Computer Vision and Image Understanding 133 (2015) 30–50

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Features for stochastic approximation based foreground detection ☆

Francisco Javier López-Rubio, Ezequiel López-Rubio *

Department of Computer Languages and Computer Science, University of Málaga, Bulevar Louis Pasteur, 35, 29071 Málaga, Spain

ARTICLE INFO

Article history:
Received 13 May 2014
Accepted 24 December 2014
Available online 31 December 2014

Keywords:
Background modeling
Foreground detection
Probabilistic mixture model
Background features
Pareto front

ABSTRACT

Foreground detection algorithms have sometimes relied on rather ad hoc procedures, even when probabilistic mixture models are defined. Moreover, the fact that the input features have different variances and that they are not independent from each other is often neglected, which hampers performance. Here we aim to obtain a background model which is not tied to any particular choice of features, and that accounts for the variability and the dependences among features. It is based on the stochastic approximation framework. A possible set of features is presented, and their suitability for this problem is assessed. Finally, the proposed procedure is compared with several state-of-the-art alternatives, with satisfactory results.

1. Introduction

The proliferation of video vigilance systems has given rise to an ever increasing need of preprocessing, analyzing, indexing and searching over huge quantities of video data. These tasks cannot be efficiently carried out by humans, since the volume of data to be processed is too large for their capabilities. Hence, activity analysis intelligent systems are an emergent research field with multiple applications, both in the private and public sectors [36].

Under a modular view of computer vision systems, the separation of moving objects from the background is one of the earliest stages. This is an essential part of any surveillance system, since its performance deeply influences the higher level stages which carry out the object detection. Consequently, much effort has been devoted to this complex problem, which includes challenges such as dynamic backgrounds, shadows, objects which integrate into the background, sudden illumination changes, color similarity (camouflage) and many others [4].

Most approaches are based on building a model of the background which is based on some statistics obtained from the previous video frames. We might classify most of them into four classes: median based, kernel density estimation based, subspace based, and probabilistic mixture based [15]. The first class computes the median of the pixel values over the last frames in order to obtain a robust estimation of the background image. These approaches exhibit a good resilience against noise and artifacts [8], but they are also computationally expensive if the frame window is large, a problem which can be alleviated by fast methods to approximate the median [29,33]. The second kind of approaches estimates the probability density function (pdf) of the pixel values, but it does not assume any particular probability distribution for them, i.e. non parametric methods are used. Gaussian kernels are commonly chosen for this purpose [9,14]; the fundamental parameters to be tuned in this case are the number of kernels and their bandwidth. There is also a need to reduce the inherent computational load of kernel density estimation, which can be done by dropping irrelevant features. Subspace based methods try to find a subspace of the space of all possible images where the background of the scene lies, so that departures from that subspace can be detected as foreground objects [31,46,44,42]. Finally, the fourth group of methods assumes that the pixel values follow certain probability distribution, usually a mixture of Gaussians, and then it tries to estimate its parameters; this means that they are parametric methods. They tend to have less memory and time requirements than non parametric ones, since the number of parameters is relatively small. This is one of the causes of their popularity [45,40,51,17], along with the possibility to introduce specific mechanisms to tackle the above mentioned challenges [4].

Even though there is a large number of background subtraction algorithms based on probabilistic mixtures, a majority of them stick to a set of simplifications which can reduce their performance. On one hand, most proposals use the RGB pixel values as inputs [34]. On the other hand, some well established and frequently used background modeling algorithms use the same variance for all the input variables [17,41,51], although there is no

☆ This paper has been recommended for acceptance by Andrea Prati.

* Corresponding author.
E-mail addresses: xavierprof@hotmail.com (F.J. López-Rubio), ezeqlr@lcc.uma.es (E. López-Rubio).
URL: http://www.lcc.uma.es/%7Eezeqlr/index-en.html (E. López-Rubio).

http://dx.doi.org/10.1016/j.cviu.2014.12.007
1077-3142/

fundamental reason not to use the full covariance matrix. In other words, the use of full covariances is an option which exists in theory but is very rarely implemented in practice. A spherical covariance matrix might not be the optimal choice because the variance of the input variables can be different, which means that those models do not adapt to the specific dispersion of each variable. For example, the variabilities of median filtered features are expected to be much lower than those of non median filtered features. Also, features based on edge information (high pass filters) tend to vary more than features based on low pass filters. Moreover, the above mentioned models do not consider the covariances among the variables, so they treat them as if they were independent, which is not the case. For example, the red, green and blue color components of a pixel will typically grow or diminish together as the lighting increases or decreases, respectively. This means that RGB color components are strongly correlated. Another example is edge features in the horizontal and vertical directions, since textured objects will have high values of the features in both directions, while homogeneous objects will have low values of the features in both directions.

Our aim here is to develop a method which overcomes the limitations we have just outlined, along with a set of relevant features that yields adequate results. Our proposal defines a probabilistic model which handles any number of pixel features. It also accounts for the correlations among the features, so that a more realistic model is obtained.

The structure of this paper is as follows. First of all, a review of previous work about probabilistic background models and feature selection is done in Section 2. Then the proposed probabilistic mixture model and the corresponding learning algorithm are considered in Section 3. The set of pixel features that we have chosen are defined and studied in Section 4. Section 5 is devoted to the experimental results, with comparisons with state-of-the-art approaches. Finally, Sections 6 and 7 deal with the discussion of the most relevant characteristics of our proposal and the conclusions, respectively.

2. Related work

As outlined above, the design of a realistic mixture model for foreground detection can be decomposed into two decisions: the choice of the probability model for each mixture component and the selection of the number and kind of input features. Next a review of previous literature on these two topics is carried out, with indications of possible enhancements.

Most methods rely on the assumption that all the variables have the same variance, and that all of them are independent. This translates to Gaussian mixture components with spherical covariance matrices, i.e. matrices of the form $\sigma^2\mathbf{I}$, where σ^2 is the common variance and $\mathbf{I}$ is the identity matrix. It has long been known that the assumption of a spherical covariance matrix does not conform to real scenes. In [12] it is found that the estimation of the mean vector is reliable, but the covariance matrix does not conform to a spherical model. The distribution of values for a background pixel in the standard RGB space conforms to a cylinder rather than a sphere, which renders the spherical model unrealistic [19,5]. The cylindrical model is proposed to overcome this problem. Under this model the major axis of the cylinder extends to the black point, and the position along this axis corresponds to luminance. The distance of a point to the major axis determines the radius of the cylinder, which is called the color distortion. The cylindrical model has the disadvantage that it is not associated to any probability density, since it only defines a distance measure of how close a color is to the model. Hence it is not possible to assign a probability that a pixel belongs to the background, and consequently the amount of learning of the background model of a pixel cannot be adapted to the likelihood that the observed color corresponds to the background, which affects the learning process negatively. On the other hand, a Gaussian with a full covariance matrix can model an elongated shape with any position, thickness and orientation, and it produces the probability that a pixel belongs to the background, so it is a viable alternative to fit the data.

A commonly used alternative is to use several spherical Gaussians for the background [40,49,6], rather than one [45]. Comparative studies show that more than one spherical Gaussians perform better than a single spherical Gaussian [11]. Nevertheless, a mixture of spherical Gaussians can only give a rough approximation of an elongated cluster of data unless the number of Gaussians grows considerably, because the points which are midway between the mean vectors of two successive Gaussian components have a lower probability density than those closer to a mean vector. Again, a Gaussian with a full covariance matrix does not have this problem, since the probability density is smooth through the elongated cluster. This is confirmed by the study in [3], which points out that a single Gaussian with a full covariance matrix outperforms a model with multiple spherical Gaussians in some situations. It is reported that this is because the covariances are able to adapt to background instabilities.

Last but not the least Gaussian mixture components cannot model the foreground adequately unless a large amount of them is used. This is because incoming foreground objects can have any aspect, so any mixture component used to model the foreground should have a substantially flat profile, i.e. no prominent modes. However, most algorithms neglect this fact and assign a small number of spherical Gaussians to the foreground [18,6]. This leads to very large variances in the foreground Gaussians as they try to adapt to heterogeneous foreground input samples. Moreover, a large part of the probability mass of these Gaussians could be out of the support of the real input distribution, since the Gaussians must adapt to the input samples which lie near the borders of the support. That is, they are modeling regions where no inputs can exist. Of course this is also true for the Gaussians assigned to the background, but in this case the effect is not so serious because the background samples are expected to be more concentrated, so that the spread of the Gaussians is smaller and little probability mass is out of the support of the real input. On the other hand, the assignment of a Gaussian to the background or the foreground is a difficult problem on its own. This situation calls for an uninformative prior for the foreground, i.e. a single flat mixture component such as a uniform distribution [35,13], which is the approach that we advocate here.

The second fundamental decision to be made when a foreground detection algorithm is developed deals with the set of input features. The most straightforward option is to use the raw RGB information from the camera, and it is chosen by many proposals. It has the advantage of its speed and ease of implementation, but it is widely acknowledged that it suffers from limitations due to illumination changes, among other artifacts. Separation of luminance and chrominance can help to reduce these undesirable effects, as done in [45], where the YUV color space is employed. The Y channel conveys the luminance information, while U and V channels carry chrominance. A further development is to remove the luminance information completely, which is commonly done by dividing the RGB values by their sum to yield the normalized RGB set of features [38,28,2,9]. This attains certain resilience to illumination changes, but the differing optical characteristics of objects imply that normalized RGB data are still subject to variation when lights are switched on or off. A way to remove the dependence from color and lighting is to use texture features [48,16], although they are