FOREGROUND DETECTION: COMBINING BACKGROUND SUBSPACE LEARNING WITH OBJECT SMOOTHING MODEL

Gengjian Xue, Li Song, Jun Sun, Jun Zhou

Institute of Image Communication and Network Engineering,
Shanghai Jiao Tong University, Shanghai, China
Shanghai Key Laboratory of Digital Media Processing and Transmissions, Shanghai, China
{xgjsword, song_li, junsun, zhoujun}@sjtu.edu.cn

ABSTRACT

Foreground detection is a challenging problem in complex scenes. In this paper, a novel foreground detection method is proposed which combines background subspace learning with object smoothing model. Considering background scenes in consecutive frames are almost the same, they are approximated using an efficient subspace learning technique which is based on 2D images. Due to the pixels of objects are usually clustered, an object smoothing model is adopted where a spatial smoothing constraint is imposed on its values during the estimation, and then it can be solved as a regularized matrix restoration problem with a spatial smoothing constraint. As a result, isolated noises can be suppressed while clustered foreground pixels can be preserved. We test our method on some challenging sequences and compare it with some other techniques. Experimental results show its effectiveness and robustness.

Index Terms— Foreground estimation, background subspace learning, object smoothing model.

1. INTRODUCTION

Effective foreground detection is often the first step in video processing applications, such as object recognition and surveillance. Its output is usually an input to a higher level process, making it a critical part of the system. Although various techniques have been proposed, it is still a challenging problem in complex scenes, *e.g.* swaying trees, rippling water, moving vegetation.

As a common approach to foreground detection, background subtraction has been widely used in the past few years. It consists of two parts: one is maintaining a background model, and the other one is subtracting a new frame from the background and thresholding the difference values to determine the foreground. Among many techniques in this kind, one of the most popular approaches is to model each pixel with a mixture of K Gaussians (GMM) and recursively update the parameters using an online approximation [1]. The GMM technique has become a baseline technique and has been continuously improved during the past decades. Another widely used technique is the nonparametric statistical approach. One representative method is the kernel density estimation technique (KDE) proposed in [2], where the probability density of pixel values was directly estimated by the kernel density functions without any assumptions on the pixel distributions. Then, the KDE method was improved in many ways. However, these techniques assume pixels are independent, which limits their applications in real scenes.

Recently, some approaches considered foreground detection from the viewpoint of signals processing, which can be expressed as:

$$Y = D + F \tag{1}$$

where Y is the observed signals, D and F denote the background and foreground signals respectively. Many properties of D and F have been explored for this signals separation task. For example, a method named RPCA [3] stacks a batch of observed frames columnwise to construct the matrix Y, then it separates them by imposing a low rank constraint on D and an element sparse constraint on F. However, one shortcoming of this method is time consuming as batch based methods often have to collect a group of frames before processing.

In this paper, with the same idea of equation (1), we propose an online foreground detection method which combines background subspace learning with object smoothing model. First, D is represented using an efficient subspace learning method as most background can be approximated by lower dimensional data. To estimate F, we utilize its clustered property and impose a spatial smoothing constraint on its values. Then, the estimation of F can be cast and solved as a regularized matrix restoration problem with a spatial smoothing constraint. By utilizing these properties, isolated noises can be suppressed while foreground pixels can be estimated. Besides, the proposed method is frame based and can deal with

This work was supported in part by 973 Program (2010CB731401,2010CB731406), NSFC (61221001, 61102098, 61025005), and the 111 Project (B07022).

each frame immediately. Experiments and comparisons show the effectiveness of the proposed method.

The rest of this paper is organized as follows: In Section 2, we review some prior related methods. In Section 3, the foreground detection method is proposed which combines background subspace learning with object smoothing model. Experiments and discussions are given in Section 4. Finally, the conclusions are drawn in Section 5.

2. PRIOR RELATED METHODS

Assuming foreground and background are in different subspaces, many theories and techniques have been adopted for subspace learning.

2.1. Subspace learning methods for background

As a popular used method, the PCA technique assumes that the static background in an image sequence forms the main components of the eigenspace, and foreground objects do not have a significant contribution to the eigenbackground. Thus, the input image can be approximated by the eigenbackgrounds.

The PCA method first collects N sample images $A_i (i = 1, 2, \cdots, N)$ with the size of $r \times c$ for each image. Then it transforms each frame into a vector X_i and stacks them columnwise so that the data matrix $X = [X_1, X_2, \cdots, X_N]$ has been constructed. Next, it computes the mean background vector μ_B and the covariance matrix C_B whose size is $rc \times rc$:

$$C_B = \frac{1}{N} \widetilde{X} \widetilde{X}^T \tag{2}$$

where $\widetilde{X} = [X_1 - \mu_B, X_2 - \mu_B, \cdots, X_N - \mu_B]$. In the following, this covariance matrix is diagonalized using an eigenvalue decomposition:

$$L_B = \Phi_B C_B {\Phi_B}^T \tag{3}$$

where Φ_B is the eigenvector matrix of C_B and L_B is the diagonal matrix of its eigenvalues. In order to reduce the dimensionality of the space, only M eigenvectors corresponding to the M largest eigenvalues in the matrix L_B are selected, which leads to the $rc \times M$ matrix Φ_M.

In the detection stage, an input image A_t is first transformed into a vector X_t. Then, the coordinate in the eigenbackground space of X_t is computed as:

$$q_t = (X_t - \mu_B)^T \Phi_M \tag{4}$$

Next, q_t is back projected onto the original space to reconstruct the background image $\hat{X}_t$:

$$\hat{X}_t = \Phi_M {q_t}^T + \mu_B \tag{5}$$

Finally, the foreground is detected as:

$$|X_t - \hat{X}_t| > Th \tag{6}$$

where Th is a threshold value.

Although the PCA method can model the static background, as we will show later, it can hardly work well in dynamic scenes. Furthermore, this shortcoming often exists in some other related methods as well, such as the independent component analysis (ICA) based technique [4].

2.2. Methods with sparse constraints on foreground

Assuming foreground objects are sparse in most cases, Dikmen *et al.* imposed a sparse constraint on foreground signals (Sparse) [5]. They further provided a solution to this constraint equation which is expressed as:

$$\hat{D} = \underset{D}{\operatorname{argmin}} \|Y - D\|_1 \tag{7}$$

where $\| \cdot \|$ means the L_1 norm. In the Sparse technique, the background is constructed by a linear combination of other frames in the same sequence. Consequently, the equation (7) has become as:

$$\hat{w} = \underset{w}{\operatorname{argmin}} \|Y - Xw\|_1 \tag{8}$$

where w is a N-dimensional coefficient vector, X is a dictionary which is composed of some past observed frames, and here Y is the vector of current image. According to (8), one can get an estimated coefficient vector $\hat{w}$. Then, the estimated background can be constructed as $\hat{D} = X\hat{w}$, and the foreground signals can be separated in the following. Later, the base construction techniques have been compared and discussed in [6].

Another technique called AdaDGS proposed in [7] uses both sparsity and group clustering priors to estimate foreground objects. However, the sparsity number has to be set in advance, which is unrealistic, thus the authors have provided an empirical solution by setting the sparsity range and running his technique until some halting conditions were satisfied. The AdaDGS technique is time consuming and is not a scalable solution, as foreground objects are typically variable in real applications.

3. THE PROPOSED METHOD

Our proposed method has the same philosophy with equation (1), and we do separation like the RPCA technique by exploring different properties of D and F simultaneously. For the signal D, as most background scenes are almost the same, it can be approximated by lower dimensional data and we adopt a subspace learning method to represent it. Due to dynamic background scenes, the subspace may be not well modeled and some noises often exist in the signal F. On the other hand, the pixels of foreground objects are usually clustered. Based on these analysis, we impose a spatial smoothing constraint on foreground values during the estimation, which can

suppress isolated noises while preserving clustered pixels. As a result, the proposed framework becomes a joint optimization problem by simultaneously setting the optimal parameter values for subspace learning and object smoothing. However, directly solving this problem is not an easy task. In this paper, we provide an alternative solution which first gets a rough estimate using a subspace learning technique, then an object smoothing model is adopted to refine the foreground.

As for subspace learning methods for background modeling, the PCA technique has already shown its validity. However, this technique is computationally intensive as the size of the covariance matrix C_B is usually very large. Take the frame with size of 60×80 for example, the covariance matrix in this case is 4800×4800 which results in much time consuming in eigenvalues decomposition. Recently, some efficient decomposition techniques have been proposed and applied in other applications, such as 2DPCA and $(2D)^2$PCA (See below). In order to improve the efficiency, we adopt the $(2D)^2$PCA method to model the background.

3.1. $(2D)^2$PCA based subspaces learning for background

To reduce the computational burden of decomposition on large scale matrices, some authors have proposed to compute the eigenspace directly on 2D images. In [8], a two-dimensional PCA (2DPCA) method was proposed for face representation, where the covariance matrix was constructed based on 2D images rather than 1D vectors. However, the 2DPCA method is only working in the row direction of images. Later, an improved method named two-directional two-dimensional PCA ($(2D)^2$PCA) [9] was proposed for face recognition which works in both the row and column directions. The $(2D)^2$PCA method is computationally more efficient as the covariance matrices are much smaller than that of the PCA. Thus, we use the $(2D)^2$PCA technique in our applications.

First, the mean image of training frames is computed as $\bar{A} = \frac{1}{N}\sum_{i=1}^{N} A_i$. Then, the image covariance matrices in the row and column directions are computed respectively as:

$$C^{row} = \frac{1}{N}\sum_{i=1}^{N}(A_i - \bar{A})^T(A_i - \bar{A}) \tag{9}$$

$$C^{column} = \frac{1}{N}\sum_{i=1}^{N}(A_i - \bar{A})(A_i - \bar{A})^T \tag{10}$$

Next, we diagonalize these two matrices and respectively select M eigenvectors from C^{column} and C^{row} that correspond to the M largest eigenvalues in the C^{column} and C^{row} matrices. As a result, the projection matrices Φ^{row} and Φ^{column} have been constructed.

For an input image A_t, it is projected onto the Φ^{row} and Φ^{column} spaces simultaneously, yielding a coefficient matrix Z:

$$Z = (\Phi^{column})^T(A_t - \bar{A})\Phi^{row} \tag{11}$$

Based on the matrix Z, a reconstructed image can be computed as:

$$\hat{A}_t = \Phi^{column} Z (\Phi^{row})^T + \bar{A} \tag{12}$$

Then, the differences between these two images are:

$$G = A_t - \hat{A}_t \tag{13}$$

Traditionally, the detection result can be got by thresholding the difference matrix G. However, it is difficult to accurately model the background in dynamic scenes. As a result, the foreground can not be well separated by simply thresholding. To verify our analysis, some detection results on a dynamic waving trees sequence are shown in Fig.1. We select different number of eigenvectors to construct the projection matrices for results comparison. It can be seen that foreground objects can be basically detected when a small number of eigenvectors are selected. However, selecting too few eigenvectors may lead to too many noises exist in the results so that they are difficult to be removed. On the other hand, if we select too many eigenvectors, the reconstructed image is so close to the original frame that some foreground objects are lost. In this case, the foreground can hardly be estimated as not enough foreground information is valid. Thus, the strategy we adopt is to set an immediate number for eigenvectors selection so that a rough estimate can be got which contains both the structure information of foreground and controllable noises, then an object smoothing model is adopted to refine the foreground which will be introduced in the following.

(a) (b) (c) (d)

Fig. 1. $(2D)^2$PCA detection results corresponding to different number of eigenvectors. (a) 1 eigenvector. (b) 10 eigenvectors. (c) 30 eigenvectors. (d) 40 eigenvectors.

3.2. The object smoothing model

The data values in matrix G has some properties in our applications: First, the foreground values are not randomly distributed but tend to be clustered. This is because the pixel values within an object are almost the same. Second, the number, position, and size of clusters are not known in advance. Third, some isolated data exist in the matrix which is due to dynamic background scenes. Based on these analysis, we propose to refine the foreground values in G with a spatial smoothing constraint, which is defined as:

$$\hat{H} = \underset{H}{\operatorname{argmin}} \frac{1}{2}\sum_{i=1}^{r}\sum_{i'=1}^{c}(G_{i,i'} - H_{i,i'})^2 + \lambda_1 E_1 \tag{14}$$

$$E_1 = \sum_{i=2}^{r}\sum_{i'=1}^{c}|H_{i,i'} - H_{i-1,i'}| + \sum_{i=1}^{r}\sum_{i'=2}^{c}|H_{i,i'} - H_{i,i'-1}| \quad (15)$$

where E_1 is a spatial smoothing term, $\hat{H}$ is the estimated result, λ_1 is a regularization parameter to control the smoothing level. A larger regularization value means a stronger smoothing constraint is imposed, and vice versa. According to (14), the refining task has become an optimization problem.

The processing using the above smoothing model is different from traditional morphological operations. First, morphological operations are on binary images while the smoothing model is for numerical matrices. Next, morphological techniques are often local operations with fixed element structures. In contrast, the object smoothing model can be considered as a global optimization technique. Given a regularization value, it can suppress isolated noises while preserving clustered data in an adaptive way. Once more, this smoothing model is more flexible and can be extended more with a sparse constraint on its values, which is expressed as:

$$\hat{H} = \underset{H}{\operatorname{argmin}} \frac{1}{2}\sum_{i=1}^{r}\sum_{i'=1}^{c}(G_{i,i'} - H_{i,i'})^2 + \lambda_1 E_1 + \lambda_2 E_2 \quad (16)$$

$$E_2 = \sum_{i=1}^{r}\sum_{i'=1}^{c}|H_{i,i'}| \quad (17)$$

where E_2 is an element sparse term, λ_2 is a regularization parameter that controls the pixel sparsity level. Actually, the equation (16) is equivalent to the 2D fused lasso problem proposed in [10]. However, as most data values in G are close to zero in our applications which means the matrix is basically sparse, we only use the spatial smoothing constraint E_1. After estimating $\hat{H}$ according to (14), we get the final detection result by thresholding its values:

$$|\hat{H}| > Th \quad (18)$$

3.3. The flow chart of the proposed method

Based on the above descriptions, the basic steps of our proposed method are:

(1). For each input frame A_t, selecting N past frames and considering them as training frames.

(2). Computing the mean image $\bar{A}$ of these training frames.

(3). Computing the row and column covariance matrices according to (9) and (10), respectively.

(4). Diagonalizing these covariance matrices; Selecting M eigenvectors from each covariance matrix to construct the projection matrices Φ^{row} and Φ^{column}, respectively.

(5). Computing the coefficient matrix Z of the input image according to (11).

Fig. 2. Comparison results on the *wavingtrees* sequence. The top row is the original frames named as 245^{th}, 247^{th}, 251^{th}, and 254^{th} frames. The second row is corresponding ground truth frames. The third, fourth, and fifth rows are results obtained by the GMM, KDE, and Sparse methods respectively. The last row is the results obtained by the proposed method.

(6). Getting the reconstructed image according to (12) and computing the difference matrix G according to (13).

(7). Optimizing the foreground values in G with a given λ_1 value according to (14).

(8). Thresholding the estimated matrix $\hat{H}$ with a value Th to yield the final result according to (18).

4. EXPERIMENTS

To evaluate the performance of the proposed method, some publicly available video sequences are adopted for testing and three challenging videos characterized by dynamic scenes are selected for demonstration. The proposed techniques is also compared with three widely used techniques including the GMM [1], KDE [2], and Sparse [5] methods. The parameters for these techniques are set as follows. For the proposed method, we evenly select $N = 20$ frames from the past 200 frames before the current frame as the training images. The number of eigenvectors to be selected in the range from 3 to 10 would yield comparable results. For simplicity and fair comparisons, we set the number to $M = 3$ for both the row

Fig. 3. Comparison results on the *ripplingwater* sequence. The top row is the original frames named as 1498^{th}, 1505^{th}, 1515^{th}, and 1526^{th} frames. The second row is corresponding ground truth frames. The third, fourth, and fifth rows are results obtained by the GMM, KDE, and Sparse methods respectively. The last row is the results obtained by the proposed method.

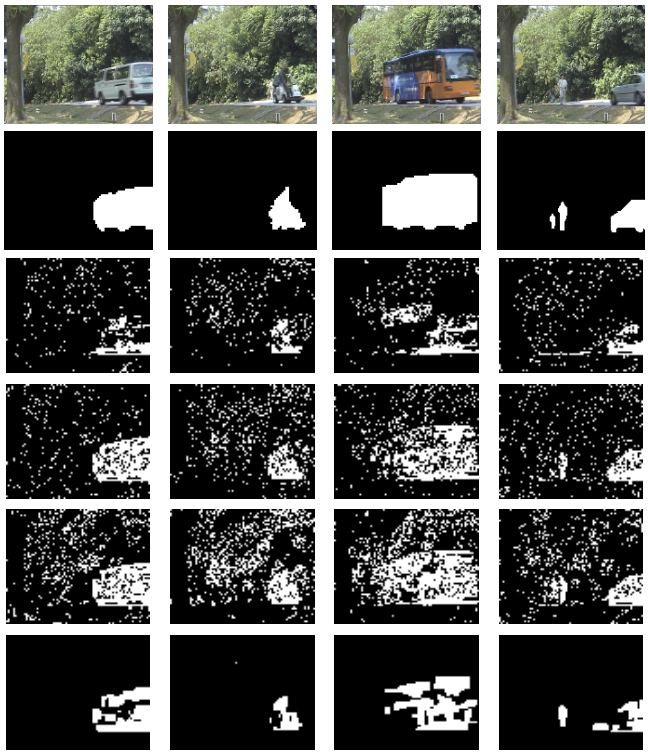

Fig. 4. Comparison results on the *campus* sequence. The top row is the original frames named as 1204^{th}, 1385^{th}, 1668^{th}, and 1812^{th} frames. The second row is corresponding ground truth frames. The third, fourth, and fifth rows are results obtained by the GMM, KDE, and Sparse methods respectively. The last row is the results obtained by the proposed method.

and column projection matrices. The regularization value is set to $\lambda_1 = 35$. The threshold value is set to $Th = 25$. For the GMM technique, the model number is set to 3, the background value is set to 0.7, and the learning rate is set to 0.01. For the KDE method, the median value for estimating the standard deviation of the kernel, window length, and threshold value are set to 2, 100, and 0.3, respectively. For the Sparse method, the selection method for training images and the threshold value are both the same as our approach. For efficient computation, we resize the video frames to half of its original size. All of the algorithms are executed on Matlab7.1 on PC computer with 2.4GHz Intel CPU, 3G RAM except that we use the R language package 'flsa' to compute the equation (14). All detection results are without post processings.

Both visual and numerical methods are adopted for comparison. As the *F-score* metric is a weighted harmonic mean of *Precision* and *Recall* to obtain a single measure that can be used to rank different methods, it is used for quantitative evaluation. This measure is defined as:

$$F = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision} = \frac{2TP}{2TP + FN + FP} \quad (19)$$

where TP, FP, and FN are true positives (true foreground pixels), false positives (false foreground pixels), and false negatives (false background pixels), respectively. Considering a video sequence usually contains a large number of frames, we randomly select 10 frames in each dataset for evaluation. Due to the limited space, the detection results of four frames in each sequence are used for demonstration.

The first sequence named *wavingtrees* is from [11] which involves heavily waving trees. The second experiment is conducted on the *ripplingwater* sequence and the third test case named *campus* is about a campus environment containing moving tree branches. The last two sequences are both from [12]. These sequences all contain dynamic background scenes but with different situations.

As shown in the Figs.2-4, the GMM, KDE, and Sparse methods detect large number of dynamic background pixels as foreground and some foreground positives in the inner areas are not detected. In contrast, the proposed method can

Table 1. Performance of F-score(%) on three video sequences

Method	GMM	KDE	Sparse	Ours
wavingtrees	68.07	73.08	76.31	**86.81**
ripplingwater	75.24	67.17	78.12	**80.88**
campus	34.42	51.05	41.93	**68.37**
Average	59.24	63.77	65.45	**78.69**

suppress most noises while detecting clustered pixels, which demonstrates that the spatial smoothing constraint on the foreground is work in these cases. The quantitative evaluations are shown in Table 1, which further verifies the effectiveness of the proposed method.

We should emphasize that the spatial smoothing model on the foreground is an integrated part of our method, and foregrounds are hardly separated without using it. Besides, this model has shown its superiority for noise removal, especially for the cases containing small foreground objects and dense noises. In these cases, traditional morphological operations are difficult to remove noises while preserving small objects. In contrast, the proposed method has yielded better results.

5. CONCLUSIONS

In this paper, a novel foreground detection method is proposed which combines background subspace leaning with object smoothing model. We use the $(2D)^2$PCA method to model the background which is computationally more efficient. Then, a spatial smoothing constraint is imposed on the estimated objects so that noises can be suppressed while clustered pixels can be preserved. Experiments on challenging sequences have shown the effectiveness of the proposed method. One of our future works is to set the optimal number for eigenvectors selection according to real scenes.

6. REFERENCES

[1] C. Stauffer and W.E.L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proceedings of the IEEE Confence on Computer Vision and Pattern Recognition*, 1999, vol. 2, pp. 246–252.

[2] A. Elgammal, D. Harwood R. Duraiswami, and L. S. Davis, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," *Proceedings of the IEEE*, vol. 90, no. 7, pp. 1151–1163, 2002.

[3] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, "Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization," *Advances in Neural Information Processing Systems*, vol. 22, pp. 2080–2088, 2009.

[4] A. Hyvarinen and E. Oja, "Independent component analysis: Algorithms and applications," *Neural Networks*, vol. 13, pp. 411–430, 2000.

[5] M. Dikmen and T. Huang, "Robust estimation of foreground in surveillance videos by sparse error estimation," in *Proceedings of the 19th International Conference on Pattern Recognition*, 2008, pp. 1–4.

[6] M. Dikmen, S.-F. Tsai, and T. Huang, "Base selection in estimating sparse foreground in video," in *Proceedings of the IEEE International Conference on Image Processing*, 2009, pp. 3217–3220.

[7] J. Huang, X. Huang, and D. Metaxas, "Learning with dynamic group sparsity," in *Proceeding of the IEEE International Conference on Computer Vision*, 2009, pp. 64–71.

[8] J. Yang, D. Zhang, A. F. Frangi, and J.Y. Yang, "Two-dimensional pca: a new approach to appearance-based face representation and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 131–137, 2004.

[9] D. Zhang and Z. Zhou, "$(2d)^2$pca: 2-directional 2-dimensional pca for efficient face representation and recognition," *Neurocomputing*, vol. 69, no. 1-3, pp. 224–231, 2005.

[10] J. Friedman, T. Hastie, H. Hoeing, and R. Tibshirani, "Pathwise coordinate optimization," *Annals of Applied Statistics,*, vol. 1, no. 2, pp. 302–332, 2007.

[11] K. Toyama, B. Brumitt, J. Krumm, and B. Meyers, "Wallflower: Principles and practice of background maintenance," in *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition*, 1999, pp. 255–261.

[12] L. Li, W. Huang, I. Y.-H. Gu, and Q. Tian, "Statistical modeling of complex backgrounds for foreground object detection," *IEEE Transactions on Image Processing*, vol. 13, no. 11, pp. 1459–1472, 2004.