HIGH AND LOW LEVEL OBJECT DESCRIPTIONS FOR VIDEO TRACKING PROCESS

David Izquierdo and Yannick Berthoumieu
Laboratoire IXL UMR 5818, ENSEIRB
1, Avenue du docteur Albert Schweitzer
BP 99, 33402 Talence Cedex, France
Tel: +33 5 56 84 65 00; fax: +33 5 56 37 20 23
e-mail : {izquierd, Yannick.Berthoumieu}@enseirb.fr

ABSTRACT

In this paper a new segmentation algorithm approach for real time traffic scenes is proposed, combining high level and low level object descriptions. Both descriptions make it possible to develop a tracking method, robust regarding occlusions, region clustering and brightness variations. High level description is defined by geometric attributes and motion model. Updating these features (associated to each object) can be obtained by a low level segmentation which is based on a background update approach, associated with a spatial-temporal segmentation. This spatial-temporal segmentation is built on a motion estimation taken out from a modified Expectation-Maximization (EM) method. These two descriptions leads to a really efficient strategy in terms of robustness, over or sub-segmentations and occlusions. Furthermore, under severe brightness changes, our new temporal algorithm also permits a perfect background update control. Some real traffic examples are included at the end of this paper.

1 INTRODUCTION

Over the last decade, many methods have been developed in the framework of video segmentation. Some of them are dedicated to motion tracking and particularly to road traffic. Two kinds of approaches are proposed in literature. On the one hand, the authors propose techniques based on the segmentation of the primitives related to the Apparent Movement (AM) [1], [2] and [3]. These techniques use a statistical formulation of the problem through a Markov Random Field exploiting a spatial-temporal neighbourhood model or a likelihood function minimization by means of an Expectation-Maximization (EM) approach. On the other hand, in the second type of methods, changes building on an Adaptive Reference Image (ARI) are extracted [4] and [5]. Generating a reference image containing only the background information, i.e. the static parts of the scene, permits a fast extraction of the moving objects. According to these segmentation classes, this paper describes a complete strategy in order to realise the object recognition and motion tracking. Our approach is driven by a high level description based on object model characteristics. The feature space spanned to this set of descriptors permits introducing matching rules and provides an efficient match process. Updating in the feature space depends on the basic segmentation, extracted by a mixture of ARI and AM approaches. In order to increase the performance of our system, we have introduced a new ARI method that was less sensitive to the photometric distortions due to natural atmospheric conditions.

This paper is organized as follows. Section 2 describes the temporal segmentation algorithm. In section 3, we explain the tracking process which involves Objects Attributes (OA) and EM algorithm. Section 4 presents some experimental results based on real traffic scenes. We conclude in section 5.

2 TEMPORAL SEGMENTATION BASED ON A NOVEL ARI APPROACH

Change detection and motion segmentation are a fundamental task for all kinds of automatic video surveillance systems. Before any tracking step, a possible initial stage consists in reducing the observation by taking out the binary mask of compact regions as a result of temporal segmentation.

A wide variety of existing techniques is proposed in the computer vision literature [5] and [6]. One kind of method is that related to an Adaptive Reference Image (ARI) [4]. Commonly, these techniques are based on a general form, given by:

$$B^{k+1} = \alpha^{k} B^{k} + \left(1 - \alpha^{k}\right) I^{k} \qquad (1)$$

where *B* represents the reference image, *I* the current image and α the memory of the system, or the system capacity to respond to a change. Nevertheless, this update procedure is inefficient behind background luminance changes. These kinds of changes could arise from natural effects such as cloud passages, sunrises or sunsets. Figures 1 and 2 illustrate different cases. We can observe the luminance profile characteristics associated to the background and object crossings for one pixel. A resume is described in Tab. 1:

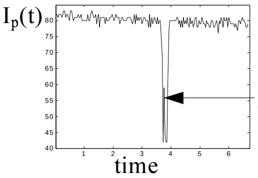

Fig. 1: Constant mean level input image.

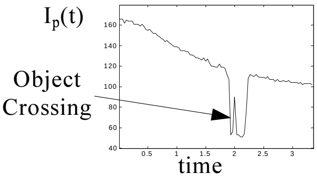

Fig. 2: Background luminance change image.

	Quasi-linear variation	Mean luminance
Background	Slow	Constant level
Object	Fast	Strong variability

Tab. 1: Profile characteristics.

In order to detect every object passage and to avoid the influence of natural effects, we propose a new framework, presenting a robust reference update scheme, based on:

- ***Object crossing arbitrator***: it is in charge of deciding when object crossings are produced. By exploiting the difference between background and object luminance characteristics, a new derivate formulation has been extracted, based on object crossing properties: *strong variability*, shown by a second order derivate and *fast quasi-linear variation*, extracted by first order derivates. This new formulation mixes these first and second order derivate properties, being able to detect both the effects occurred in object passages. This point is developed in section 2.1.
- ***Two different auxiliary references*** associated to the luminance signal slope, B_1^k and B_2^k, are included into the framework, helping us to track the image variations. The first one, B_1^k, is in charge of maintaining the latest luminance value before object crossing (figure 1). And the second one, B_2^k, is built on a slope prediction scheme (figure 2). Section 2.2 explains this update procedure.
- ***State machine*** controlling the whole update process (see section 2.3).

2.1 Derivative Approach

The development of an efficient decision rule behind background luminance change needs to introduce a forward vision. This non-causal observation allows to extract the first and second derivates associated to linear and non-linear variations. The extraction of both effects is based on an operator of "maximum":

$$d^k = \max\left(\left|d_1^k\right|, \left|d_2^k\right|\right) \qquad (2)$$

The forward vision produced by this derivate operator is defined by the length of the first and second derivates respectively d_1^k et d_2^k.

Deciding if an object crossing is being produced over a pixel $p=(x, y)$, involves derivate operator thresholding:

$$D^k(p) = \begin{cases} 1 & \text{if } \ d^k(p) \leq T^k(p) \\ 0 & \text{if not} \end{cases} \qquad (3)$$

The quantity $T^k(p)$ is an adaptive threshold defined by the product of a constant c and the second moment associated to d^k, which is updated following:

$$T^k(p) = c \cdot \sqrt{m_2^k(p)} \ \text{ in } \ m_2^{k+1} = m_2^k \beta + (1-\beta)(d^k)^2 \qquad (4)$$

where β permits a selective update, defined by α, the update memory, and D^k:

$$\beta = 1 - \alpha D^k \qquad (5)$$

2.2 Reference Update

Taking into account the natural luminance evolution in the reference image suppose a knowledge of the input image variations. These variations may be completely defined by their slope identification. Two different auxiliary references, associated to this luminance signal slope, are extracted from a recursive scheme. This scheme is developed according to a state machine (see section 2.3). In this scheme, when no passages are detected, the reference pixel location is always updated with the current pixel value. On the contrary, if object crossings are detected, luminance conditions must be regarded.

With this purpose, under a constant mean level pixel value, if an object crossing is observed (see Fig. 1), the reference must be updated with a zero slope value. B_1^k, is the recursive reference function that maintains the latest luminance value found in the reference image, before detecting the object passage:

$$\begin{aligned} B_1^k = &\left(\frac{(1-s^k)(2-s^k)}{2}\right)\left(I^k(1-D_1^k) + B_1^{k-1}D_1^k\right) \\ &+ \left(\frac{3s^k - (s^k)^2}{2}\right)B_1^{k-1} \end{aligned} \qquad (6)$$

s^k represents the state value (see section 2.3).

Under luminance variation, if an object crossing is detected (see Fig. 2), reference must be updated according to a slope prediction, L^k. L^k is updated with the same basic scheme like in (1):

$$L^k = L^{k-1}\rho + (1-\rho)(B_2^k - B_2^{k-1}) \qquad (7)$$

where ρ represents the slope memory and B_2, the second reference function. The associated recursive equation could be written as:

$$\begin{aligned} B_2^k = &\left(\frac{(1-s^k)(2-s^k)}{2}\right)\left(I^k(1-D_1^k) + B_2^{k-1}D_1^k\right) \\ &+ \left(\frac{3s^k - (s^k)^2}{2}\right)\left(B_2^{k-1} + L^{k-1}\right) \end{aligned} \qquad (8)$$

The final decision between these two different update references is made according to the error minimization with the current image values:

$$B^k = \min_{B_i}\left(\left|B_i^k - I^k\right|\right) \text{for } i = 1, 2 \qquad (9)$$

2.3 State Machine

Separate states have been created, because of decisions concerning the update of the reference are object-crossings dependent. A state machine handling each pixel state evolution is carried out.

A state machine consists on a particular decisional algorithm, exploiting the idea that same inputs could cause different outputs, depending on the system situations, denominated *states*. Therefore, outputs depend not only on inputs but on the system state.

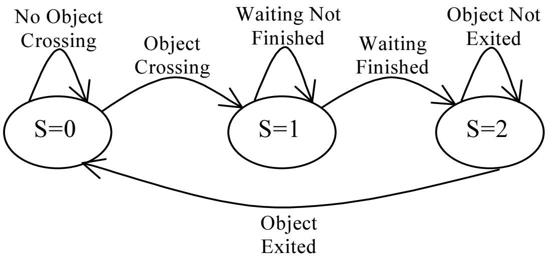

Fig. 3: State Diagram

The figure above represents the cyclical sequence of every pixel state value in the image controlled by the derivate function, d^k. s^k represents the pixel state and takes values {0, 1, 2} for the whole set of states.

Each state is defined as follows:

- $s^k = 0 \Rightarrow$ "Rest State": this is the state of those pixels that has had no lighting changes due to object crossings. Transitions to '1' state are produced when object passages: $D^k(p) = 0$ at pixel p: $s^{k+1}(p) = 1 - D^k(p)$

- $s^k = 1 \Rightarrow$ "Wait State": this is a transitional state. We just wait for a constant number of images before arriving at the last state. "Wait state" is introduced in order not to take fast and wrong decisions concerning the total object crossing at pixel *p* which might produce false reference updates.

- $s^k = 2 \Rightarrow$ "Crossing State": in this state, we must take the decision if an the object is totally past through the pixel *p* or not.

2.4 ARI Process

The complete scheme of the temporal segmentation, including the reference update and the state machine is represented by the Fig. 4.

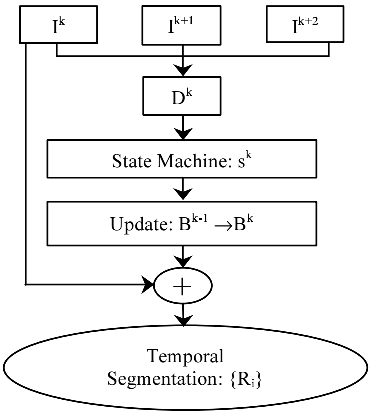

Fig. 4: Complete temporal algorithm block scheme

In this representation, the derivate approach of our implementation is based on only three consecutive images. Let $\mathbf{R}=\{R_i\}_{i=1,..,r}$ be the set of *r* connected regions for the kth image. This set of regions characterizes the initialisation of the tracking process developed in the following part.

3 TRACKING PROCESS

According to the temporal segmentation, the tracking procedure now introduces a high level description of objects. High level description defines each object present in the kth image via a model, M_j^k. Let $\mathbf{M}=\{M_j\}_{j=1,..,q}$ be the set of *q* objects' models at the kth image. Each M_j^k is composed by a set of Object Attributes (OA) allowing to identify the object in the next image through the set of regions R_i^{k+1}. This identification between regions and models is performed by a *Matching Process*. The matching process takes every region, R_i^{k+1}, and compares it with every predicted model $\widetilde{M}_j^{k+1}$, extracted from M_j^k via motion compensation: $\widetilde{M}_j^{k+1} = M_j^{k}(\theta_j^k)$ where θ_j^k represents the motion parameters corresponding to the model M_j^k at the kth image.

An EM algorithm is started up, solving every occlusion and sub-segmentation problem and updating every OA of the model.

3.1 Matching Process

The matching process is performed by a special pairing of Regions-Models, $[R_i^{k+1}; M_j^k]$. This coupling is available according to a feature space based on OA. OA's are based on geometrical parameters and are defined by:

- *form factor*: this attribute is provided by the convex polygon generated by the merging of holes and regions given by the spatial-temporal segmentation.

- *barycentre* of the model.

- *state factor*: this state is associated to each object following the position of the object in the scene. This attribute permits to supervise the updating of the form factor and barycentre. This feature can take three values, {entering, exiting or alive} for the space feature, and also shows whether object occlusion is occurred.

- *motion model*: it is used to warp the object model M_j^k in order to obtain the predicted model $\widetilde{M}_j^{k+1}$.

From previous consideration, the match test framework takes every R_i^{k+1} region and compares it with every predicted model $\widetilde{M}_j^{k+1}$, extracted after motion compensation of M_j^k.

3.2 EM Algorithm

The Expectation-Maximization algorithm extracts a spatial segmentation of an image based on a procedure of mixture of classes [3]. In this procedure, each class of the mixture is associated to a motion model present in the frame. In this way every pixel has different probabilities for every different class of the mixture and the spatial segmentation is obtained associating each pixel to the class in which it has largest probability.

From this idea, we have modified the EM procedure according to the necessities of our process. On the one hand, the spatial-temporal segmentation based on the EM approach is applied to each region R_i^{k+1}. On the other hand, the classes of the mixture are chosen from the models M_j^k matching the region. By this selectivity, the sensitivity of the EM algorithm behind the selection of the number of classes is avoided. Occlusion and sub-segmentation problems are solved by this segmentation as a result of the EM procedure.

Models' attributes are updated in the M step of the EM process by a LMS algorithm [7]. In order to avoid common problems found in LMS, due to large displacements, we extract the new motion parameters according to the kth image I^k and the warped version of the (k-1)th image. The warping is according to the latest computed motion parameters:

$$\tilde{I}^k = I^{k-1}(\theta^{k-1}) \qquad (10)$$

so, the new motion parameters are:

$$\theta^k = \theta^{k-1} + \tilde{\theta}^k \qquad (11)$$

4 RESULTS

In order to test the complete algorithm in presence of luminance changes, we have chosen a real road traffic scene involving cloud passages. Figures 5 to 10 show the process evolution. Fig. 5 presents the mask resulting from the temporal segmentation, including a sub-segmentation case where the object 2 is divided in two different regions. Fig. 6 contains the three connected regions, R_i^{k+1} (in solid lines) and the two latest models, M_j^k, matching the regions (in dotted lines) and merging the regions of object 2. Fig. 7 shows the real image, I^{k+1}, with the updated models superimposed, M_j^{k+1}. Fig. 8 presents an occlusion case with two objects overlapped, forming a unique region. Fig. 9 contains this region, R_i^{k+1} and the two matching models, M_j^k. These models breaks down the region into two different parts, via the EM algorithm. Finally, Fig. 10 shows the real image with the two updated models, M_j^{k+1} separating the different objects.

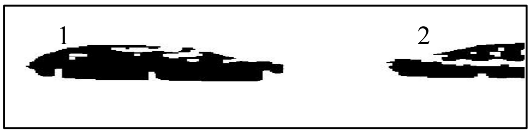

Fig. 5: Temporal Segmentation results.

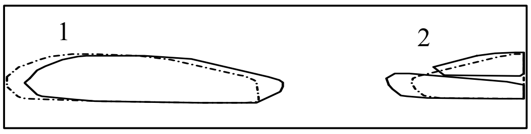

Fig. 6: Normal case (1) and sub-segmentation case (2). Models M_j^k (dotted lines) and regions R_i^{k+1} (solid lines).

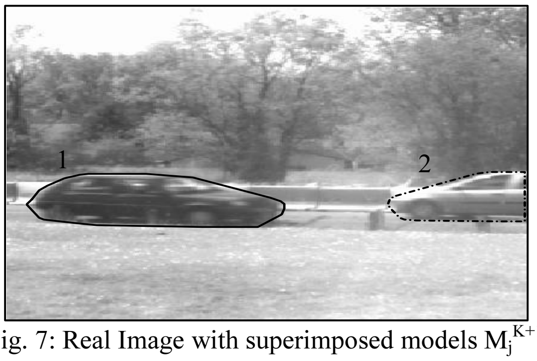

Fig. 7: Real Image with superimposed models M_j^{K+1}.

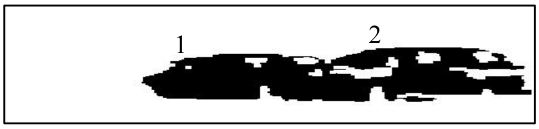

Fig.8: Temporal Segmentation results

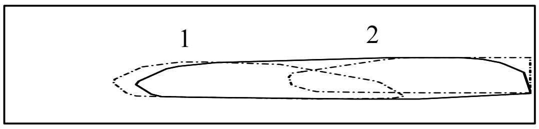

Fig. 9: Occlusion case: 2 Models (dotted lines) and 1 Region (solid line).

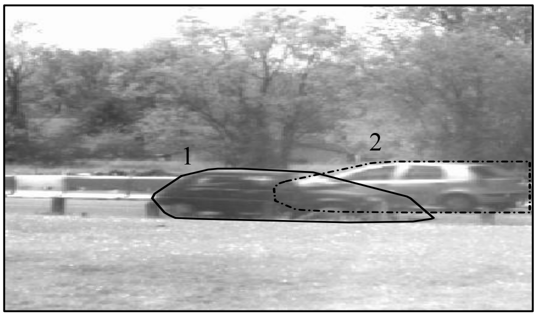

Fig. 10: Real image with superimposed models.

5 CONCLUSIONS

We have proposed an approach for video-segmentation of traffic sequences using a combination of unsupervised and supervised motion segmentation. The supervised segmentation is based on the similarity of object attributes in an optimal procedure resolved by an EM approach. Moreover, this previous step is initialised by an unsupervised motion segmentation, developed by a new method in the framework of ARI process. This local derivative method behaves well against background luminance changes due to natural atmospheric conditions.

References

[1] J.M. Odobez and P. Bouthemy, *Direct incremental model-based image motion segmentation analysis for video analysis*, Signal Processing, vol.66, pp. 143-155, 1998.

[2] C. Dumontier, F. Luthon, J.P. Charras, *Real Time DSP Implementation for MRF-Based Video Motion Detection, IEEE Transactions on Image Processing*, vol.8 n°10, pp.1341-1347, October 1999.

[3] H. Sawhney and S. Ayer, *Compact representation of videos through dominant and multiple motion estimation.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 18(8), August 1996.

[4] D. Koller, J. Weber and J. Malik, *Robust Multiple Car Tracking with Occlusion Reasoning*, 3rd European Conference on Computer Vision, ECCV '94, Stockholm Sweden, pp. 189-196, May 1994.

[5] C. Stauffer and W.E.L. Grimson, *Adaptative background mixture models for real-time tracking CVPR99,* Fort Colins, CO, (June 1999)

[6] I. Grinias and G. Tziritas, *Motion segmentation and tracking using a seeded region growing method*, EUSIPCO 98, Rhodes, Greece, September 1998.

[7] A. Randriantsoa, Y. Berthoumieu, *Optical Flow Estimation Using Forward-backward Constraint Equation*,ICIP 00, TP03.08, 2000.