

Learning Contextual Variations for Video Segmentation

Vincent Martin and Monique Thonnat

INRIA Sophia Antipolis, PULSAR project-team
2004 route des lucioles, BP 93
F-06902 Sophia Antipolis

{Vincent.R.Martin, Monique.Thonnat}@sophia.inria.fr,
<http://www-sop.inria.fr/pulsar/>

Abstract. This paper deals with video segmentation in vision systems. We focus on the maintenance of background models in long-term videos of changing environment which is still a real challenge in video surveillance. We propose an original weakly supervised method for learning contextual variations in videos. Our approach uses a clustering algorithm to automatically identify different contexts based on image content analysis. Then, state-of-the-art video segmentation algorithms (e.g. codebook, MoG) are trained on each cluster. The goal is to achieve a dynamic selection of background models. We have experimented our approach on a long video sequence (24 hours). The presented results show the segmentation improvement of our approach compared to codebook and MoG.

keywords: video segmentation, weakly supervised learning, context awareness, video surveillance, cognitive vision

1 Introduction

Figure-ground segmentation consists in separating the foreground pixels of the background pixels. In video applications, the variability of the two classes makes the detection of foreground pixels fairly impossible to predict without motion information. A widely used method to tackle this problem is to model the background in order to detect only moving pixels. In this paper, we consider the problem of the figure-ground segmentation task in video surveillance applications where both quick-illumination changes and long term changes are present. In this situation, the major difficulty at the segmentation level is to deliver robust results whatever lighting changes occur in the scene. These lighting effects can be due to weather conditions changes in outdoor scenes, to the switching of an artificial lighting source in indoor scenes, or to a combination of changes of different natures. The consequences at the pixel level are variations of intensity, color saturation, or inter-pixel contrast. At the image level, these changes can affect just a local area or the whole image. Another source of problems arises from the presence of non-static objects in the background as swaying trees or mobile objects as chairs.



Our objective is to cope with all these issues with a cognitive vision approach by endowing video segmentation methods with learning and adaptation faculties. To this end, we first relate some work dealing with these issues then present our learning-based approach for dynamic selection of background model. Finally, we show the effectiveness of our approach on a difficult video sequence.

2 Related Work

A basic approach to estimate the motion is to compute the difference between a background image, called the reference image, and the current frame. The result is then thresholded to get a binary image of moving pixels. The result is obviously very sensitive to the threshold. Most of the time, the user must tune this threshold in a trial-and-error process. One difficulty arises when the background pixels are varying along the time. In this case, more elaborated approaches build a background model for each pixel based on the pixel's recent history by using, for instance a chronological average or median of the n previous frames [1]. Parametric models as Mixture of Gaussian (MoG) [2], Kernel Density Estimator (KDE) [3], and codebooks [4] have been proposed to cope with multiple modal background distributions. These algorithms are based on a training stage to estimate the Gaussian parameters (for MoG), to compute the probability density functions (for KDE), or to construct the codebooks. The training data are composed of background samples, i.e. a set of frames without any moving objects of interest. The training stage of the Mog model consists in estimating k Gaussian parameters set (ω, μ, Σ) for each pixel using an expectation-minimization algorithm, where k is the number of gaussians in the mixture. For the codebook model, the learning stage consists in constructing the set of codewords (i.e. a codebook) for each pixel. A codework is composed of a vector of mean RGB values and of a five-tuple vector containing intensity (brightness) minimum and maximum values, the frequency with which the codeword has occurred with its first and last access time. Each of these techniques can provide acceptable accuracy in specific applications: MoG are adapted to multi-modal background distributions but fail to provide sensitive detection when background has fast variations. KDE overcomes this problem but are limited to short-term videos due mostly to memory constraints. Codebooks alleviate this computation limitation by constructing a highly compressed background model but produce too wide background models when the environment is highly variable as in long-term videos. We propose to add to these algorithms a learning-based layer. We will compare the performance of our approach with codebooks and MoG.

3 Proposed Approach

Our approach is based on a preliminary (off-line) weakly supervised learning module (see Figure 1) during which the knowledge of the context variations is acquired. In our approach we suppose that: (1) the video camera is fixed and (2) background images are available as training samples. We define the context of an

image as the numerical representation of its local and global characteristics. We call this approach weakly supervised because the role of the user is restricted to establish a training image set composed of background samples that point out context variations, e.g. the different illuminations changes that could be encountered in real-time use. In a practical point of view, the collection can be achieved by a manual selection of frame sequences. These assumptions fit quite good with the targeted applications where videos can be acquired continuously, typically 24 hours per day and seven days per week. The quick availability of data allows to build huge training image set.

We tackle the context modelling problem by performing an unsupervised clustering of the training images. The goal is to make the background modelling problem more reliable by restricting the model parameter space. This approach is particularly interesting for motion segmentation algorithms relying on a training stage of models as mixture of Gaussian [2] or codebook [4]. The clustering is based on the analysis of global image characteristics like color variations. At the end of the clustering process, each cluster gathers training images sharing similar global features, i.e. images of the same context.

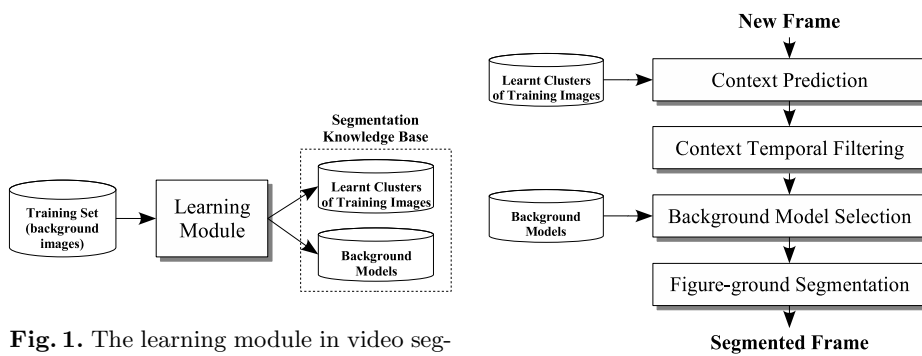


Fig. 1. The learning module in video segmentation task.

Fig. 2. Adaptive figure-ground segmentation schema based on context identification and background model selection.

3.1 Context Analysis by Image Sequence Clustering

The fixed field of view of the video camera allows to analyze the image variations both globally and locally. To this end, a straightforward approach consists in a global histogramming of pixel intensity as in [5]. However, this technique is not fully adapted. Actually, classic histograms lack spatial information, and images with different appearances can have similar histograms. To overcome this limitation, we use an histogram-based method that incorporates spatial information [6]. This approach consists in building a coherent color histogram based

on pixel membership to large similarly-colored regions. For instance, an image presenting red pixels forming a single coherent region will have a color coherence histogram with a peak at the level of red color. An image with the same quantity of red pixels but widely scattered, will not have this peak. This is particularly significant for outdoor scene with changing lighting conditions due to the sun rotation, as in Figure 3.

In our experiment, we have used a Density-Based Spatial clustering algorithm called DBScan [7] to identify the image clusters. This algorithm is well-adapted for clustering noisy data of arbitrary shape in high-dimensional space as histograms. Starting from one point of the data set, the algorithm searches for similar points in its neighborhood based on a density criteria to manage noisy data. Non clustered points are considered as ‘noise’ points. The runtime of the algorithm is of the order $O(n \log n)$ with n the dimension of the input space. DBScan requires only one critical input parameter, the *Eps*-neighborhood, and supports the user in determining an appropriate value for it. A low value will raise to many small clusters and may also classify a lot of points as noisy points, a high value prevents from noisy point detection but produces few clusters. A good value would be the density of the least dense cluster. However, it is very hard to get this information on advance. Normally one does not know the distribution of the points in the space. If no cluster is found, all points are marked as noise. In our approach, we set this parameter so as to have at the most 15% of the training images classified as ‘noise’ data.

Then, for each identified cluster, the corresponding training frames are put together and used to train a background model (the codebooks for instance). Internal knowledge of the DBScan algorithm as the tree nodes and elements are also stored for further classifications of new images. So, to each cluster of training image corresponds a trained background model. The next step is the real-time adaptive segmentation of the video using a dynamic selection of trained background models.

3.2 Real-Time Adaptive Figure-ground Segmentation

We denote κ a cluster of training images belonging to the same context θ . The set of the n clusters is noted $\mathcal{K} = \{\kappa_1, \dots, \kappa_n\}$ and the corresponding context set $\Theta = \{\theta_1, \dots, \theta_n\}$. For a new incoming image I not belonging to the training set, a global feature vector $\mathbf{v}(I)$, here a coherent color histogram in the HSV color space, is extracted and classified into a cluster. The classification is based on the minimization of the L2 distance between the feature vector and the cluster set $\{\kappa_i\}$ as follows:

$$I \in \theta_i \Leftrightarrow \mathbf{v}(I) \in \kappa_i \mid i = \arg \min_{i \in [1, n]} \text{dist}(\mathbf{v}(I), \kappa_i) \quad (1)$$

The background model associated with the detected context θ_i , is returned.

We also use a temporal filtering step to reduce the instability of the clustering algorithm when a foreground object appears. Indeed, in this case, a noise context is detected most of the time. So, it is important to smooth the analysis by

balancing the current result with respect to previous ones. Our temporal filtering criterion is defined as follows. Let us define θ the context cluster identifier (the buffered context), θ_I the cluster identifier for the incoming image I , and μ_θ the mean of cluster probability computed on a temporal window. To decide if θ_I is the adequate cluster for an incoming image I , we compare it with θ as in Algorithm 1. In this algorithm, three cases are investigated. If θ_I is equal to θ or to 0 (noise context), μ_θ is averaged based on the last context probability $p(\theta_I)$ and θ remains unchanged. In the third case, (θ_I differs from θ and 0), θ is updated and μ_θ is updated according to $p(\theta_I)$.

Algorithm 1 Context Temporal Filtering Algorithm

Input: I

Output: θ

```

 $\theta \leftarrow 0$  {set buffered context identifier to ‘noise’ (for the first frame only)}
 $\mu_\theta \leftarrow 0$  {set  $\theta$  probability to 0 (for the first frame only)}

 $[\theta_I, p(\theta_I)] \leftarrow \text{ContextAnalysis}\{I\}$  { $\theta_I =$  context ident. of  $I$ }
if  $\theta = \theta_I$  or  $\theta = 0$  then
     $\mu_\theta \leftarrow \frac{\mu_\theta + p(\theta_I)}{2}$  { $\mu_\theta$  averaging}
else
     $\theta \leftarrow \theta_I$  { $\theta$  updating}
    if  $p(\theta_I) \geq \mu_\theta$  then
         $\mu_\theta \leftarrow p(\theta_I)$  { $\mu_\theta$  updating}
    else
         $\mu_\theta \leftarrow \frac{\mu_\theta + p(\theta_I)}{2}$  { $\mu_\theta$  averaging}
    end if
end if
return  $\theta$ 

```

When the context is identified, the corresponding background model is selected and the figure-ground segmentation of I is performed, as sketched in Figure 2.

4 Experimental Results

4.1 Experiment

The experimental conditions are the followings: the video data are taken during a period of 24 hours, at eight frames per second, from a video surveillance camera fixed above an outdoor cash desk of a car park. The video camera parameters are set in automatic mode. The size of the images is 352×288 pixels and are stored in JPEG format. For the experiment, we have taken one frame on five which correspond to 138000 frames in total. Four samples picked from the image set are shown in Figure 3. They have been chosen to illustrate the background modelling

problem. In the learning stage, we have manually defined a training image set I composed of 5962 background frames (i.e. without foreground objects) along the sequence. This corresponds to pick one frame every 15 seconds in mean and represents 4.3% of the whole image set. Figure 4 gives a quick overview of the

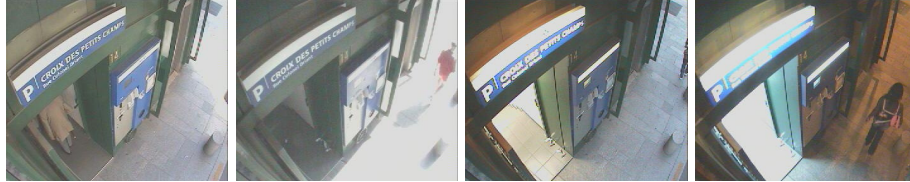


Fig. 3. Four frames representative of the illumination changes at different times of the day.

global feature distribution along the sequence. In this figure, each X-Z slice is an histogram which represents the percentage of the number of pixels (Z axis) belonging to a given color coherent feature (X axis). The coherent color feature scale has been divided into 3 intervals for the three HSV channels. The Y axis represents the time in the course of a day. Several clusters of histograms can be easily visually discriminated as notified for cluster number 1, 10 and 2. Other clusters, not represented here, are intermediate ones and mainly correspond to transitions states between the three main clusters. Sixteen clusters are found (see Figure 5 for context class distribution). Three major clusters can be identified (number 1, 2 and 10). The order of class representation does not necessary correspond to consecutive time instants. Cluster 1 corresponds to noon (sunny context), cluster 2 corresponds to the morning (lower contrast) and cluster 14 to the night. We compare the results obtained with different segmentation settings (with or without the context adaption, etc.) at different times of the day and in several difficult situations.

4.2 Model Selection Effects

In this section, we show three examples (three columns) in Figure 6 where the selection of the background model helps to improve the segmentation. Boundaries of the detected regions (in green) have been dilated for a better visualization. **context ID** is the identifier of the detected context and **prob** is the estimate probability of the identified context. In this figure, the first row corresponds to the direct codebook segmentation when trained on the whole training image set. The second row corresponds to our context-based codebook segmentation. We can see that our approach achieves a better detection rate without adding false detection.

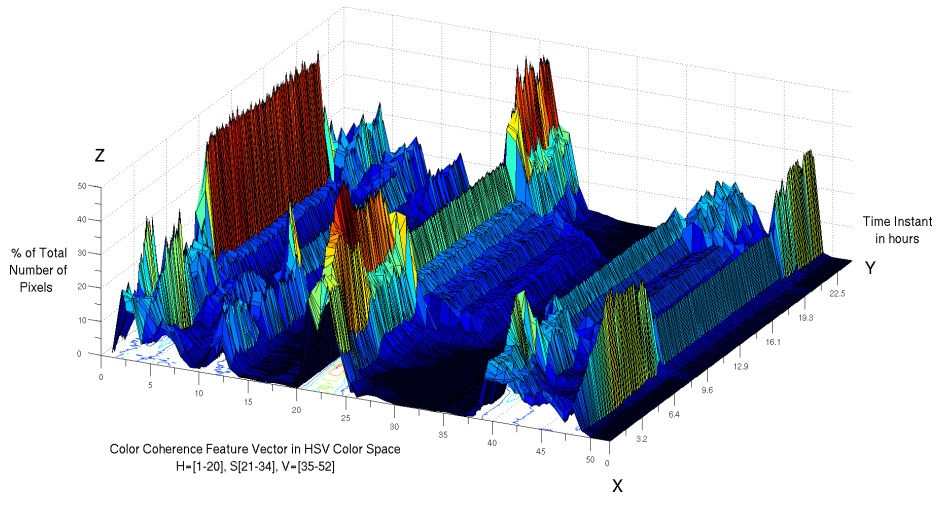


Fig. 4. 3-D histogram of the training image set used for the clustering (see Figure 3 for samples).

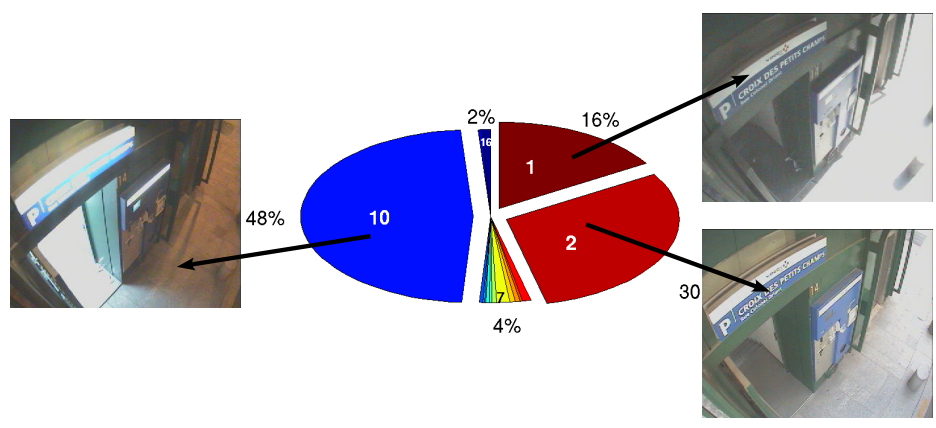


Fig. 5. Pie chart of the context distribution. The three most represented image clusters are shown corresponding to night, sunny, and cloudy contexts.



Fig. 6. Comparison of direct search codebook segmentation (first row) with our context-based codebook segmentation (second row) for three different contexts.

4.3 Temporal Filtering Effects

In this section, we present some situations where the temporal filtering algorithm can help to correct classification mistakes. The columns of Figure 7 correspond to the segmentation result with the codebook algorithm based on respectively one background model (left column), dynamic selection of the background model (middle column), and dynamic selection of the background plus temporal filtering (right column). The presence of a person modifies the pixel distribution of the scene and then disturbs the context classification. Consequently, a ‘noise’ context (ID:0) is often detected as shown in Figure 7 (second row middle column). The temporal filtering algorithm smooths the context analysis by integrating the results of the previous frames, and then helps in keeping a correct context classification in such cases. We can also see on the second row that the man’s shadow is not detected. In fact, context ID:1 gathers frames from sunny and shaded illumination conditions of this scene part. The corresponding background model has thus integrated these values during the training.

4.4 Comparison with Mixture of Gaussian

In this section, we compare our approach with the MoG approach. We use an implementation of the algorithm proposed in [2]. We use the default parameter setting¹. A MoG background model is trained for each identified cluster then dynamically selected during the real-time segmentation. Figure 8 shows the high sensitivity of Mog to global changes (first frame) and the effects of a too large

¹ number of gaussians = 3, learning rate = 0.05, μ and σ update rate = 0.005

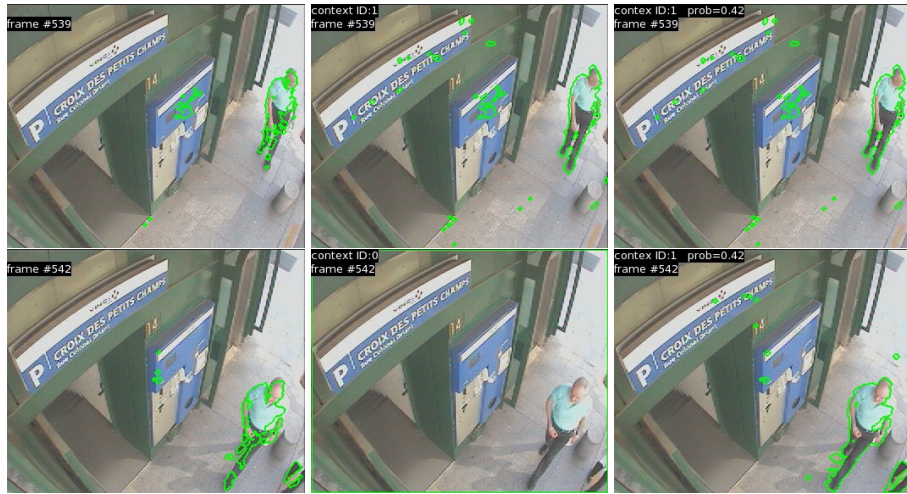


Fig. 7. Illustration of the temporal filtering effect on the context analysis. Columns are, from left to right: without context adaptation, with context adaptation, with filtered context adaptation. Rows are frame at time t and $t+1.87''$.

learning rate (second frame): foreground pixels from the first frame still remain 231 frames later (ghost formations).

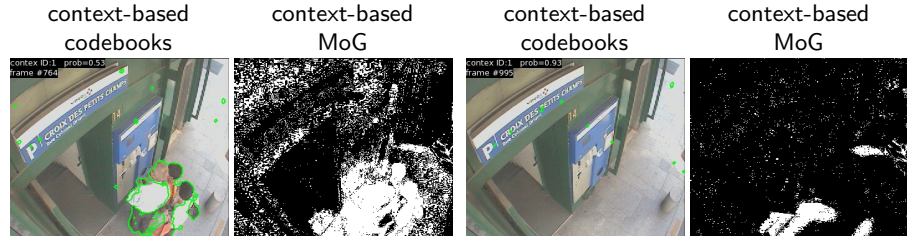


Fig. 8. Comparison between the proposed approach applied to the codebook model [4] and the MoG model [2] for two frames at time t and $t+2'24''$.

5 Conclusion

In this paper, we have presented an original weakly supervised learning approach for the dynamic selection of background model. This approach, consisting in generating sub-goals and training learning-based algorithms on each sub-goal is similar to a meta-learning approach. Our main contribution is thus at the

context modelling level: based on local and global image information, different contextual situations are automatically identified and learned thanks to a clustering algorithm. This approach is particularly interesting for very long video sequences (several hours) where both quick and long-term image variations do not allow to maintain robustly a background model.

Promising results are presented on a very long-term video surveillance application (outdoor car park entrance surveillance) where both gradual and sudden changes occur. In a weakly supervised learning stage, the user collects background samples of the different situations. The clustering algorithm has successfully identified meaningful clusters of training images like sunny context, night context, or dawn context. For each identified image cluster, a background model has been trained using the codebooks [4] and the MoG [2]. In real-time figure-ground segmentation, the different contexts are successfully retrieved thanks to the temporal filtering algorithm. The codebook model has shown to be well-adapted to deal with background model splitting and real-time constraints. Comparisons with the MoG model reveal its robustness in different situations as quick illuminations changes variations or shadows removal.

However, some problems remain in the context adaptation especially when unforeseen changes occur. We plan to cope with these problems by applying incremental clustering techniques. Moreover, a quantitative evaluation study remains to be done to objectively assess our approach against other algorithms. We are currently investigating video databases with ground truth data.

References

- [1] Prati, A., Mikic, I., Trivedi, M., Cucchiara, R.: Detecting moving shadows: algorithms and evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25**(7) (2003) 918–923
- [2] Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. In: *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*. (1999) 246–252
- [3] Elgammal, A.M., Harwood, D., Davis, L.S.: Non-parametric model for background subtraction. In: *ECCV '00: Proceedings of the 6th European Conference on Computer Vision-Part II*, London, UK, Springer-Verlag (2000) 751–767
- [4] Kim, K., Chalidabhongse, T.H., Harwood, D., Davis, L.: Real-time foreground-background segmentation using codebook model. *Real-Time Imaging* **11**(3) (2005) 172–185
- [5] Georis, B., Bremond, F., Thonnat, M.: Real-time control of video surveillance systems with program supervision techniques. *Machine Vision and Applications* **18**(3-4) (2007) 189–205
- [6] Pass, G., Zabih, R., Miller, J.: Comparing images using color coherence vectors. In: *ACM International Conference on Multimedia*, ACM Press, New York, USA (1997) 65–73
- [7] Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: *Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining*, Portland (1996) 226–231