Neurocomputing 119 (2013) 439–453

Contents lists available at SciVerse ScienceDirect

Neurocomputing

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

Future-data driven modeling of complex backgrounds using mixture of Gaussians

Xin Liu *, Chun Qi

School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an, China

ARTICLE INFO

Article history:
Received 23 August 2012
Received in revised form
12 February 2013
Accepted 31 March 2013
Communicated by M. Wang
Available online 8 May 2013

Keywords:
Background modeling
Background subtraction
Gaussian mixture models
Learning-rate
Foreground detection

ABSTRACT

Mixture of Gaussians (MoG) is well-known for effectively in sustaining background variations, which has been widely adopted for background subtraction. However, in complex backgrounds, MoG often traps in keeping balance between model convergence speed and its stability. The main difficulty is the selection of learning rates. In this paper, an effective learning strategy is proposed to provide better regularization of background adaptation for MoG. First, the video-data is splitted into the future-data and history-data, then a set of background distributions (MoG) is computed for each case. To distinguish between dynamic and static background, the equality of these two sets is tested by the hypothesis testing method. Next, a two-layer LBP-based method is proposed for foreground classification. Finally, the global and static learning rates are replaced by the adaptive learning rates for image pixels with distinct properties for each frame. By means of the proposed learning strategy, a novel background modeling for detecting foreground objects from complex environments is established. We compare our procedure against the state-of-the-art alternatives, the experimental results show that the performance of learning speed and accuracy obtained by proposed learning rate control strategy is better than existing MoG approaches.

1. Introduction

The segmentation of moving foreground objects from video stream is the fundamental step in many computer vision applications, such as intelligent visual surveillance [1,2], human–machine interaction [3,4], and content based video coding [5]. Background subtraction is generally regarded as an effective method for extracting foreground objects [6–8]. The performance of background subtraction mainly depends on the algorithm used for modeling background. However, the background is a complex environment usually includes distracting motions which make the task more challenging, such as waving tree branches and rippling water. An adaptive background modeling algorithm should also detect shadows cast by the moving objects, and handle various changes due to the situation where new objects are introduced to the background or old ones removed from it. Furthermore, an ideal background modeling algorithm should be able to tolerate sudden background variations like the changing weather conditions or the turn on/off lights, without losing sensitivity to detect real foreground objects.

Many background modeling algorithms have been proposed (surveys [8,9]). A very popular approach is to model each pixel in a video frame with mixture of Gaussians, instead of using the exact Expectation Maximum (EM) algorithm by Friedman and Russell et al. [10], an online K-means approximation proposed by Stauffer and Grimson [11], which has become the standard formulation for the MoG approach in this field. In their approach, an online learning was used to train background model. For every image pixel, regardless of its intensity being changed or not, the learning rate of updating is controlled by a global, static parameter α that ranges between 0 and 1. Because only one new sample is observed, typically, a very small constant is used to maintain the stability of models. Unfortunately, this small constant leads to a very slow learning speed which cannot update model parameters in time for the changing background [8,12]. In contrast, by setting a larger value for α would improve the convergence speed for dynamic background. However, the system will become sensitive to foreground and noise.

Many papers have proposed improvements and extensions in such aspect. In [13], Elgammal et al. proposed a non-parametric model for background modeling, where a kernel-based function is employed to represent the color distribution of each background pixel. The kernel-based distribution is a generalization of MoG which does not require parameter estimation. But the computation is high for this method. KaewTraKulPong et al. [14] utilized expected sufficient statistics at the initial phases to improve convergence to resolve the slow learning at the beginning, and also using an online K-means algorithm with a constant learning

* Corresponding author. Tel.: +86 18687188968.
E-mail address: linuxsino@gmail.com (X. Liu).

0925-2312/$ - see front matter
http://dx.doi.org/10.1016/j.neucom.2013.03.013

rate. However, the background features of initial phase were limited after all, this two-phase method would not improve adaption for modelling at later phase. In [15], Harville discussed some trade-off encountered by MoG and introduced a framework for guiding MoG evolution with feedback from high-level modules. In [12], not only the learning parameters but also the number of models of the mixture was adapted for each pixel. Tian et al. [16] detected the static foreground regions that were wrongly modeled as the background and proposed a weight exchange scheme to avoid a fragment problem in MoG. Tuzel et al. [17] proposed to estimate the probability distribution of mean and covariance of each Gaussian using recursive Bayesian learning. In [18], Lee et al. proposed an effective learning algorithm that improved convergence rate of background model estimation without obvious side-effects on system stability. Yang et al. [19] proposed a two-layer MoG to learn foreground and background models at different learning rates. Chen et al. [20] also proposed a hierarchical method with the MoG, the method employs block-based and pixel-based strategies, but shadows cannot be removed with their method. In [21], White et al. proposed a particle swarm optimization method to automatically determine the parameters of the MoG. In [22,23], Maddalena et al. proposed a self-organizing approach to background subtraction, where the background model can automatically adapt to a self-organizing manner through artificial neural networks. In [24], Lin et al. proposed a new rate control scheme based on high-level feedback to resolve the trade-off and adjustments of the MoG's learning rates for different foreground pixel types. Furthermore, Lin et al. adopted frame difference method to detect over-quick lighting change and adjust learning rates. However, a quick variety of lighting occurred in very short time intervals and only a few times of successful detection by frame difference, so the chance for learning the over-quick change is pretty rare.

2. Future-data driven background modeling

In this section, we will present the proposed background modeling in four parts. First, the work of MoG and the convergence of the learning process are analyzed. We highlight the importance of learning rate control for MoG and elaborate its relationship with pixel classification. Second, a fast and efficient convergence method in background modeling based on MoG for future-data is proposed, and a hypothesis testing method is used to test the equality of history Gaussian models and future ones, and to decide whether the background will change in future. By this decision, a classification for pixel types is proposed, the whole background pixels are divided into dynamic and static types. Third, a two-layer texture method based on LBP for foreground classification is proposed, the true foreground is distinguished from the false foreground. Fourth, a learning strategy that controls learning rate for MoG is detailed. Under this control, different learning rates can be applied to different pixel types which labeled already. The sudden illumination changes are detected by testing if the background likelihood ratio is smaller than a threshold. If a change is detected, the whole background is updated with a larger rate to respond to the once-off changes.

2.1. Mixture of Gaussians approach

For a pixel p, at position (x,y), at time t, what is known about its history $\{X_1,\ldots,X_{t-1}\}=\{I(x,y,i):1\le i\le t-1\}$ is modeled by a mixture of K Gaussian distributions, where $I(\cdot)$ is the pixel's intensity value. The probability of observing the current pixel value is

$$p(X_t)=\sum_{i=1}^{K}\omega_{t,i}\eta(X_t,\mu_{t,i},\sigma_{t,i}^2) \tag{1}$$

where $\eta(\cdot)$ symbolizes a Gaussian probability density function, and μ is the mean, σ^2 is the covariance, are the Gaussian parameters of the ith distribution, where $\omega_{t,i}$ is the respective weight

$$\eta(X,\mu,\sigma^2)=\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(X-\mu)^2}{2\sigma^2}\right) \tag{2}$$

The K Gaussians are ordered based on ω_i/σ_i, the first B Gaussians are used as models of the background, estimated as

$$B_t=\arg\min_b\left(\sum_{i=1}^{b}\omega_{t,i}>T\right) \tag{3}$$

$$\sum_{i=1}^{K}\omega_{t,i}=1 \tag{4}$$

where T is a measure of the minimum portion of the data that should be accounted for by the background. The pixel value X_t lying within 2.5 standard deviations of a distribution is defined as a match. If a match is found with one of the first B Gaussians, the pixel is classified as background. If none of the first B distributions be matched, the pixel is classified as foreground, namely, foreground detection is performed by

$$\forall i\in(1,\ldots,B_t)\ |X_t-\mu_{t,i}|>2.5\sigma_{t,i} \tag{5}$$

To maintain and update mixture Gaussian models, the weight of the each model is to be updated by the equation

$$\omega_{t+1,i}=(1-\alpha)\omega_{t,i}+\alpha p(\omega_{t,i}|X_t) \tag{6}$$

If i is the matched Gaussian model $p(\omega_{t,i}|X_t)=1$, otherwise it equals 0. The first Gaussian model (noted as ma) that matches the current pixel value will be updated by

$$\mu_{t+1,ma}=(1-\beta)\mu_{t,ma}+\beta X_t \tag{7}$$

$$\sigma_{t+1,ma}^2=(1-\beta)\sigma_{t,ma}^2+\beta(X_t-\mu_{t,ma})^2 \tag{8}$$

$$\beta=\alpha\eta(X_t|\mu_{t,ma},\sigma_{t,ma}^2) \tag{9}$$

If none of the K distributions match that pixel value, the least probable model is replaced by a distribution with the current value as its mean μ, an initially high variance σ^2, and a low weight parameter ω.

The α defines the rate of updating that controls how fast the model converges to a new one. In the conventional MoG, for every pixel, a very small constant is commonly used for background adaptation. Unfortunately, this setting leads to slow convergence when background needs to adapt to a new cluster. For example, if a new background object comes into a scene, suppose at present the weights sum of the first B models equals unity. It will take N_T frames until the genuine object can be considered as a background. From (3) and (5) we can obtain that if the new object model's weight becomes larger than $1-T$, namely, until the weights sum of the old B models becomes smaller than T, the genuine object can be considered to be part of the background. From (6) we can conclude

$$(1-\alpha)^{N_T}<T \tag{10}$$

So, the object should be static for at least $N_T=\log T/\log(1-\alpha)$ frames. As noted in [25], using $\alpha=0.0025$ and $T=0.8$ in MoG, if we assume that the new object will presented in every frame ideally, it would take 89 frames for the component to be included as part of the background. If the frame-rate being 20 fps, the system will take at least 4.5 s to respond to background changes. The situation