproduce almost real sequences in a controlled way.

Index Terms— Video segmentation, camera motion, compressed-domain analysis, real-time analysis.

1. INTRODUCTION
Unsupervised video object segmentation plays an essential role for many applications. Existing approaches addressing this task can be categorized according to the homogeneity criteria followed to separate the spatio-temporal coherent regions or video-objects from the background. These criteria are highly dependent on the specific application, being color, texture and motion the most common alternatives. Due to the existing heterogeneity in shape, color or texture for most real-world objects, motion is often preferred when for domain independent approaches.

There are plenty of reports on segmentation of moving-objects in video sequences, and an increasing number of them that confront this problem working directly in compressed domains like MPEG-1/2 or H.264. Despite possible inconveniences of this approach [1], the direct availability of coding oriented analysis data is definitely worthwhile from a computational resources point of view.

In this work we focus on a low-complexity moving-object segmentation algorithm that operates on MPEG sequences showing camera motion. There are many approaches dealing with sequences recorded with static cameras [2][3] yielding convincing results; but this is a rather restricting constraint when it comes to generic videos. Briefly, the most common strategy followed in these reports relies on a pixel intensity model of the background, being deviations from this model an indicator of object presence.

Works considering camera motion are much more scarce. [4] uses both a watershed transformation and motion vector similarity to look for regions undergoing a similar motion different enough from the dominant pattern. However, as it is stated, the approach is not suitable as a stand-alone segmentation system. In [5], the authors create a dense motion field by using motion information of several frames. Then they estimate the number of moving objects by clustering, and their borders using the EM algorithm. Despite being robust in cases of smooth camera motion, the improved motion field can be inaccurate in changes in the camera motion patterns, and unacceptable when dealing with hand-held jerky cameras. Additionally, the complexity of the involved operations makes this method unsuitable for real time applications. In [6], Mezaris et al. present an efficient scheme consisting on simultaneously estimating the dominant motion while discarding the potential objects. Temporal consistency of object regions is reinforced by means of a tracking procedure.

The approach we present in this paper starts from the Mezaris’s technique, which has been extended to cope with a broader range of common situations. The description of the starting approach as well as the proposed strategies to extend it is the topic of section 2. These strategies are detailed in section 3. Section 4 presents a quantitative evaluation and Section 5 concludes the paper.

2. BASE SYSTEM OVERVIEW
The algorithm described in [6] operates on MPEG-1/2 coded sequences and obtains, for every frame, an object segmentation mask with macroblock resolution. For the
sake of simplicity motion information is extracted just from P frames, interpolated to I frames, and the achieved results replicated to B frames. Operation is as follows:

First, an Iterative Rejection (IR) module extracts an initial segmentation mask in which the macroblocks with a significant motion deviation from the global motion are activated. Global motion is obtained by fitting the MPEG motion vectors to a 8-parameter bilinear model with an iterative rejection scheme similar to [7]; the outlier vectors correspond to the activated macroblocks. Intra-coded macroblocks are assumed to belong to potential objects and therefore also included in this initial mask ($M_{IR}$).

In order to improve $M_{IR}$, a Macroblock-Level Tracking (MLT) module tests temporal persistence to generate a new temporally improved mask ($M_{IR}^t$): a macroblock is activated in $M_{IR}^t$ for frame $f_i$ (denoted as $M_{IR}^{t_i}$) if it is activated in $M_{IR}^t$ and can be tracked forward from $M_{IR}^{t_i-1}$ (i.e. significantly overlaps a motion prediction of the active macroblocks of $M_{IR}^{t_i-1}$). To allow the detection of new objects in any frame apart from the first one, this recursive tracking schema is applied in windows of $N_f = 4$ frames.

Finally, an Object Formation module handles the active macroblocks from the $M_{IR}^t$ masks, assigning them either to new or to preexisting objects (according to connectivity, overlapping and motion criteria), hence forming a last set of object masks ($M_o$). This module also manages occlusions and semantic constraints (such as preventing the creation of too small or too fleeting objects).

2.1 Overview of the proposed improvements

We have considered the summarized approach as a convenient base algorithm to confront object segmentation in more generic situations, both because of its coherent functional decomposition and its excellent trade-off between the achieved results and complexity.

Several aspects can be refined to improve the algorithm’s behavior in many ordinary situations, not or partially considered by the authors. This is the case of the simple management of intra-coded macroblocks or the biased scheme to exploit the temporal object predictability, focused on removing wrongly activated macroblocks but disregarding the spuriously missed ones. Moreover, macroblock color information is ignored for motion segmentation, while it can be a great help to reinforce decisions in specific areas like object contours.

Additionally we have devised an innovative technique to handle background multimodality, which separates foreground objects from typical background varying objects (e.g., water, flames). It must be noted however that, as the definition of object is strongly application dependent, this technique must be considered at the same level as the aforementioned semantic constraints imposed to avoid segmenting non-relevant objects.

Figure 1 depicts an overview of the algorithm; proposed improvements are depicted in dark shaded boxes.

3. IMPROVEMENTS DESCRIPTION

3.1 Dealing with intra-coded macroblocks

Since, for efficiency, the initial segmentation stage of the base algorithm exclusively relies on the MPEG motion vectors to assign macroblock motion, intra-coded macroblocks represent a problem. These macroblocks are especially likely in a number of situations: uncovered background (either due to objects or camera motion), noise, emerging objects or changes in the appearance of the existing ones. Ideally, just the last two situations should lead to activation in the $M_{IR}$ mask.

The base algorithm activates all the intra-coded macroblocks in the $M_{IR}$ masks, which ensures that object information is never lost. Additionally it systematically deactivates all the macroblocks lying in the frame borders, thus partially solving the frequent problem of background uncovering due to the camera motion, particularly tricky because the resulting intra-coded macroblocks persist over time, so that their influence would be hardly diminished by the MLT module. This simple solution works well under slow camera motion. However, intra-coded macroblocks due to noise, fast camera motion or uncovered background other than the frame borders are wrongly activated.

We propose to activate an intra-coded macroblock in $M_{IR}^t$ either if it significantly overlaps or is an adjacent neighbor of the activated macroblocks in the tracked mask $M_{IR}^{t-1}$, or if it was not intra-coded in $M_{IR}^{t-1}$. This ensures that intra-coded macroblocks resulting from either new objects coming into the scene or from the change in size or appearance of the existing ones are properly activated. At the same time it prevents activation of time-persisting intra-coded macroblocks not being adjacent to
any potential object (like those due to noise or to background uncovering in the frame borders).

Intra-coded macroblocks due to uncovered background emerging after object motion are still pending, but their effect is an object trail that can be easily corrected using basic color information (see section 3.3). In any case, as they are not likely to remain being intra-coded for more than one or two frames, their influence would be readily mitigated after a short time by the MLT module.

3.2 Further exploiting the temporal coherence of moving objects

The $M_{IR}$ masks usually include many inconsistencies, which correspond either to activated macroblocks not belonging to moving objects, or to non-activated macroblocks being part of moving objects. These are respectively known as type I and type II errors in statistics.

Type I errors can be due to the inaccuracy of the motion vectors when describing real motion, to background noise or to new background being uncovered after object motion. As these macroblocks are not persistent, they are effectively tackled (see Section 2) by the MLT module.

Type II errors are likely to occur when the macroblock motion matches the dominant motion despite being part of a moving object. This arises when dealing with non-rigid objects and some of their parts become temporally motionless (e.g., the alternatively static feet of a walking person while the rest of the body is propelled forward), when objects are just changing their trajectory, or when their relative velocity becomes negligible respect to the dominant motion.

The base algorithm presented in Section 2 does not consider type II errors, which produces annoying results, like incomplete moving objects or objects that lose their connectivity. Additionally, due to the MLT Module policy these errors have a multiplicative effect: after erroneously deactivating a macroblock, at least $N_r$ frames are required to make it reappear.

We have confronted both type of errors with a slow-vanishing strategy that exploits the temporal coherence of the moving objects. We generate a new prediction mask from $M_{IR}$, so that activated macroblocks in this new mask are kept activated within the subsequent $M'_{IR}(k < r \leq k + N_y)$, independently of their activation state in the corresponding $M_{IR}$. However, if a given macroblock remains deactivated in $N_r$ iterative rejection masks starting from $M'_{IR}$, it is definitely deactivated in $M'_{IR}^{N_r}$.

The use of this macroblock level temporal window is prone to the formation of a trail around the moving objects. Therefore, the window size $N_r$ must be carefully chosen as a trade-off between achieving an accurate object mask and minimizing the trail. $N_r = 2$ achieves good results, as the emerging trails are small enough to be easily discriminated and removed by means of simple color procedures.

3.3 Using the color information

Color information can be efficiently extracted from estimations of the DC coefficients in P frames, and it is an important cue to reduce the fraction of misclassified macroblocks when motion information is not enough. We have used it to refine the final $M_o$ in two stages:

1) Reclassification of suspicious macroblocks lying on the background. This deals with type II errors: background macroblocks immediately adjacent to the object borders as well as their contiguous neighbors are re-examined.

2) Reclassification of suspicious macroblocks lying on the foreground. This deals with type I errors: foreground macroblocks that were explicitly activated by the slow-vanishing strategy are inspected.

Macroblock reclassification is based on a K-Nearest-Neighbors classifier (K=3 has proved to yield good results). Macroblock feature vectors have been built with the three components obtained by averaging the YUV values of the DC coefficients of its four DCT blocks. Class samples for the background and foreground classes have been obtained for every classified macroblock from the macroblocks in its neighborhood within a radius ranging from 1 to 3 macroblocks. Note that the foreground macroblocks activated by the slow-vanishing strategy must be discarded for this process, as they could be object trails.

3.4 Dealing with background multimodality

Our aim is to discard moving regions showing characteristics of objects typically lying on the background (twinkling water, swaying trees, glowing flames...) or persisting noise normally caused by unstable illumination conditions. This enhancement endows our system with the functionality of the traditional mixture models used in static scenes, while keeping the ability to cope with the more generic case of scenes with camera motion.

Considering the chaotic motion arrangement and the usual non-rigid structure characteristic of these background objects, they can be expected to yield higher fitting error when trying to describe their motion with a parametric model. Additionally, the obtained motion parameters are likely to show a greater variation when observed over several consecutive frames. These two hypothesis have guided the design of discriminative feature vectors for every new object $O$ emerging in the $M_o$ masks:

Since there exists valuable motion information of these objects in the $N_f$ previous $M_{IR}$ masks, for each frame $k = 1..N_f$ of this temporal window we first fit $O$’s motion vectors to a bilinear model, to obtain the set of parameters $d_k^f (i = 1..8)$. Then we obtain the median $e^k$ and the standard deviation $s^k$ of the fitting errors for each

---

2670
motion vector, and we finally define the feature vector
\[ V^0 = [\mu(e^x), \sigma(e^x), \mu(s^x), \sigma(s^x), \sigma(a^1)...\sigma(a^k)] \].

We have selected a Fisher classifier to separate foreground from background objects, and previously estimated the classification accuracy and trained the classifier (via the leave-one-out cross-validation technique) with a representative set of manually annotated foreground and background objects (water, fire, smoke, vegetation...). Table 1 summarizes the achieved results after evaluation.

<table>
<thead>
<tr>
<th>Object detection</th>
<th>Noise detection</th>
</tr>
</thead>
<tbody>
<tr>
<td>95.10%</td>
<td>24.48%</td>
</tr>
<tr>
<td>86.99%</td>
<td>39.84%</td>
</tr>
</tbody>
</table>

Table 1: Achieved detection accuracy (%) on foreground objects vs. background objects (noise) for two settings of the classifier.

4. RESULTS AND EVALUATION

This section presents our ongoing efforts on objective evaluation and a quantitative comparison of the improvements achieved respect to the base algorithm.

4.1 Segmentation ground-truth

We have designed a set of scripts consisting of a static camera scene involving interacting people and other moving objects. Objects motion has been designed to obtain a representative set of the situations that should take into account any segmentation algorithm. These scripts have been performed and filmed in a chroma studio to obtain high quality video sequences (720x576, 4:2:2, 25 fps., progressive, raw), so that an accurate segmentation has been obtained. Then, we applied chroma-key techniques to add selected backgrounds in order to simulate desired situations that we may find in a real video. A detailed description of these sequences can be found in [8].

4.2 Comparative results

First we have used two of the aforementioned sequences with a simple background to show comparisons (see Table 2 and Figure 2) between the base algorithm and the improvements described in Sections 3.1 to 3.3.

<table>
<thead>
<tr>
<th>Base algorithm</th>
<th>Proposed algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seq. 1</td>
<td>R: 88.3, P: 40.4, R*: 35.7, P: 80.7, R*: 55.5, P: 44.8</td>
</tr>
<tr>
<td>Seq. 2</td>
<td>R: 48.8, P: 62.7, R*: 30.6, P: 49.9, R*: 75.5, P: 37.7</td>
</tr>
</tbody>
</table>

Table 2: Percentage of achieved recall (R), precision (P) and their product in the detection of macroblock objects vs. background objects for two video sequences (Figure 1 top (1) and middle (2)).

Then we have tested a sequence including a multimodal background to qualitatively evaluate the functionality described in Section 3.4 (Figure 2, third row).

5. CONCLUSIONS

This work describes improvements to a State of Art video object segmentation algorithm, which we have considered excellent as a starting point, and an innovative approach to consider multimodal background in sequences with camera motion. Finally, it presents results under an ongoing rigorous evaluation method that supports the reported achievements.

REFERENCES