Neurocomputing 72 (2009) 2450–2454

Contents lists available at ScienceDirect

Neurocomputing

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

Scene segmentation based on IPCA for visual surveillance

Yuan Yuan [a,*], Yanwei Pang [b], Jing Pan [c], Xuelong Li [d]

[a] School of Engineering and Applied Science, Aston University, Birmingham B4 7ET, UK
[b] School of Electronic Information Engineering, Tianjin University, Tianjin 300072, China
[c] Electronic Engineering Department, Tianjin University of Technology and Education, Tianjin 300222, China
[d] School of Computer Science and Information Systems, Birkbeck College, University of London, London WC1E 7HX, UK

ARTICLE INFO

Article history:
Received 26 August 2008
Received in revised form
17 November 2008
Accepted 22 November 2008
Communicated by T. Heskes
Available online 3 December 2008

Keywords:
Incremental principal component analysis
Visual surveillance
Video surveillance
Scene segmentation

ABSTRACT

This paper proposes a simple scene segmentation method based on *incremental principal component analysis* (IPCA). Instead of segmenting moving objects in a conventional *frame by frame* manner, the newly proposed method segments a scene into *unchanged background zone* (UBZ) and *moving object zone* (MOZ). As a result, moving objects normally appear in MOZs rather than UBZs, and therefore, detection and behaviours analysis can be performed in MOZs. In visual communication, UBZs do not need to be encoded and transmitted. Moreover, if an object is in UBZs, it can be linked to abnormal events. Experimental results demonstrate the contribution of the proposed method.

1. Introduction

Automated visual (video) surveillance is a challenging task in computer vision researches and applications. This task can be divided into several subtasks: (1) detection of moving objects, (2) tracking these objects over a period of time, (3) classification of these moving objects into various (motion) categories, and (4) recognition of (at least some of) their activities semantically. Reliably performing each subtask is time-consuming and, therefore, real-time implementation of the entire surveillance system remains a very difficult case. In this paper, we propose to reduce the computational cost of the procedure of detecting moving objects (i.e., the detection of foreground) by segmenting the scene into *moving object zone* (MOZ) and *unchanged background zone* (UBZ). With help of the segmentation result of the proposed method, traditional background modeling and detection foreground can be applied to the MOZ, which is actually the region of interest (ROI). Moreover, object tracking can also be limited to the corresponding MOZ. Therefore, the proposed method accelerates the visual surveillance system.

Foreground detection has been standing as a hot research topic over years [4,6,11,13,16,17]. The most popular approaches for foreground detection are probably the pixel-wise density-based ones, and Gaussian mixture model (GMM)-based background subtraction [11] is one of the most widely utilised foreground segmentation algorithms. In GMM, each pixel $\mathbf{x}$ is modeled by an adaptive mixture model:

$$p(\mathbf{x}) = \sum_{i=1}^{K} w_i \exp\left(-\frac{-0.5(\mathbf{x}-\mathbf{u}_i)^{\mathrm{T}}\boldsymbol{\Sigma}_i^{-1}\mathbf{x}-\mathbf{u}_i)}{(2\pi)^{n/2}|\boldsymbol{\Sigma}_i|^{1/2}}\right),$$

where $\mathbf{u}_i$ and $\boldsymbol{\Sigma}_i$ are the mean vector and covariance matrix of the ith component, respectively, w_i is the weight of the ith mixture component, and K is the number of components. Each pixel is classified based on whether the Gaussian distribution is considered as part of the background model.

Instead of first detecting moving regions and then recognising their class labels, one can detect and locate objects directly from each frame without relying on motion information. For example, if we are monitoring pedestrians, we can learn a pedestrian classifier/learner in an offline manner. With the learner classifier, we then check all the possible sub-windows in the image at different positions and scales. The approach of HOG+SVM is a representative pedestrian detection work [2]. HOG+SVM applies support vector machine (SVM) on features extracted from histogram of oriented gradients (HOG). The success of HOG lies in its capability of capturing the intrinsic shape of the pedestrians. However, the large computational complexity of HOG+SVM makes it hard to satisfy the real-time requirement. It has to scan the whole image using windows with different sizes and scales.

GMM is pixel density-based method, while HOG+SVM is a sub-window searching-based method. Both are time-consuming. Consider that automated visual surveillance system has to conduct tracking and behaviour analysis tasks, it requires that foreground/object detection module occupies less computation time. The proposed method is a preprocessing step which can be

* Corresponding author.
E-mail addresses: yuany1@aston.ac.uk (Y. Yuan), pyw@tju.edu.cn (Y. Pang), jingpan23@mail.ustc.edu.cn (J. Pan), xuelong@dcs.bbk.ac.uk (X. Li).

0925-2312/$ - see front matter
doi:10.1016/j.neucom.2008.11.013

used to facilitate and accelerate object detection, tracking, and analysis.

In this paper, we propose to segment the scene into two types of zones, namely MOZ and UBZ. The moving objects usually are included in the MOZs and seldom appear in the UBZs. Such segmentation is not aimed at directly detecting moving objects. Instead it needs to be used in conjunction with common object detection methods such as the approach of GMM and HOG+SVM. The proposed method is suitable for statistic camera. The field of view of the camera usually contains considerable unchanged background where moving objects never or seldom appear. We can learn the UBZs by observing the image sequence for a long time. The usefulness of such segmentation is at least three-fold. (1) One can detect moving objects only on the MOZs. For GMM-based moving objects detection method, it is enough to build the mixture model only on the pixels in MOZs. Similarly, for HOG+SVM one can avoid extracting the HOG features in UBZs and a lot of sub-windows are excluded beforehand. (2) If the task is to monitor abnormal behaviours/objects, one should pay more attention on the UBZs. The assumption is that with high probability the UBZs do not contain any moving objects. However, once some moving objects appear in the zone, they are abnormal objects with high probability. (3) Once the UBZs are determined, we can avoid encoding it frame by frame.

The proposed scene segmentation method is based on principal component analysis (PCA) [12], particularly incremental PCA (IPCA) [14]. Our method determines the moving region by using the basis vectors of IPCA. The key to this method relies on understanding the physical meaning of the absolute value of each entry of the basis vector. As images come in a frame by frame manner, the principal vectors can be updated by the IPCA algorithm. After a period of time, a simple function (i.e., (8)) of the principal vectors captures "moving-ability" of each pixel of the scene. We then threshold the resulting function and obtain the termed MOZs and UBZs.

It is worth noting that the proposed scene segmentation is different from the traditional image segmentation approaches, such as level sets-based [15], hierarchical cluster model-based [9], mean-shift-based, and normalised cuts-based methods. The image segmentation approaches basically segment an image in some questions that are based on the similarity of neighbourhood pixels. Our novel scene segmentation method classifies each pixel location as either static or moving based on whether or not moving objects have passed the location. So, the newly proposed one is simple but effective and efficient. In fact, in future work, other features, e.g., colour information [18], could be considered to further improve the performance.

The rest of this paper is organised as follows: Section 2 gives an introduction to PCA and IPCA. We present the proposed method in Section 3. Section 4 gives experimental results. Section 5 concludes the paper.

2. PCA and IPCA

In this paper, we propose to use incremental PCA to segment the scene into UBZs and MOZs. In this section, we give a brief introduction to PCA [12] and IPCA [14].

2.1. PCA

PCA is the most classical subspace learning algorithm. It learns d basis vectors $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_d) \in \mathbb{R}^{D\times d}$ from N training samples $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N) \in \mathbb{R}^{D\times N}$ based on the least squares reconstruction error criterion or maximum variance criterion. Batch PCA can be formulated as an eigen-decomposition problem:

$$\mathbf{C}\mathbf{u}_i = \lambda_i \mathbf{u}_i, \tag{1}$$

$$\mathbf{C} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^{\mathrm{T}}, \tag{2}$$

$$\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i. \tag{3}$$

The batch PCA requires that all the training samples be available before the basis vectors can be estimated. Because the covariance matrix is usually very large, both the computation and storage complexity are very large. Thus batch PCA is not suitable for real-time applications.

2.2. IPCA

We are studying video data. The number of training images/frames of the video is very large. The image size is typical 320×240 and the corresponding covariance matrix is of size $76\,800 \times 76\,800$. Thus it is infeasible to apply batch PCA on such data. IPCA can overcome the limitations of batch PCA. IPCA does not need to store all the training images before computing eigenvectors (basis vectors). There are two types of IPCA: covariance-involved IPCA [3,10] and covariance-free IPCA [14]. In covariance-free IPCA, eigenvectors are updated directly according to the previous eigenvectors and a new observation image. In this paper, we use the covariance-free IPCA proposed in [14].

In the following we shall introduce how one can derive IPCA [14] from batch PCA. Let $\mathbf{v} = \lambda\mathbf{u}$ where $\|\mathbf{u}\| = 1$, then $\mathbf{u} = \mathbf{v}/\|\mathbf{v}\|$ and $\lambda = \|\mathbf{v}\|$. Then the eigen-decomposition problem in Eq. (1) becomes [14]:

$$\mathbf{v} = \mathbf{C}\mathbf{u}. \tag{4}$$

That is,

$$\mathbf{v} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}\mathbf{u}. \tag{5}$$

Formulate Eq. (5) in the incremental version:

$$\begin{aligned}\mathbf{v}(N) &= \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}\mathbf{u}(i) = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}\frac{\mathbf{v}(i-1)}{\|\mathbf{v}(i-1)\|} \\ &= \frac{1}{N}\left(\sum_{i=1}^{N-1}\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}\frac{\mathbf{v}(i-1)}{\|\mathbf{v}(i-1)\|}\right) + \frac{1}{N}\mathbf{x}_N\mathbf{x}_N^{\mathrm{T}}\frac{\mathbf{v}(N-1)}{\|\mathbf{v}(N-1)\|}.\end{aligned} \tag{6}$$

Because of

$$\mathbf{v}(N-1) = \frac{1}{N-1}\sum_{i=1}^{N-1}\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}\frac{\mathbf{v}(i-1)}{\|\mathbf{v}(i-1)\|},$$

Eq. (6) can be written as

$$\mathbf{v}(N) = \frac{N-1}{N}\mathbf{v}(N-1) + \frac{1}{N}\mathbf{x}_N\mathbf{x}_N^{\mathrm{T}}\frac{\mathbf{v}(N-1)}{\|\mathbf{v}(N-1)\|}. \tag{7}$$

The above Eq. (7) together with $\mathbf{u} = \mathbf{v}/\|\mathbf{v}\|$, $\lambda = \|\mathbf{v}\|$ and $\mathbf{v}_0 = \mathbf{x}_1$ constitute the algorithm of IPCA. One can observe from Eq. (7) that IPCA does not need to compute and update any covariance matrix [7]. For a new input observation, the scaled eigenvector $\mathbf{v}$ is updated in one step instead of many iterations. Thus the storage and computation complexity is very efficient. Hence, it is reasonable to employ such IPCA to video data.

As an useful trick, one should compute the second item of Eq. (7) by first calculating $\mathbf{x}_N^{\mathrm{T}}\mathbf{v}(N-1)/\|\mathbf{v}(N-1)\|$ and then multiply $\mathbf{x}_N/N$ by the value of $\mathbf{x}_N^{\mathrm{T}}\mathbf{v}(N-1)/\|\mathbf{v}(N-1)\|$. This trick avoids compute $\mathbf{x}_N\mathbf{x}_N^{\mathrm{T}}$ whose size is equal to the size of the covariance matrix.