GraDED: A GRAPH-BASED PARAMETRIC DICTIONARY LEARNING ALGORITHM FOR EVENT DETECTION

Tamal Batabyal, Rituparna Sarkar and Scott T. Acton, Fellow, IEEE

Virginia Image and Video Analysis Laboratory,
Department of Electrical & Computer Engineering, University of Virginia, USA
P.O. Box 400743, Charlottesville, VA 22904 U.S.A.

ABSTRACT

Short-time event detection from videos obtained using hand-held or car-mounted cameras is an overarching challenge in surveillance. The problem demands simultaneous spatio-temporal localization of the event along with removal of a dynamic background. Existing state-of-the-art techniques are sensitive to non-uniform jitter, changing background, and clutter. In this paper, we propose graph Laplacian assisted parametric dictionary learning, *GraDED* to account for the aforementioned variations. The temporal occurrence and duration of the event is determined from weights learned using a dynamic graph, while the spatial localization is performed by graph based dictionary learning. We demonstrate the efficacy of our approach by comparing with three state-of-the-art methods and achieve on average an overall increase of 0.08 in specificity and 0.6 in sensitivity for event detection.

***Index Terms*—** event detection, graph theory, dictionary learning, graph Laplacian.

1. INTRODUCTION

Designing an automated methodology for event detection and recognition is a critical task in security and surveillance. This problem demands localization of an event both in space and time from video data. In order to obtain a potentially acceptable solution to the problem, it is imperative to separate the foreground, which is the region of our interest, from the background. Conventional methods assume a static or quasi-static background [1] where the foreground is designed as a sparse matrix by considering the event as a local phenomenon. Other methods such as the saliency-driven event detection developed in [2] also assume the static nature of background. As an alternative, frame-wise or block-wise principle component analysis (PCA) [3] captures the direction and magnitude of maximum variability in a sequence of video frames. Robust PCA [4] extends this framework by retrieving the background as a low-rank matrix to incorporate minor spatio-temporal non-stationarity due to jitter, clutter and variation in illumination. In other work, the background is modeled as a mixture of Gaussians [5] to capture variations along with scene changes.

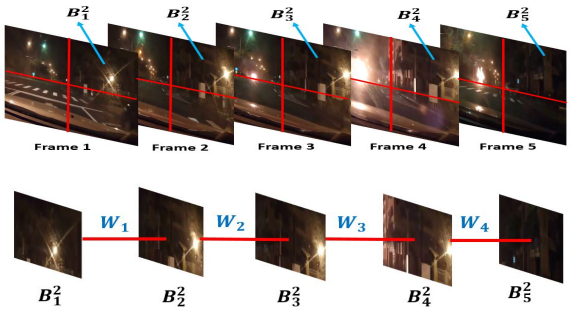


Fig. 1. Top row: An event video with four segments per frame. Bottom row: The linear graph with five nodes and four weight parameters for the 2^{nd} sub-volume.

Real-time videos captured by car-mounted or hand-held cameras cannot be effectively analyzed with the aforementioned models. The high-degree of non-stationarity in the background, sporadic changes in camera angle (especially while driving on uneven road surface or persistent swaying motion of human gait while holding the camera for example), and illumination changes affect the performance of existing algorithms. In this work, we aim at detecting events exhibiting limited duration and sufficient local extent in presence of dynamic background. In the majority of these scenarios (one such example is shown in Fig. 1, there may be no actual object present to track. So, analyzing the motion of objects to detect anomalies in videos using a tracking algorithm [6, 7] or trajectory estimation [8, 9] is not an appropriate solution for such data.

In recently reported work, it has been shown that subspace-based methods are keen to detect subtle changes of dynamic background [10–12]. The key idea is to represent variability of a feature in terms of the coefficients of a subspace. It is expected that the high-degree nonlinearity caused by the variation in an agent such as jitter can be projected onto a subspace allocated for that agent. We seek to find the assembly of these subspaces, which we call a dictionary [13], to detect events in a video. There have been few emerging methods that used dictionary learning to build subspaces and exploit the reinforced sparsity of the input data to formulate

(a) Video 1 (b) Video 2 (c) Video 3

Fig. 2. The video in (a) shows disappearance and reappearance of pilot boat; (b) shows a spontaneous fire on a street. (c) shows a accident on a highway.

suitable measures for event detection (SSPARED) [14]. The SSPARED algorithm exploits sparse codes obtained by cross-dictionary representation in conjunction with K-L divergence to detect substantial changes in consecutive frames. The algorithm is computationally expensive due to the dictionary construction per frame and cross-dictionary representation for each pair of consecutive frames and only provides the temporal extent of an event.

In this paper, we develop a block-based graph-assisted dictionary learning algorithm (*GraDED*) to identify both spatial and temporal extents of an event in a video with a dynamic background as shown in fig 1. In this work, the graph Laplacian weights are employed to detect the temporal extent of the event. The learned dictionary is analyzed to determine the spatial localization of the event.

2. METHODOLOGY

Before delving into the formulation of our algorithm, we give an overview of the graph tools that is used in the derivation.

2.1. Graph theory

A graph with P vertices and a set of edges E with corresponding weights W can be symbolically represented as a triplet $\mathcal{G} = \{P, E, W\}$ [15]. A graph is said to be *simple* if it does not contain any self-loop or multiple edges between two vertices. The *structural* degree of a vertex is defined as the number of edges linked to the vertex. In this work, we consider a simple, *linear* graph, where all vertices have structural degree 2 except terminal ones, which have unit degree. Mathematically, a graph can be expressed in terms of different matrices. The *incidence* matrix, $\Upsilon \in \{0, 1, -1\}^{(P\times E)}$ of a graph $\mathcal{G}$ is a $(P \times E)$ matrix with each entry in Υ denotes whether the j^{th} edge is going to or coming out from i^{th} vertex. Let W be a diagonal matrix with all the weights along the main diagonal. The Laplacian matrix of $\mathcal{G}$ is given by $L = \Upsilon W \Upsilon^T$. T is the matrix transpose operator.

2.2. Spatio-temporal graph representation of video

Let V be a video containing K frames, where each frame is partitioned into P non-overlapping blocks (see fig. 1). Collectively the set of i^{th} block of all frames constitutes the i^{th} sub-volume of the video. $B^i_j \in \mathcal{R}^{1\times S}; j \in \{1, 2, \cdots, K\},\ i \in \{1, 2, \cdots, P\}$ is a feature extracted from i^{th} block. The set of all features in i^{th} sub-volume, denoted as $B^i \in \mathcal{R}^{K\times S}$, is used to construct the dictionary of i^{th} block.

Firstly, we perform PCA on B^i to obtain maximum coherence within the basis of each block. We take a subset of leading eigenvectors in terms of the magnitude of eigenvalues. The span of the eigenvectors form the subspace of principle variation in i^{th} block over all the frames. The coherence [16] between two basis matrices can be defined as

$$\mu\big(X, Y\big) = \max_{k,j} | < x_k, y_j > |, \tag{1}$$

where x_k and y_j are the k^{th} and j^{th} columns of X and Y respectively. The coherence within i^{th} block is called *intra-block coherence*; whereas between two different blocks, it is *inter-block coherence*. As eigenvectors are orthonormal, μ_i is unity for i^{th} block. However, no inference can be made regarding the coherence between two different blocks.

Since, B^i The principle components of B^i can be exploited to analyze the temporal variations in a video. Let M_i be the number of leading eigenvectors, which is denoted as $\chi_i \in \mathcal{R}^{K\times M_i}$. The choice of the number of eigenvectors is critical, since a large number would increase the computational time of the algorithm, where as a small number may fail to capture the background and foreground variations.

Now, motivated by [17], the idea is to subject χ_i by a distance-preserving linear transformation such that a subset of P blocks will have significant mutual incoherence with respect to the rest without effecting the *intra-block mutual coherence*. Then, that subset of blocks can be identified as the spatial localization of the event in a video. Let U be such desired transformation. Precisely, we want $\mu_i(U^T\chi_i, U^T\chi_i) = \mu_i(\chi_i, U * U^T\chi_i) = \mu_i(\chi_i, \chi_i)$. It is evident that U has to be unitary. We want such transformation to be data-driven, which motivates us to apply graph theory.

The graph provides a useful framework to iteratively retrieve meaningful relationships among data samples. We consider each block of a frame as a vertex of a graph. In i^{th} sub-volume, there are K number of vertices which are connected as a linear graph having $(K-1)$ weights as defined in section 2.1. The eigenmatrix of the graph Laplacian, L_i, is derived by using the set of weights is our desirable transformation for the i^{th} sub-volume. The weights are essentially free parameters, which are updated according to a cost function. The changes in weights lead to a change in L_i, which consequently induces a change in U_i, which is the eigenmatrix of L_i.

To build a compact mathematical framework, let, χ_i is left-multiplied by a unitary matrix U_i, which is considered to be an the i^{th} sub-dictionary, D_i. The overall dictionary structure is given by,

$$D = [D_1\ D_2\ \ldots\ D_P] = [U_1^T\chi_1\ U_2^T\chi_2\ \ldots\ U_P^T\chi_P] \tag{2}$$

Here $M = \sum_{i=1}^{P} M_i$ and $D \in \mathcal{R}^{K\times M}$. In eq. (2), there are total $P(K-1)$ parameters of the dictionary which needs to be updated iteratively for P number of linear graphs. We

select each graph to be linear as it contains the least number of edges for a given set of vertices in a connected graph. Therefore, this configuration contains least number of parameters, which, in effect, reduces the possibility of over-fitting during optimization. It should be noted here, from graph-theoretic perspective, that there are multiple isomorphic candidates which share the property of least number of edges. For example, a star-graph contains the same number of edges as of a linear graph with a fixed cardinality of vertex sets. However, only linear graph maintains the temporal order of the event occurring in a video.

2.3. Graph based parametric dictionary learning

In this section, we attempt to find the set of transformations, U_is by using graph Laplacians. Let us consider, the sequential stack of sub-volume features as $Y = [B^1\ B^2\ ...\ B^P]$. The corresponding high-dimensional sparse code is $X = [X^1\ X^2\ ...\ X^P]$. The optimization problem by using D, Y, and X can be given by,

$$\big(D^*, X^*\big) = \min_{D,X} \|Y - DX\|_F^2 \text{ s.t.} \|X\|_0 \leq T. \tag{3}$$

The objective function in the above equation is non-convex. Conventionally, we resort to an alternating minimization technique in which each quantity is minimized by keeping the other one fixed. The above non-convex cost function and the constraints in eq. (3) can be stitched together with the help of Lagrangian coefficient λ as

$$\phi\big(D, X, \lambda\big) = \|Y - DX\|_F^2 + \lambda \|X\|_0 . \tag{4}$$

It is to be noted here that the dictionary D is a function of a set of Laplacians L_i via U_i as evident from eq. (2). In addition, each L_i is a function of diagonal weight matrix W_i. Therefore, the cost function in eq. (4) can be rewritten as

$$\begin{aligned} \phi\big(\{W_1, ..., W_P\}, X, \lambda\big) &= \|Y - DX\|_F^2 + \lambda \|X\|_0 \\ L_i &= U_i \Gamma_i U_i^T, L_i = \Upsilon W_i \Upsilon^T; i \in \{1, 2, ..., P\} \end{aligned} \tag{5}$$

In alternating minimization method, the sparse feature matrix X is kept fixed during the update step of dictionary D. By using eq. (2), the feature Y in eq. (5) can be expanded as $Y = \sum_{i=1}^{P} D_i X_i$.

Note that X_i is not the sparse code for B^i, which we distinguish by subscript and superscript in notation. Rather, it can be interpreted by the row-wise partition of X i.e. the i^{th} row-block of X is multiplied by D_i. By using $Y = \sum_{i=1}^{P} D_i X_i$, the first term in eq. (5), which is the reconstruction error, can be restructured as a function of D_i as

$$\phi\big(D_i\big) = ||E_i - D_i X_i||_F^2;\ E_i = Y - \sum_{j \neq i} D_j X_j. \tag{6}$$

To update the block dictionary D_i iteratively, the parameters W_i needs to be estimated by gradient descent. With the help of eq. (5) and (6), the update equation for W_i can be given by,

$$\begin{aligned} w_{ij}^{(t+1)} &= w_{ij}^t - \eta Tr\Big(\big[\frac{\partial \phi(D_i)}{\partial U_i}\big]^T \frac{\partial U_i}{\partial w_{ij}}\Big); \\ \frac{\partial U_i}{\partial w_{ij}} &= \Big[Tr\Big((\frac{\partial L}{\partial u_{kl}})^{-T} \frac{\partial L}{\partial w_{ij}}\Big)\Big]_{kl} \end{aligned} \tag{7}$$

Here, w_{ij}^t denotes the j^{th} weight of W_i in eq. (5) at t^{th} iteration. η is the learning step parameter, a scalar value selected from $(0, 1)$. Tr is the matrix trace operator which sums up the diagonal values of a matrix. $[\bullet]_{kl}$ is the kl^{th} element of the matrix expressed as $[\bullet]$. It is evident from eq. (7) that in order to update weight w_{ij}, three partial derivatives are to evaluated at each step - $\frac{\partial \phi(D_i)}{\partial U_i}$, $\frac{\partial L}{\partial u_{kl}}$, and $\frac{\partial L}{\partial w_{ij}}$. Precisely, in order to obtain $\frac{\partial U_i}{\partial w_{ij}}$, for each (k, l) a total of K^2 computation of $\frac{\partial L}{\partial u_{kl}}$ is needed, which is computationally expensive. However, it appears that the loss of computational simplicity is compensated by considerably fast convergence of the algorithm.
Now, $\frac{\partial \phi(D_i)}{\partial U_i}$ in eq. (7) can be evaluated by using eq. (6).

$$\begin{aligned} \frac{\partial \phi(D_i)}{\partial U_i} &= \frac{\partial \|E_i - D_i X_i\|_F^2}{\partial U_i} \\ &= \frac{\partial}{\partial U_i}\Big(E_i - D_i X_i\Big)^T \Big(E_i - D_i X_i\Big) \\ &= \frac{\partial}{\partial U_i}\Big(E_i^T E_i - 2E_i^T D_i X_i + X_i^T D_i^T D_i X_i\Big). \end{aligned} \tag{8}$$

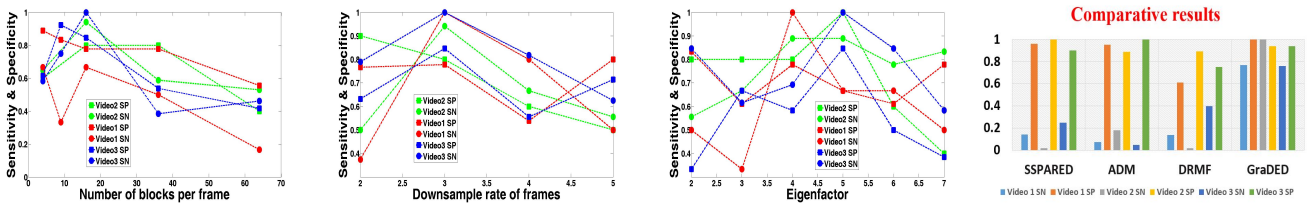
By construction $D_i = U_i \chi_i$ from eq. (2), which asserts that $D_i^T D_i = \chi_i^T U_i U_i^T \chi_i = \chi_i^T \chi_i = I$ by using $U_i U_i^T = I$. It immediately appears that $X_i^T D_i^T D_i X_i = X_i^T X_i$. After inserting this expression in eq. (8), $\frac{\partial \phi(D_i)}{\partial U_i}$ becomes $-2\chi_i X_i E_i^T$. The second partial derivative, $\frac{\partial L}{\partial u_{kl}}$ can be evaluated by using $L_i = U_i \Gamma_i U_i^T$, where Γ is a diagonal matrix containing the eigenvalues of L_i. By standard rule of matrix derivative and using $J^{mn} = \delta_{mk}\delta_{nl}$,

$$\frac{\partial L_i}{\partial u_{kl}} = \frac{\partial}{\partial u_{kl}} U_i \Gamma_i U_i^T = U_i \Gamma_i J^{mn} + J^{nm} \Gamma_i U_i^T. \tag{9}$$

2.4. Event detection using learned graph weights

The event detection is performed by identifying its spatial and temporal extent independently. Both problems can be categorized as binary classification problems - associated with event or no-event.

To figure out the temporal changes, each set of weights between consecutive frames is taken as a vector to represent temporal variability. The weight vectors are then clustered by k-means with Euclidean distance. The spatial localization is obtained via clustering dictionary atoms. As dictionary atoms are inherently orthonormal, a refined approach would be to consider clustering over a Grassmann manifold. However, such clustering involves a computationally intensive intermediate step of Karcher mean calculation. Instead, we perform



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

Fig. 3. For three video datasets: sensitivity and specificity vs. (a) number of blocks per frame P, (b) downsample rate of frames and (c) eigenfactor; (d) comparison with state-of-the-art methods. SP = Specificity, SN = Sensitivity.

agglomerative clustering to sort out the subset blocks carrying event signatures. We use an unweighted average distance to measure similarity between clusters and inter-block coherence as the distance between two features in the hierarchical clustering.

3. IMPLEMENTATION, RESULTS & DISCUSSION

We evaluate the performance of our proposed algorithm on three video datasets [14], as shown in fig. 2. Video I contains 75 frames depicting the disappearance and reappearance of a pilot boat. In this video, the mounted camera position is kept fixed but the background variation is primarily caused by intermittent sea-waves. Video II with 69 frames shows an explosion in a gas station. The camera mounted on a running vehicle captures significant background motion. Video III contains 77 frames showing a car accident followed by a fire during daytime from a car-mounted camera.

For temporal localization of an event, we set the ground-truth by tagging the frames manually in which the event persists including the start and end frames. We compare our results with three recent methods - SSPARED [14], ADM [12], and DRMF [11]. In addition, we also provide spatial localization by identifying the subset of blocks containing the event. We use *sensitivity* and *specificity* as metrics to evaluate our results for spatial and temporal localizations separately.

In all of our experiments, we extract block-wise histogram of oriented gradient (HOG) [18] features with a cell size of 16×16, followed by dimension-wise normalization of the features as a prerequisite for orthogonal matching pursuit to generate sparse code. With the user-specified parameters, the alternative minimization procedure is executed for dictionary update and sparse code generation. During this procedure, as an intermediate step, the weights of all the graphs are normalized before obtaining the Laplacian matrices at every iteration.

We demonstrate the effects of different parameters - number of blocks (P), downsampling of frames, and number of leading eigenvectors (M) on sensitivity and specificity. We do not set M directly for our experiments. Instead, we introduce a parameter, *eigenfactor* such that $M_i = \left\lceil \frac{K}{eigenFactor} \right\rceil \ \forall i$. The results shown in fig. 3(a) suggest that the desirable choice of the number of blocks, P would be 16, implying the preference for coarse spatial partition. It is because, for significantly large number of blocks per frame, the signature of an event will be distributed among the blocks making it harder to distinguish from the background. Similarly, from fig. 3, the preferred downsample rate is 3. For higher rates, the variation between consecutive frames would be significant leading to an erratic prediction of event frames. It is to note that downsample reduces the number of graph weights $(K-1)$, which provides two advantages - computational speed and reduced possibility of over-fitting. Likewise, according to fig. 3(c), the preferred eigenfactor would be 4 or 5. Higher eigenfactors make the sub-dictionaries thinner, which precludes to capture major variations. Dictionaries with a lower eigenfactor encounter almost every possible unwanted variation and loss of computational speed. Fig. 3(d) presents the comparison with SSPARED, ADM, and DRMF methods, and it shows improvements in specificity and sensitivity by 0.08 and 0.6 respectively for time localization. The results of spatial localization for the three video datasets by our method are given in table 1.

Table 1. Sensitivity & specificity scores of GraDED for spatial localization.

Dataset	Sensitivity	Specificity
Video I	0.75	0.59
Video II	1	0.54
Video III	0.8	0.64

4. CONCLUSION

We present a parametric dictionary learning approach by leveraging the graph framework to detect short-time, spatially-local rare events. In this work, the variations of moving backgrounds, jitter, clutter and varied illumination are overcome by utilizing block-wise subspaces in the dictionary. The graph plays a crucial role in spatial localization of events by iteratively updating the edge-weights. We show the effectiveness of our algorithm by identifying the temporal extent of the events compared to three different state-of-the-art methods. In addition, we perform simultaneously spatial localization by locating the set of blocks in frames in which the events happened. One future endeavor involves the derivation of the graph Laplacian from the sparse code, and the control of the regularization of the Laplacian in order to obtain spatio-temporal event localization.

5. REFERENCES

[1] Massimo Piccardi, "Background subtraction techniques: a review," in *Systems, man and cybernetics, 2004 IEEE international conference on*.

[2] Fahad Fazal Elahi Guraya, Faouzi Alaya Cheikh, Alain Tremeau, Yubing Tong, and Hubert Konik, "Predictive saliency maps for surveillance videos," in *Distributed Computing and Applications to Business Engineering and Science (DCABES), 2010 Ninth International Symposium on*. IEEE, 2010, pp. 508–513.

[3] P Wayne Power and Johann A Schoonees, "Understanding background mixture models for foreground segmentation," in *Proceedings image and vision computing New Zealand*, 2002, vol. 2002, pp. 10–11.

[4] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright, "Robust principal component analysis?," *Journal of the ACM (JACM)*, vol. 58, no. 3, pp. 11, 2011.

[5] Chris Stauffer and W Eric L Grimson, "Adaptive background mixture models for real-time tracking," in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*

[6] R Sarkar, Samarjit Das, and Namrata Vaswani, "Tracking sparse signal sequences from nonlinear/non-gaussian measurements and applications in illumination-motion tracking," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*.

[7] Suvadip Mukherjee, Rituparna Sarkar, Joshua Vandenbrink, Scott T Acton, and Benjamin Blackman, "Tracking sunflower circumnutation using affine parametric active contours," in *Image Analysis and Interpretation (SSIAI), 2014 IEEE Southwest Symposium on*.

[8] Weiming Hu, Xuejuan Xiao, Zhouyu Fu, Dan Xie, Tieniu Tan, and Steve Maybank, "A system for learning statistical motion patterns," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 9, pp. 1450–1464, 2006.

[9] Arslan Basharat, Alexei Gritai, and Mubarak Shah, "Learning object motion patterns for anomaly detection and improved object detection," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*.

[10] Jihun Hamm and Daniel D Lee, "Grassmann discriminant analysis: a unifying view on subspace-based learning," in *Proceedings of the 25th international conference on Machine learning*.

[11] Liang Xiong, Xi Chen, and Jeff Schneider, "Direct robust matrix factorizatoin for anomaly detection," in *Data Mining (ICDM), 2011 IEEE 11th International Conference on*.

[12] Amir Adler, Michael Elad, Yacov Hel-Or, and Ehud Rivlin, "Sparse coding with anomaly detection," *Journal of Signal Processing Systems*, vol. 79, no. 2, pp. 179–188, 2015.

[13] Michal Aharon, Michael Elad, and Alfred Bruckstein, "rmk-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on signal processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[14] Rituparna Sarkar, Andrea Vaccari, and Scott T Acton, "Sspared: Saliency and sparse code analysis for rare event detection in video," in *Image, Video, and Multidimensional Signal Processing Workshop (IVMSP), 2016 IEEE 12th*.

[15] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[16] Emmanuel Candes and Justin Romberg, "Sparsity and incoherence in compressive sampling," *Inverse problems*, vol. 23, no. 3, pp. 969, 2007.

[17] Xuan Zhang, Xiaowen Dong, and Pascal Frossard, "Learning of structured graph dictionaries," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*.

[18] Navneet Dalal and Bill Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*.